Local Stochastic Sensitivity Analysis For Dynamical Systems

Nishant Panda Los Alamos National Lab.
Jehanzeb Chaudhry University of New Mexico
Natalie Klein Los Alamos National Lab.
James Carzon Carnegie Mellon University
Troy Butler University of Colorado Denver

Abstract

We derive local sensitivities of statistical quantities of interest with respect to model parameters in dynamical systems. Our main contribution is the extension of adjoint-based *a posteriori* analysis for differential operators of generic dynamical systems acting on states to the Liouville operator acting on probability densities of the states. This results in theoretically rigorous estimates of sensitivity and error for a broad class of computed quantities of interest while propagating uncertainty through dynamical systems. We also derive Monte-Carlo type estimators to make these estimates computationally tractable using spatio-temporal normalizing flows and exploiting the hyperbolic nature of the Liouville equation. Three examples demonstrate our method. First, for verification of the theoretical results, we use a 2D linear dynamical system with an initial multivariate Gaussian density. Then, we apply our method to the challenging task of propagating uncertainty in a double attractor system to illustrate sensitivities in bimodal distributions. Finally, we show that our method can provide sensitivities with respect to the parameters of Neural Ordinary Differential Equations (here, in the context of classification).

1 Introduction

Dynamical systems play a vital role in modeling complex temporal and spatio-temporal phenomena, with applications that span finance, geophysics, climate [Dijkstra, 2013, Stanley et al., 2024], and biology [Furusawa and Kaneko, 2012, Zhao, 2017]. These systems are often characterized by parameter-dependent differential equations that describe the evolution of state variables over time and space. However, accurate predictions hinge on elusive initial and boundary conditions. In settings where neural ordinary differential equations (ODE) [Chen et al., 2018, Biloš et al., 2021, Dupont et al., 2019, Norcliffe et al., 2021] are utilized for supervised learning or time series forecasting, the parameters of the neural network are influenced by uncertain or incomplete training data; consequently, prediction tasks must cope with inherent uncertainties no matter the setting [Ott et al., 2023]. Rather than seeking precise point predictions, we often prioritize the estimation of probabilities and their sensitivities.

For example, pipeline networks model energy supply and demand. Such networks are characterized by the pipeline topology, the components (e.g. compressor) and the parameters of these components, for example, a friction parameter, the length of the pipes connections, etc. Important questions to consider include: 1) What is the probability that a compressor will fail if there is a cold freeze and the network is tasked to meet the energy demands, and 2) How sensitive is the probability to the various compressor parameters? This manuscript is written with this in mind to address questions like this, since even in a deterministic system, initial conditions are often uncertain and the sensitivity of QoIs, such as those defined by the probability of rare events to network parameters, are critically important to compute.

We conceptualize the evolution of states, denoted as $\boldsymbol{x}(t)$, as a stochastic process, where at each time t the state $\boldsymbol{x}$ is a random element within a suitable space. To rigorously define the probabilistic evolution of states in this work, we articulate a process model defined by a transport operator (specifically, a Liouville operator) that defines the evolution of an arbitrary probability density function (PDF) denoted by $\varrho(\boldsymbol{x}, t)$ with an appropriately defined initial condition that summarizes uncertainties on initial conditions along with model and statistical parameters. This paper concentrates on the specific case of ODE-governed dynamics, which

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

enables an analytical description of the process model. Our primary contributions are twofold:

Contribution 1. We derive an *evolution equation of sensitivity* by leveraging adjoint-based techniques to provide a rigorous framework that allows computable estimates of errors and sensitivities in Quantities of Interest (QoI) that are computed as expected values of arbitrary functions of uncertain dynamical states at a given time. We believe that this is the first such result in the current literature.

Contribution 2. The fundamental challenge in the practical application of this framework is the estimation of spatio-temporal derivatives of the PDF. We introduce the Liouville Normalizing Flow (ℓNF), a spatio-temporal density estimator that approximates the time evolution of probability density across a state space governed by ODEs and provides readily accessible estimates of derivatives.

2 Probabilistic Description of Uncertainties In ODEs

We investigate a given dynamical system governed by an ODE that describes the time evolution of a finite-dimensional state $\boldsymbol{x}$. This is a fairly general setting since, for example, most evolutionary PDEs are evolved (through numerical discretization) as a system of coupled ODEs; time series and Bayesian flows are commonly modeled as ODEs [Eisenhammer et al., 1991, Rubanova et al.,]; and so on. Consider the ODE describing the evolution of the state $\boldsymbol{x} \in \mathbb{R}^d$ for some positive integer d given by the equation

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{\theta}), \qquad \boldsymbol{F} : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^d. \tag{1}$$

For simplicity, we present results only for autonomous ODEs and note that any non-autonomous ODE is easily rewritten as an autonomous ODE via the usual state augmentation trick. The parameters $\boldsymbol{\theta}$ in the governing dynamics can appear as part of any source or data term (e.g., temporal functions exciting the system or initial conditions), or can be vectors/finite-dimensional arrays that quantify physical parameters (appearing as coefficients in the ODE), or can be parameters of a neural network. If the parameters themselves evolve according to some ODE, then we will consider them as part of the state vector $\boldsymbol{x}$. In contrast to the temporally variable state $\boldsymbol{x}(t)$, we reserve the term *parameters* to describe stationary *model parameters* i.e. $\dot{\boldsymbol{\theta}} = \boldsymbol{0}$. The (model) parameter space, $\Theta \subset \mathbb{R}^p$, is then defined as the space of all stationary variables. We assume that the ODE is solvable for *any* parameter in Θ on at least an open interval (t_0, t_f).

We define uncertainty in the system by encoding probability spaces associated with the state over time, which leads to a stochastic process in the state space $\mathcal{X}$. The methods and algorithms developed in this paper are easily extended to account for uncertainty in Θ as well as $\mathcal{X}$ by considering a *augmented space* similar to how a non-autonomous ODE is transformed into an autonomous ODE (see Appendix A). Therefore, it is with no loss of generality that we consider the uncertainty only in $\mathcal{X}$ for an arbitrary but fixed choice of $\theta \in \Theta$. Let $(\mathcal{X}, \mathcal{B}_\mathcal{X})$ denote the measurable space where $\mathcal{B}_\mathcal{X}$ denotes the Borel σ-algebra inherited from $\mathbb{R}^d$. Since $\boldsymbol{x}(t)$ evolves according to Eq. (1), we denote by $\mathbb{P}_{\mathcal{X},t}$ for $t \in (t_0, t_f)$ the entire collection of probability measures for which Eq. (1) is solvable. Assume that the initial probability measure is described by the PDF $\pi(\boldsymbol{x}; \gamma)$ where γ denotes any *statistical parameters* used to define this PDF (e.g., mean and covariance if π is Gaussian). By $\boldsymbol{X}(t)$ we denote the random state at time $t \in (t_0, t_f)$. The collection $\{\boldsymbol{X}(t) : t \in (t_0, t_f)\}$ is then a stochastic process in $(\mathcal{X}, \mathcal{B}_\mathcal{X})$ where $\boldsymbol{x}(t)$ is a sample of $\boldsymbol{X}(t)$ at time t and *law* of $\boldsymbol{X}(t)$ is given by $\mathbb{P}_{\mathcal{X},t}$. This is not a stochastic process in the *common meaning* attached to stochasticity since given a state at time t we always have a deterministic trajectory of the flow to time $t + \delta t$; the stochasticity of the process lies solely in its initial dynamical uncertainty. Assuming $\pi(\cdot; \gamma) \in \mathcal{C}^k(\mathcal{X})$ (a.e.) for some suitable integer k, the probability measure $\mathbb{P}_{\mathcal{X},t}$ for $t \in (t_0, t_f)$ is characterized by the PDF $\varrho(\cdot, t)$ satisfying the Liouville initial-boundary value problem

$$\frac{\partial \varrho}{\partial t} + \operatorname{div}(\varrho \boldsymbol{F}) = 0 \tag{2a}$$

$$\varrho(\boldsymbol{x}, t_0) = \pi(\boldsymbol{x}; \gamma) \tag{2b}$$

$$\varrho(\boldsymbol{x}, t) = 0, \ (\boldsymbol{x}, t) \in \partial\mathcal{X} \times (t_0, t_f). \tag{2c}$$

For a proof of the above result, see [Lasota and Mackey, 1998](7.6.11)).

Definition 2.1. The *Liouville differential operator* $L[\cdot]$ is given by

$$L[\cdot] := \frac{\partial [\cdot]}{\partial t} + \operatorname{div}([\cdot]\boldsymbol{F}). \tag{3}$$

Remark 2.2. For any time $t \in (t_0, t_f)$, given an initial PDF $\pi(\boldsymbol{x}; \gamma)$ in the state, the equation above describes *evolution* of this PDF. Note that $\varrho(\boldsymbol{x}, t)$ depends on the statistical parameters γ via the initial PDF and also on the model parameters $\boldsymbol{\theta}$ via $\boldsymbol{F}$. We therefore write the time-varying density given by Eq. (2) as $\varrho(\boldsymbol{x}, t; \gamma, \boldsymbol{\theta})$ to make this dependence explicit.

Although the Liouville equation has been known for a long time, in practice, its stochastic cousin, the

Fokker-Planck equation, has received more attention. [Lasota and Mackey, 1998] provides discussion and derivation of both equations, which are directly related to gradient flows (see [Ambrosio et al., 2005, Santambrogio, 2017]). A complete uncertainty quantification (UQ) analysis of a dynamical system such as Eq. (1) is accomplished by a backward-and-forward framework. Backward UQ uses observed data to infer model parameters and the initial condition (e.g., via maximum-likelihood [McGoff et al., 2015a, McGoff et al., 2015b, McGoff and Nobel, 2020]), whereas forward UQ determines the pushforward of the initial uncertainty and model parameters through the model, although the pushforward of the entire state is not of interest for the present work. In this work, we address the situation in which some set of m (potentially nonlinear) functionals of the state, called Quantities of Interests (QoIs), $\{Q^{(j)}\}_{j=1}^m$, are deemed important. The forward UQ problem reduces to two tasks: (i) determine the expected values of these QoIs under initial condition uncertainty and (ii) compute the sensitivities of these expectations to both model parameters θ and statistical parameters γ. The focus of this paper is on these forward UQ problem tasks.

3 Proposed Work

In this work, we define a QoI as an expected value for a broad class of functions.

Definition 3.1. A quantity of interest (QoI) is defined as

$$Q(t,\gamma,\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{x}\sim\varrho(\boldsymbol{x},t;\gamma,\boldsymbol{\theta})}[g(\boldsymbol{x})], \tag{4}$$

where $g:\mathcal{X}\to\mathbb{R}$ is assumed only to be $\mathbb{P}_{\mathcal{X},t}$-integrable on $\mathcal{X}$.

Typical QoIs are probabilities of events, e.g., where $g(\boldsymbol{x}) = 1_A(\boldsymbol{x})$ for some $A\in\mathcal{B}_{\mathcal{X}}$. Other common QoIs correspond to moments of particular states, for example, where $g(\boldsymbol{x}) = x_i^k$ for some $1\le i\le d$ and $k\in\mathbb{N}$. An important undertaking in UQ is the determination of sensitivity of such QoIs w.r.t. either model or statistical parameters, as both types impact the QoI computation. To that end, let $\boldsymbol{p} = (p_1,\cdots,p_N)$ be a parameter vector in $\mathbb{R}^N$ where a p_i could be either a model parameter θ_j or a statistical parameter γ_k. We consider *local sensitivities* given by derivatives. Thus, we are interested in computing $\frac{\partial Q}{\partial p_i}$ (which we often write compactly as Q_{p_i}) for $1\le i\le N$. With this new notation of a parameter vector, the QoI $Q(t,\gamma,\boldsymbol{\theta})$ is rewritten more compactly as $Q(t,\boldsymbol{p})$ and we denote *sensitivity gradient* by $\nabla_{\mathbf{p}}Q$. By $\boldsymbol{F}_{p_i}$ we denote the partial derivative $\frac{\partial \boldsymbol{F}}{\partial p_i}$.

Remark 3.2. Three observations are critical to prove the main results. First, the Liouville operator L in Definition 2.1 is *linear in the density* ϱ. Second, the QoIs in Definition 3.1 are *linear functionals on the space of probability densities on* $\mathcal{X}$. Finally, the formal adjoint of the Liouville operator is itself hyperbolic.

With this insight, we first present the *adjoint-based sensitivity result* (proof in Appendix B).

Theorem 3.3. *Let the initial pdf on the state be given by a smooth function $\pi(\boldsymbol{x};\gamma)$. Then the partial derivative of Q w.r.t p_i is given by,*

$$Q_{p_i}(t^*) = -\int_{t_0}^{t^*}\int_{\mathcal{X}} v\,\mathrm{div}(\varrho\boldsymbol{F}_{p_i})d\boldsymbol{x}dt + \int_{\mathcal{X}} v(\boldsymbol{x},t_0)\frac{\partial\pi}{\partial p_i}d\boldsymbol{x}, \tag{5}$$

where $t=t^*$ is a fixed time of interest and $v(\boldsymbol{x},t)$ is the solution to the adjoint problem*,*

$$L^\dagger[v] = 0 \quad \text{on } \mathcal{X}\times(t_0,t^*], \quad v(\boldsymbol{x},t^*) = g(\boldsymbol{x}(t^*)) \quad \text{on } \mathcal{X}, \tag{6}$$
$$v(\cdot,t) = 0 \quad \text{on } \partial\mathcal{X}\times[t_0,t^*).$$

The differential operator is $L^\dagger$ given by

$$L^\dagger[\cdot] := \left[-\frac{\partial[\cdot]}{\partial t} - \langle grad\,[\cdot],\boldsymbol{F}\rangle\right], \tag{7}$$

is called the adjoint Liouville operator *or the* continuous Koopman operator *(see [Lasota and Mackey, 1998]).*

Hence, given a parameter vector $\mathbf{p} = (p_1,\cdots,p_N)$ where p_i is either a model parameter θ_j or a statistical parameter γ_k, the sensitivity gradient is $\nabla_{\mathbf{p}}Q = (Q_{p_1},\cdots,Q_{p_N})$ where each Q_{p_i} is given by the above theorem.

Remark 3.4. Observe that to compute QoIs given in Eq. (4), we could employ Monte Carlo techniques with some sort of importance sampling without having to construct an estimator for the time-varying density $\varrho(\boldsymbol{x},t;\boldsymbol{\theta},\boldsymbol{\gamma})$. However, as Theorem 3.3 shows, in order to compute the sensitivity at some time $t^*\in(t_0,t_f)$ we not only need to know $\varrho(\boldsymbol{x},t;\theta,\gamma)$ but also the derivatives $\varrho(\boldsymbol{x},t;\theta,\gamma)$ with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$.

Denote by $\widehat{\varrho}(\boldsymbol{x},t)$ any computable *spatio-temporal density estimator*. Substituting this estimator into Eq. (4) and Eq. (5) yields the following estimate of the QoI.

$$\widehat{Q} := \int_{\mathcal{X}} g(\boldsymbol{x},t)\widehat{\varrho}(\boldsymbol{x},t)d\boldsymbol{x}, \tag{8}$$

and the following computable estimate of the QoI sensitivity

$$\widehat{Q_{p_i}}(t^*) = -\int_{t_0}^{t^*}\int_{\mathcal{X}} v(\boldsymbol{x},t)\,\mathrm{div}(\widehat{\varrho}\boldsymbol{F}_{p_i})d\boldsymbol{x}dt + \int_{\mathcal{X}} v(\boldsymbol{x},t_0)\frac{\partial\pi}{\partial p_i}d\boldsymbol{x}. \tag{9}$$

Furthermore, the adjoint operator $L^\dagger$ allows us to derive the following *computable error estimate* for Eq. (8) (proof in Appendix B), which is valid for *any* density estimator.

Theorem 3.5. *If $\widehat{\varrho}(\boldsymbol{x},t)$ is any spatio-temporal density estimator for the density on the state $\mathcal{X}$ at time t, then for a fixed time $t^* \in (t_0,t_f)$ the error $e_Q := Q-\widehat{Q}$ is given by,*

$$e_Q(t^*) = -\int_{t_0}^{t^*}\int_{\mathcal{X}} v(\boldsymbol{x},t)\, L[\widehat{\varrho}]\, d\boldsymbol{x}dt \qquad (10)$$
$$+ \quad \int_{\mathcal{X}} v(\boldsymbol{x},t_0)\,(\pi(\boldsymbol{x}) - \widehat{\varrho}(\boldsymbol{x},t_0))\, d\boldsymbol{x}$$

For many QoIs (i.e., when m is large), it may seem we have a potential intractability due to the presence of m adjoint equations. However, each adjoint equation is hyperbolic and constant along the same flow (trajectories) given by Eq. (1). We describe this in Appendix B.2.

The equations above suggest that we look for an estimator $\widehat{\varrho}(\boldsymbol{x},t)$ that allows us to sample and evaluate likelihood simultaneously. In fact, if it is indeed possible to do so, we can utilize change of measure to construct a Monte-Carlo type estimator for Eqs. (8) to (10). We make use of the following definition.

Definition 3.6. Let n be a positive integer. Let $\boldsymbol{X}(t) = \{\boldsymbol{x}_1(t), \boldsymbol{x}_2(t), \cdots, \boldsymbol{x}_n(t)\}$ be n samples from $\widehat{\varrho}(\cdot,t)$. Let $w(\cdot,t) := v(\cdot,t)/\widehat{\varrho}(\cdot,t)$ and

$$\widehat{Q}^n(t) := \frac{1}{n}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}(t)} g(\boldsymbol{x}_i,t) \qquad (11)$$

$$G_1^n(t) := \frac{1}{n}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}(t)} w(\boldsymbol{x}_i,t)\,\mathrm{div}(\widehat{\varrho}(\boldsymbol{x}_i,t)\boldsymbol{F}_{p_i}(\boldsymbol{x}_i,t)), \qquad (12)$$

$$G_2^n := \frac{1}{n}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}(t_0)} w(\boldsymbol{x}_i,t_0)\frac{\partial\pi}{\partial p_i}(\boldsymbol{x}_i) \qquad (13)$$

$$H_1^n(t) := \frac{1}{n}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}(t)} w(\boldsymbol{x}_i,t)\, L[\widehat{\varrho}(\boldsymbol{x}_i,t)], \qquad (14)$$

$$H_2^n := \frac{1}{n}\sum_{\boldsymbol{x}_i\in\boldsymbol{X}(t_0)} w(\boldsymbol{x}_i,t_0)\,(\pi(\boldsymbol{x}_i) - \widehat{\varrho}(\boldsymbol{x}_i,t_0)) \qquad (15)$$

Utilizing the above definition, we have the following result (proof in Appendix B). In future work, we will include effective sample size estimates and CLT-type bounds for variance.

Theorem 3.7. *Assuming all expectations are finite and $\widehat{\varrho}(\cdot,t) > 0$ a.e. we have the following convergences as $n\to\infty$:*

$$\widehat{Q}^n(t) \to \widehat{Q}(t),$$
$$G_1^n(t) \to \mathbb{E}_{\boldsymbol{x}\sim\widehat{\varrho}(\cdot,t)}[w(\cdot,t)\,\mathrm{div}(\widehat{\varrho}\boldsymbol{F}_{p_i})],$$
$$G_2^n \to \mathbb{E}_{\boldsymbol{x}\sim\widehat{\varrho}(\cdot,t_0)}\left[w(\cdot,t_0)\frac{\partial\pi}{\partial p_i}\right]$$
$$H_1^n(t) \to \mathbb{E}_{\boldsymbol{x}\sim\widehat{\varrho}(\cdot,t)}[w(\cdot,t)\,L[\widehat{\varrho}(\cdot,t)]],$$
$$H_2^n(t) \to \mathbb{E}_{\boldsymbol{x}\sim\widehat{\varrho}(\cdot,t_0)}\left[w(\cdot,t_0)(\pi - \widehat{\varrho}(\cdot,t_0))\right].$$

Moreover, if the following functions are bounded

$$w(\cdot,t)\,\mathrm{div}(\widehat{\varrho}(\cdot,t)\boldsymbol{F}_{p_i}(\cdot,t)), \quad w(\cdot,t_0)\frac{\partial\pi}{\partial p_i},$$
$$w(\cdot,t)\,L[\widehat{\varrho}(\cdot,t)], \text{ and } \left[w(\cdot,t_0)(\pi - \widehat{\varrho}(\cdot,t_0))\right],$$

then we have the following convergences as $n\to\infty$:

$$\left(G_2^n - \int_{t_0}^{t^*} G_1^n(t)\right) \to \widehat{Q_{p_i}}(t^*), \text{ and} \qquad (16)$$

$$\left(H_2^n - \int_{t_0}^{t^*} H_1^n(t)\right) \to e_Q(t^*). \qquad (17)$$

A naive approach to construct an estimator is to solve the Liouville equation Eq. (2) via a numerical scheme for hyperbolic PDEs or use a neural PDE. Unfortunately, Eq. (2) describes a PDE on the potentially high-dimensional $\mathcal{X}$, which introduces significant computational challenges to this naive approach in practice while also failing to address the sampling issue. We therefore develop a *spatio-temporal* density estimator similar to that of [Feng et al., 2021] using normalizing flows (NFs) [Papamakarios et al., 2021]. NFs are a flexible class of density estimators that allow efficient likelihood estimation and/or data generation from complex distributions by parameterizing a *diffeomorphic* map in the state space. The main idea is to express the state $\boldsymbol{x}$ as a diffeomorphic transformation T of a real vector $\boldsymbol{z}$ sampled from a simpler density $p_{\mathcal{Z}}$ (typically standard Gaussian) and use the change of variables [Folland, 1999] to connect $p_{\mathcal{Z}}$ to the density of the state $\varrho(\boldsymbol{x})$. Let $\mathbb{P}_{\mathcal{Z}}$ be another probability measure in the state space $\mathcal{X}$. We assume that $\mathbb{P}_{\mathcal{Z}}$ is given by a simple (typically standard multivariate Gaussian) probability density function $p_{\mathcal{Z}}(\boldsymbol{z})$ defined in $\mathbb{R}^d$ and referred to as *base density on the state space.*

By a temporal Normalizing Flow (tNF), we mean a time-indexed parameterized map $T_{\phi_{\mathrm{NF}}} : \mathcal{Z} \to \mathcal{X}\times[t_0,t_f]$ that is diffeomorphic in $\mathcal{X}$ and $\mathcal{Z}$. That is, time plays the role of an index and, for a fixed $t\in[t_0,t_f]$, we denote the diffeomorphic mapping as $T_{\phi_{\mathrm{NF}},t} : \mathcal{Z}\to\mathcal{X}$. Since $T_{\phi_{\mathrm{NF}},t}$ is a diffeomorphism, the pushforward density [Bogachev and Ruas, 2007, Folland, 1999] of the base density $p_{\mathcal{Z}}$ under $T_{\phi_{\mathrm{NF}},t}$, denoted by $\widehat{\varrho}(\boldsymbol{x},t)$, is

given by the change of variables.

$$\widehat{\varrho}(\boldsymbol{x},t) = \quad p_{\mathcal{Z}}\left(\boldsymbol{z}\right)\left|\det J_{T^{-1}_{\phi_{\mathrm{NF}},t}}\left(\boldsymbol{z}\right)\right|, \qquad (18)$$
$$\text{where } \boldsymbol{z} = T^{-1}_{\phi_{\mathrm{NF}},t}(\boldsymbol{x}).$$

Note that we have for any $t \in [t_0, t_f]$,

$$\mathbb{E}_{\boldsymbol{x}\sim\widehat{\varrho}(\boldsymbol{x},t)}\left(\boldsymbol{x}\right) = 1,$$

that is, $\widehat{\varrho}(\boldsymbol{x},t)$ is a time-indexed density on $\mathcal{X}$. This idea of treating time as an index is similar to conditioning a normalizing flow on a latent variable [Winkler et al., 2023].

We learn the parameters ϕ_{NF} so that this pushforward density $\widehat{\varrho}$ is as close to ϱ at each time $t \in (t_0, t_f)$. While there are many metrics and pseudometrics utilized in loss functions (integral probability metrics, f-divergences, etc.), here, we utilize a combination of KL divergence and a physics-based residual loss. The forward KL divergence $D_{\mathrm{KL}}\left(\varrho(\boldsymbol{x},t) \,||\, \widehat{\varrho}(\boldsymbol{x},t)\right)$ is an expectation (w.r.t. ϱ) of $\log(\widehat{\varrho}/\varrho)$, which can be turned into a learning algorithm by minimizing the negative log-likelihood given samples of true density at time t. Although the true density $\varrho(\boldsymbol{x},t)$ is unknown in some $t > t_0$, samples of this density can be obtained (modulo numerical integration error) by sampling the initial density π and applying the dynamics F in Eq. (1) to evolve the initial samples from t_0 to t. The residual loss, also called the PINN loss [Karniadakis et al., 2021], involves the residual of Eq. (2). We describe this loss function precisely below.

Definition 3.8. Let λ_1, λ_2 and λ_3 be positive numbers, and let $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \lambda_3)$. Let $\mathcal{U}[t_0, t_f)$ denote a uniform probability density function on the interval $[t_0, t_f)$. The *loss function* is,

$$\mathcal{L}(\phi_{\mathrm{NF}}, \boldsymbol{\lambda}) := \lambda_1 \mathcal{L}_{KL} + \lambda_2 \mathcal{L}_{PINN} \qquad (19)$$

where the KL divergence loss term $\mathcal{L}_{KL}$ is,

$$\mathcal{L}_{KL} := \mathbb{E}_{t\sim\mathcal{U}[t_0,t_f)} D_{\mathrm{KL}}\left(\varrho(\cdot,t^*) \,||\, \widehat{\varrho}(\cdot,t^*;\phi_{\mathrm{NF}})\right), \quad (20)$$

and the PINN loss term $\mathcal{L}_{PINN}$ is,

$$\mathcal{L}_{PINN} := \mathcal{L}_{Res} + \lambda_3 \mathcal{L}_{Init},$$

where

$$\mathcal{L}_{Res} = \mathbb{E}_{t\sim\mathcal{U}[t_0,t_f)} \mathbb{E}_{\boldsymbol{x}\sim\varrho(\boldsymbol{x},t)} |L[\widehat{\varrho}(\boldsymbol{x},t)]|^2, \qquad (21)$$
$$\mathcal{L}_{Init} = \mathbb{E}_{\boldsymbol{x}\sim\pi(\boldsymbol{x};\gamma)} |\widehat{\varrho}(\boldsymbol{x},0) - \pi(\boldsymbol{x};\gamma)|^2. \qquad (22)$$

The terms $\mathcal{L}_{Res}$ and $\mathcal{L}_{Init}$ measure the residual loss in the space-time domain and the initial condition mismatch respectively. The PINN loss may be thought of as a regularizer to the KL divergence loss term, with tunable hyperparameters $\lambda_1, \lambda_2, \lambda_3$.

Remark 3.9. In problems governing physics, an alternative to the forward KL divergence is the reverse KL divergence. The reverse KL divergence, $D_{\mathrm{KL}}\left(\widehat{\varrho}(\boldsymbol{x},t) \,||\, \varrho(\boldsymbol{x},t)\right)$, involves an expectation over the base density. In Appendix Theorem B.2 (proof in Appendix B) we derive an estimator for the reverse KL. The key fact that is useful in our discussion now is that this estimator must evaluate the true likelihood; in other words, we must be able to evaluate the function $\varrho \circ T_{[\phi_{\mathrm{NF}},t^*]}$ on an arbitrary sample from the base density. To do this, we exploit the fact that Liouville equation Eq. (2) is linear and hyperbolic. We describe the full details in Appendix B.1. This was first discussed in the context of neural networks by [Chen et al., 2018] and was referred to as an instantaneous change of variables. In the context of UQ this was introduced by [Halder and Bhattacharya, 2011]. For general properties of this equation, we refer the reader to [Ambrosio et al., 2005]. In practice, any learning algorithm that uses gradient-decent with this reverse KL loss function must require adjoint sensitivity of Eq. (1) which can be (and usually is) expensive. Hence, we do not employ the reverse KL divergence estimator in the current work.

An unbiased estimate (modulo numerical integration error) of $\mathcal{L}(\phi_{\mathrm{NF}}, \boldsymbol{\lambda})$ is then obtained in the usual way. This defines our sampling strategy; that is, we sample space-time points to approximate the expectations in Eq. (20), Eq. (21) and Eq. (22). This is a unique feature of our ℓNF and focuses the points to sample where the density $\varrho(\boldsymbol{x},t)$ is high, removing dependence on any prior information from the space-time domain. This contrasts to the sampling strategy in [Feng et al., 2021], where only a PINN loss term is considered. The sampling strategy in that work samples the points distributed according to the learned ℓNF $\widehat{\varrho}$. However, the approximation $\widehat{\varrho}$ may be quite inaccurate in the early iterations of the optimization algorithm, and hence points may be sampled from regions where ϱ has a low value.

The mapping $T_{\phi_{\mathrm{NF}}}$ is often modeled by a deep neural network, and in this context the parameters ϕ_{NF} refer to the parameters of the neural network. Although our ℓNF framework is agnostic to the choice of NF architecture, in the examples in this paper, we employ two main NF architectures: an affine coupling architecture similar to [Feng et al., 2021], which is based on the RealNVP flow [Dinh et al., 2017], and a neural spline flow (NSF) which may be better suited for multimodal densities [Durkan et al., 2019]. Further details of the architectures are given in the Appendix C.

The algorithm to compute the sensitivity of a QoI is presented in Algorithm 1.

Remark 3.10. We were pointed to by a reviewer that

Algorithm 1 Compute sensitivity Q_{p_i} via proposed ℓNF

1: Generate samples at $\{\boldsymbol{x}_i\}_{i=1}^{N_x} \sim \pi(\boldsymbol{x})$
2: Form an equispaced set of points $\{t_i\}_{j=1}^{N_t}$ from $[t_0, t_f]$
3: Evolve $\{\boldsymbol{x}_i\}_{i=1}^{N}$ via Eq. (1) to generate samples $\{\boldsymbol{x}_i(t_j))\}_{i=1}^{N_x} \sim \varrho(\boldsymbol{x}, t_j)$ for $j = 1, \ldots, N_t$
4: Estimate loss $\mathcal{L}$ in Eq. (19) using the sample points
5: Minimize $\mathcal{L}$ w.r.t ϕ_{NF} to compute normalizing flow $T_{\phi_{\text{NF}}}$ to yield $\widehat{\varrho}$
6: Take Space-Time Samples $(t_i^q, \boldsymbol{x}_i^q)_{i=1}^{N_q}$ from $\widehat{\varrho}$.
7: **for** $i \in 1, \ldots, N_q$ **do**
8: Evaluate $\widehat{\varrho}(\boldsymbol{x}_i^q, t_i^q)$
9: Evaluate adjoint $v(\boldsymbol{x}_i^q, t_i^q)$ as in §B.2
10: **end for**
11: Compute $G_1^{N_q}(t_i)$ for $t_i \in [t_0, t_*]$ and compute $G_2^{N_q}$ at all points with $t_i = t_0$.
12: Compute $\widehat{Q_{p_i}}$ using $G_1^{N_q}, G_2^{N_q}$ via Eq. (16) (1).

a variational type argument would yield an approximation of the sensitivity of a QoI $Q(t, \gamma, \theta)$, provided the score can be esitmated. This is based on the fact that the gradient term can be written as

$$\begin{aligned}\nabla_\theta Q(t,\gamma,\theta) &= \int_X g(\boldsymbol{x}) \nabla_\theta \rho(\boldsymbol{x}, t, \gamma, \theta) d\boldsymbol{x} \\ &= \int_X g(\boldsymbol{x}) \nabla_\theta \log \rho(\boldsymbol{x}, t, \gamma, \theta) \rho(\boldsymbol{x}, t, \gamma, \theta) d\boldsymbol{x} \\ &= E_{\boldsymbol{x} \sim \rho(\cdot, t, \gamma, \theta)}[g(\boldsymbol{x}) \nabla_\theta \log \rho(\boldsymbol{x}, t, \gamma, \theta)]\end{aligned}$$

The above can be naively estimated by constructing an approximate pushforward for each θ starting from samples of π. This leads to an approximate sensitivity analysis and can be thought of as a discrete adjoint formulation; furthermore, the error estimation in Theorem 3.5 needs the exact adjoint.

4 Related Work

We attempt to situate the proposed work in the context of other UQ and machine learning/artificial intelligence (AI) concepts that are most closely related to adjoint-based methods for error estimation and sensitivity analysis, neural ODEs, and NFs.

The foundational work on adjoint-based error and sensitivity estimation dates back several decades. Excellent entry points in the context of ODEs are the works of [Estep, 1995] and [Cao and Petzold, 2004] that focus on error estimation/bounds while [Cao et al., 2003] focuses on sensitivity analysis of parameter-dependent differential-algebraic equations. [Chaudhry et al., 2015] utilizes a semidiscrete formulation to transform parabolic and hyperbolic PDEs into a system of ODEs for which error estimates are derived for a class of implicit-explicit time-stepping schemes. Broader entry points to the field are found in [Bangerth and Rannacher, 2003] and the review provided in [Grätsch and Bathe, 2005], both of which summarize the concepts in the context of finite element methods for general PDEs. Other works have leveraged related techniques for estimating errors in the forward propagation of uncertainties via surrogate models, e.g., polynomial chaos expansions (see [O'Hagan et al., 2013, Gerritsma et al., 2010, Butler et al., 2011, Bespalov et al., 2014] for just a few examples). However, all of these works focus on the adjoint for the state-space operator as opposed to the adjoint for the Liouville operator describing the evolution of the density on the states.

As mentioned in Section 3, [Chen et al., 2018] was the first to discuss the Liouville equation in the context of neural networks, while in the context of UQ this was introduced by [Halder and Bhattacharya, 2011]. However, as mentioned in Section 2, its stochastic cousin, the Fokker-Planck equation, has received more attention. [Lasota and Mackey, 1998] provides discussion and derivation of both equations, which are directly related to gradient flows (see [Ambrosio et al., 2005, Santambrogio, 2017]). The proposed work explicitly exploits the properties of the Liouville equation within the adjoint-based theory and computational algorithms.

Sensitivity estimation also appears in the statistics and machine learning literature. In statistics, global sensitivity methods, including Sobol/variance-based [Sobol', 1993], Bayesian [Oakley and O'Hagan, 2004], Monte Carlo [Helton and Davis, 2003] approaches, or polynomial chaos-based [Sudret, 2008], are widely used in analysis of computer models. These methods differ from local sensitivity analysis, which focuses on perturbations in local regions of input or parameter space [Saltelli et al., 2004]. In machine learning, methods similar to statistical sensitivity analysis are often applied for feature importance and explainability (including saliency maps [Simonyan, 2013], Shapley values [Strumbelj and Kononenko, 2010], gradient-based [Baehrens et al., 2010], and LIME [Ribeiro et al., 2016]). Although such methods bear some resemblance to our framework, they generally do not operate in a probabilistic setting to propagate uncertainty.

The ℓNF in the proposed work is the key step in constructing a practical estimator and fall under neural density estimators (see [Magdon-Ismail and Atiya, 1998, Papamakarios, 2019, Liu et al., 2021, Lou et al., 2020]

to name a few). Some good starting points for NFs are [Kobyzev et al., 2021, Papamakarios et al., 2021] and [Grathwohl et al., 2018]. As mentioned above, the ℓ NF is similar to the one in [Feng et al., 2021] except that the work focused on the Fokker-Planck equation in contrast to the proposed work that utilizes the Liouville equation. It is also worth noting three other specific advances in NFs and optimal transport in the context of UQ compared to the proposed work. [Graves et al., 2023] introduces a new class of generative models called Bayesian flow networks (BFNs) that utilize Bayesian inference for noisy data samples. Although this appears interesting in the context of UQ, it is not immediately obvious how to incorporate such BFNs in the context of the proposed work since the data incorporated in the ℓNF are not experimental / noisy, but are simulated through evolution through the Liouville operator. [Wan et al., 2023] proposes a deep learning based method for solving the dynamic optimal transport in high-dimensional space. Some features of that work are similar in spirit to the proposed work. For instance, they adopt the Lagrangian discretization method to deal with high dimensionality (similar to the proposed use of characteristics of the Liouville equation). They then adopt adjoint techniques to compute derivatives of the loss function for the purposes of back propagation to learn neural network parameters. This is promising within the general context of any learning algorithm for a NF or optimal transport problem, but it is distinct from the contributions of the proposed work that focuses on the utilization of the ℓNF output to obtain an estimator that, when combined with a general adjoint-based analysis for the Liouville equation, is useful for computing sensitivities and errors in QoI.

There is complementary literature on data-driven transfer learning techniques for state spaces of dynamical systems that employ the Koopman and Perron-Frobenius theory; see, e.g., [Lusch et al., 2018, Klus et al., 2016]. Although this literature is not focused on the goal-oriented analysis of error and sensitivity estimation, the methods are attractive when the objective is to analyze the full state of the system.

5 Results

We tune Liouville normalizing flow-based models for some key examples; the model implementation is described in detail in the Appendix C. Although the state space of all of our examples is small (2 dimensional), they exhibit all the characteristics of an *real-life* application and are meant to support the theory developed in this manuscript. Extensions to suitable applications, including modeling of the physics system or out-of-distribution sensitivity analysis for AI models, will be considered in future work. While it is difficult to do any baseline comparison, our Example 1 is intended to address this concern (at least partially) as the method is compared with a linear dynamical system with Gaussian initial condition whose time-varying density depends non-linearly on the parameter.

5.1 Linear Dynamical System

As a first example, we benchmark the performance of the ℓNF with a linear dynamical system.

$$\frac{d}{dt}\begin{bmatrix}x_1\\x_2\end{bmatrix} = \begin{bmatrix}-1 & 0\\ \theta & -1\end{bmatrix} x, \quad t \in [0,1], x \in \mathbb{R}^2, \tag{23}$$

and consider the initial distribution $\varrho(\boldsymbol{x}, t_0) = \pi(\boldsymbol{x}) \sim \mathcal{N}(\mathbf{0}, I_{2\times 2})$. We consider the following quantities of interest; $Q_1(\theta, t^*) = \mathbb{E}[1_A]$, along with the second moments $Q_2(\theta, t^*) = \mathbb{E}[x_1^2(t^*)], Q_3(\theta, t^*) = \mathbb{E}[x_1 x_2(t^*)], Q_4(\theta, t^*) = \mathbb{E}[x_2^2(t^*)]$ where A is the rectangle $[-1, 2] \times [0, 2]$. We use ℓNF hyper parameters as follows: 100 training epochs, 200,000 space-time sample points to estimate the loss function, six act norm / affine coupling layers with each affine layer having 32 neurons, Adam optimizer with a learning rate of 0.001, and a StepLR scheduler with step size of 50 and learning rate decay of 0.5. We estimate both the error in computing Q_i and the sensitivities $\frac{dQ_i}{d\theta}$ using ℓNF for each $1 \le i \le 4$. The exact time-varying distribution is given by $\varrho(\mathbf{x}, t, \theta) = \frac{e^{2t}}{2\pi} e^{-g(\mathbf{x},t;\theta)}$ where the quadratic form is $g(\mathbf{x}, t; \theta) = \frac{1}{2} e^{2t}(x_1^2 + x_2^2 + x_1^2\theta^2 t^2 - 2x_1 x_2 \theta t)$. The quantities of interest and their derivatives w.r.t θ are thus known exactly:

$$\begin{aligned}
&Q_1(\theta, t^*) = \frac{e^{2t^*}}{2\pi}\int_A e^{-g(\mathbf{x},t^*;\theta)} d\mathbf{x};\\
&\frac{dQ_1}{d\theta} = \frac{e^{2t}}{2\pi}\int_A -e^{-g(\mathbf{x},t^*;\theta)}\frac{dg}{d\theta} d\mathbf{x},\\
&Q_2(\theta, t^*) = e^{-2t^*};\ \frac{dQ_2}{d\theta} = 0,\\
&Q_3(\theta, t^*) = \theta e^{-2t^*} t^*;\ \frac{dQ_3}{d\theta} = e^{-2t^*} t^*,\\
&Q_4(\theta, t^*) = \theta^2 e^{-2t^*} t^{*2} + e^{-2t^*};\ \frac{dQ_4}{d\theta} = 2\theta e^{-2t^*} t^{*2}.
\end{aligned} \tag{24}$$

While this is a linear dynamical system, the QoIs are still non-linear functions of the parameters and/or time, as Eq. (24) show. Moreover, as time evolves, the covariance shrinks making sensitivity calculations challenging. In Table 1, we show the sensitivities and error in computing these QoIs with our estimator ℓNF for $t^* = 1$ and $\theta = 3$.

	Q_1	Q_2	Q_3	Q_4
$\frac{dQ}{d\theta}$	−0.043±1e-2	2e-3±1e-2	0.14±3e-2	0.84±8e-2
True $\frac{dQ}{d\theta}$	−0.046	0	0.135	0.812
e_Q	−9e-3	1.4e-3	1e-2	5e-2

Table 1: Mean $\pm 3\sigma$ estimates using 1k samples and 3 ensemble runs for sensitivity. Error estimates e_Q are point estimates with 1k samples.

5.2 Non-linear Dynamics

In this example we show bimodal behavior in the non-linear *double attractor* system

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \theta x_1 - x_1 x_2 \\ \theta x_1^2 - x_2 \end{bmatrix}, \quad t \in [0,1], \tag{25}$$

with the following initial condition, $\varrho(\boldsymbol{x}, t_0) = \pi(\boldsymbol{x};\boldsymbol{\gamma}) \sim \mathcal{N}(\boldsymbol{\mu}(\boldsymbol{\gamma}), \Sigma)$ where

$$\mu_{2\times 1}(\boldsymbol{\gamma}) = \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix}, \Sigma_{2\times 2} = \begin{bmatrix} 0.75 & 0 \\ 0 & 0.75 \end{bmatrix}. \tag{26}$$

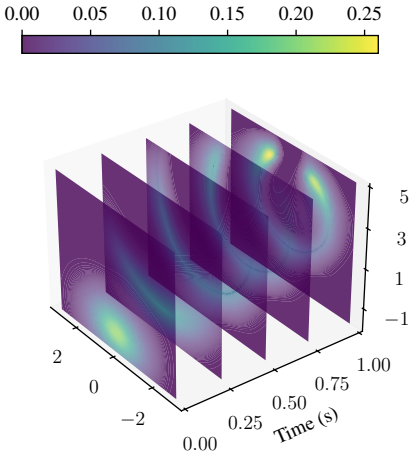

Figure 1: Evolution of the Double Attractor densities via the ℓNF that shows a Gaussian distribution evolving to a bimodal density centered at the two *attractors*.

In Fig. 1 we show the evolution of the initial uncertainty into a bimodal distribution as the dynamics specified by Eq. (25) evolves. This is an extremely challenging flow even though the dimension of state space is small. We consider two QoIs $Q_i = \mathbb{E}[x_i]$ for $1 \le i \le 2$. We use ℓNF hyperparameters as follows: 240 training epochs, 800,000 space-time sample points to estimate the loss function, six act norm / affine coupling layers with each affine layer having 32 neurons, Adam optimizer with a learning rate of 0.001, and a StepLR scheduler with step size of 50 and learning rate decay of 0.7. We set the parameter vector $p = (\theta, \gamma_1, \gamma_2)$ and compute the gradients $\nabla_{\mathbf{p}} Q_1$ and $\nabla_{\mathbf{p}} Q_2$ in Table 2 at the point $(\theta = 2, \gamma_1 = 0, \gamma_2 = -1)$ for this demonstration. This highlights the power of our framework, as the parameter vector can also contain parameters from the initial distribution.

	Q_1	Q_2
$\frac{dQ}{d\theta}$	$-0.03 \pm 6\text{e-}2$	$1.76 \pm 3\text{e-}1$
$\frac{dQ}{d\gamma_1}$	$2.09 \pm 5\text{e-}2$	$0.07 \pm 2\text{e-}1$
$\frac{dQ}{d\gamma_2}$	$0.05 \pm 3\text{e-}1$	$-1.13 \pm 8\text{e-}1$

Table 2: Mean $\pm 3\sigma$ estimates of the sensitivity gradient $\nabla_{\mathbf{p}} Q$ using 1k samples and 3 ensemble runs for Q_1 and Q_2 w.r.t the model parameter θ in Eq. (25) and the statistical parameters γ_1, γ_2 describing the mean of the initial uncertainty in Eq. (26) for the double attractor system.

5.3 Neural ODE Sensitivity Analysis

We consider the Two Moons dataset (as implemented in `scikit-learn` with noise level 0.1) which can be cast as having two classes C_0 and C_1 (corresponding to each moon). We consider a training dataset $\mathcal{D}_{\text{train}}$ and use `torchdyn` [Poli et al.,] to train a two-layer neural ODE classifier $\dot{\boldsymbol{x}} = \boldsymbol{F}_{NN}(\boldsymbol{x};\boldsymbol{\theta})$ (with 16 hidden units and a tanh activation function) to predict whether $\boldsymbol{x} \in C_0$. We train the neural ODE for 200 epochs on 20,000 spatial samples with binary cross-entropy loss utilizing the Adam optimizer with learning rate 0.01 and 20 discrete time steps for the ODE solver. We focus on the quantity of interest $Q(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{x}\in\mathcal{D}_{\text{train}}}[\text{Prob. } (\boldsymbol{x} \in C_0)]$ and compute $\nabla_{\boldsymbol{p}} Q$ where $\boldsymbol{p}$ is the vector of all the NN parameters. First, we train a non-temporal (spatial-only) NF to characterize the initial density corresponding to $\mathcal{D}_{\text{train}}$; the propagation of this initial density through the neural ODE is then characterized by a trained ℓNF with a conditional NSF architecture (3 transforms, 64 features) [Rozet et al., 2022]. The hyperparameters for the ℓNF are 100 training epochs with batch size of 200, 400,000 space-time sample points to estimate the loss function, three neural spline flow layers with 8 bins and 64 neurons each, and Adam optimizer with a learning rate of 0.001. The same hyperparameters were used for training the non-temporal NF for the initial condition but for only 30 epochs. Fig. 2 shows the evolution of densities through the ℓNF trained on the neural ODE classifier trajectories.

To investigate how sensitivities might change under different initial conditions, we consider two additional initial densities, trained on $\mathcal{D}_{test}^{(1)}$ and $\mathcal{D}_{test}^{(2)}$, where these data sets come by rotating the original dataset by 45° and 90°, respectively. In Table 3, we summarize the sensitivities (computed using 1,000 spatial

points and 10 time points) for the two neural ODE layers (each having 32 parameters) across different initial conditions using a ℓ_2 norm. In addition, for the two rotated initial densities, we computed the angle of the sensitivity vectors to the original initial density. It appears that the sensitivity changes on rotated data and generally differs somewhat from the sensitivity on the original data; Appendix D.1 shows the sensitivities for each neuron in each layer.

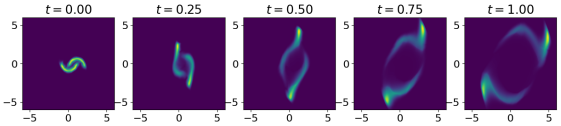

Figure 2: Evolution of the Two Moons initial density via the ℓNF (based on the neural ODE dynamics).

6 Strengths and Limitations

To the best of our knowledge, this is the first result on the exact sensitivity of *arbitrary* QoIs (e.g., probability of events or moments of any order) to arbitrary parameters in a dynamical system with uncertainty. The adjoint-based methods can also be easily extended to stochastic ODEs. Moreover, sensitivity to any QoI w.r.t. to any parameter can be handled in *parallel* within the same framework. Thus, our method works well when the parameter space is large. However, there are some limitations. First, the computable estimates require a space-time estimator in the state space, which can be prohibitively large for many applications. In such cases, the methods developed in this paper could be applied in suitable low-dimensional latent spaces. Second, sensitivity calculations require derivatives of dynamics $\boldsymbol{F}$ w.r.t. the parameters, and while this can be obtained via automatic differentiation, it is certainly not trivial for many complex scientific models in which $\boldsymbol{F}$ comes implicitly from some discretization scheme, thus limiting its applicability.

7 Conclusion

We derive local sensitivities of statistical quantities of interest with respect to the model parameters in dynamical systems whose initial state is unknown and characterized by some probability density function $\pi(\boldsymbol{x})$. Such sensitivity information is an important tool in UQ for dynamical systems, whose applications abound in physical modeling, climate prediction, biological systems, and time series forecasting. We extend classical adjoint-based methods for functions on states of a dynamical system to the space of probability densities on the state and derive the *exact sensitivity evolution* equation for the sensitivity of arbitrary QoIs to any model or statistical parameter of the system. We also develop a neural density estimator via a spatio-temporal normalizing flow and derive computable estimates of both sensitivities and error in computing the QoIs via this estimator. In three examples, we illustrate the theory developed in this paper while highlighting the challenges in computing local sensitivities in uncertain dynamical systems. In future work, we hope to extend this method to real-world applications.

Code availability

The Feynmann Center for Innovation of Los Alamos National Laboratory is currently reviewing our codes for unlimited release. Once approved, we will deposit the codes at https://github.com/lanl/likeprop.

Acknowledgment

The first author acknowledges continued support from Laboratory-Directed Research and Development (LDRD) at Los Alamos National Laboratory. This work was supported by the LDRD project 'Learning Uncertainties In Coupled-Physics Models via Operator Theory' (20230254ER).

T. Butler's work is supported by the National Science Foundation under Grant No. DMS-2208460. T. Butler's work is also supported by NSF IR/D program while working at National Science Foundation. However, any opinion, finding, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

[Ambrosio et al., 2005] Ambrosio, L., Gigli, N., and Savaré, G. (2005). *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media.

[Baehrens et al., 2010] Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., and Müller, K.-R. (2010). How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831.

[Bangerth and Rannacher, 2003] Bangerth, W. and Rannacher, R. (2003). *Adaptive finite element methods for differential equations*. Springer Science & Business Media.

[Bespalov et al., 2014] Bespalov, A., Powell, C. E., and Silvester, D. (2014). Energy norm a posteriori error estimation for parametric operator equations. *SIAM Journal on Scientific Computing*, 36(2):A339–A363.

	Sensitivity Norm			**Angle to** $0°$	
layer	$\|\|0°\|\|_2^2$	$\|\|45°\|\|_2^2$	$\|\|90°\|\|_2^2$	$\angle(45°)$	$\angle(90°)$
1	0.27 (0.012)	0.46 (0.017)	0.18 (0.005)	79.4° (5.1°)	73.5° (4.9°)
2	0.16 (0.008)	0.50 (0.012)	0.22 (0.004)	79.3° (3.8°)	79.6° (4.0°)

Table 3: Sensitivity vector norms estimates for in-distribution (0° rotation) Two Moons initial density compared to rotated (45° or 90° rotation) initial densities and angles between rotated and original sensitivity vectors; reported values are mean (standard error) across random initializations.

[Biloš et al., 2021] Biloš, M., Sommer, J., Rangapuram, S. S., Januschowski, T., and Günnemann, S. (2021). Neural flows: Efficient alternative to neural odes. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W., editors, *Advances in Neural Information Processing Systems*, volume 34, pages 21325–21337. Curran Associates, Inc.

[Bogachev and Ruas, 2007] Bogachev, V. I. and Ruas, M. A. S. (2007). *Measure theory*, volume 1. Springer.

[Butler et al., 2011] Butler, T., Dawson, C., and Wildey, T. (2011). A posteriori error analysis of stochastic differential equations using polynomial chaos expansions. *SIAM Journal on Scientific Computing*, 33(3):1267–1291.

[Cao et al., 2003] Cao, Y., Li, S., Petzold, L., and Serban, R. (2003). Adjoint sensitivity analysis for differential-algebraic equations: The adjoint dae system and its numerical solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089.

[Cao and Petzold, 2004] Cao, Y. and Petzold, L. (2004). A posteriori error estimation and global error control for ordinary differential equations by the adjoint method. *SIAM Journal on Scientific Computing*, 26(2):359–374.

[Chaudhry et al., 2015] Chaudhry, J. H., Estep, D., Ginting, V., Shadid, J. N., and Tavener, S. (2015). A posteriori error analysis of imex multi-step time integration methods for advection–diffusion–reaction equations. *Computer Methods in Applied Mechanics and Engineering*, 285:730–751.

[Chen et al., 2018] Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

[Dijkstra, 2013] Dijkstra, H. A. (2013). *Nonlinear climate dynamics*. Cambridge university press, Cambridge.

[Dinh et al., 2017] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using real NVP. In *International Conference on Learning Representations*.

[Dupont et al., 2019] Dupont, E., Doucet, A., and Teh, Y. W. (2019). Augmented neural odes. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[Durkan et al., 2019] Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019). Neural spline flows. *Advances in Neural Information Processing Systems*, 32.

[Eisenhammer et al., 1991] Eisenhammer, T., Hübler, A., Packard, N., and Kelso, J. A. S. (1991). Modeling experimental time series with ordinary differential equations. *Biological Cybernetics*, 65(2):107–112.

[Estep, 1995] Estep, D. (1995). A posteriori error bounds and global error control for approximation of ordinary differential equations. *SIAM Journal on Numerical Analysis*, 32(1):1–48.

[Feng et al., 2021] Feng, X., Zeng, L., and Zhou, T. (2021). Solving time dependent fokker-planck equations via temporal normalizing flow. *arXiv preprint arXiv:2112.14012*.

[Folland, 1999] Folland, G. B. (1999). *Real analysis: modern techniques and their applications*, volume 40. John Wiley & Sons.

[Furusawa and Kaneko, 2012] Furusawa, C. and Kaneko, K. (2012). A Dynamical-Systems View of Stem Cell Biology. *Science*, 338(6104):215–217.

[Gerritsma et al., 2010] Gerritsma, M., Van der Steen, J.-B., Vos, P., and Karniadakis, G. (2010). Time-dependent generalized polynomial chaos. *Journal of Computational Physics*, 229(22):8333–8363.

[Grathwohl et al., 2018] Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. (2018). Ffjord: Free-form continuous dynamics for scalable reversible generative models.

[Graves et al., 2023] Graves, A., Srivastava, R. K., Atkinson, T., and Gomez, F. (2023). Bayesian flow networks.

[Grätsch and Bathe, 2005] Grätsch, T. and Bathe, K.-J. (2005). A posteriori error estimation techniques in practical finite element analysis. *Computers & Structures*, 83(4):235–265.

[Halder and Bhattacharya, 2011] Halder, A. and Bhattacharya, R. (2011). Dispersion analysis in hypersonic flight during planetary entry using stochastic liouville equation. *Journal of Guidance, Control, and Dynamics*, 34(2):459–474.

[Helton and Davis, 2003] Helton, J. C. and Davis, F. J. (2003). Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety*, 81(1):23–69.

[Karniadakis et al., 2021] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440.

[Kingma and Dhariwal, 2018] Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative Flow with Invertible 1x1 Convolutions. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Klus et al., 2016] Klus, S., Koltai, P., and Schütte, C. (2016). On the numerical approximation of the perron-frobenius and koopman operator. *Journal of Computational Dynamics*, 3(1):51–79.

[Kobyzev et al., 2021] Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2021). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979.

[Lasota and Mackey, 1998] Lasota, A. and Mackey, M. C. (1998). *Chaos, fractals, and noise: stochastic aspects of dynamics*, volume 97. Springer Science & Business Media.

[Liu et al., 2021] Liu, Q., Xu, J., Jiang, R., and Wong, W. H. (2021). Density estimation using deep generative neural networks. *Proceedings of the National Academy of Sciences*, 118(15):e2101344118.

[Lou et al., 2020] Lou, A., Lim, D., Katsman, I., Huang, L., Jiang, Q., Lim, S. N., and De Sa, C. M. (2020). Neural manifold ordinary differential equations. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 17548–17558. Curran Associates, Inc.

[Lusch et al., 2018] Lusch, B., Kutz, N. J., and Brunton, S. L. (2018). Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(4950).

[Magdon-Ismail and Atiya, 1998] Magdon-Ismail, M. and Atiya, A. (1998). Neural networks for density estimation. *Advances in Neural Information Processing Systems*, 11.

[McGoff et al., 2015a] McGoff, K., Mukherjee, S., Nobel, A., and Pillai, N. (2015a). Consistency of maximum likelihood estimation for some dynamical systems. *The Annals of Statistics*, 43(1).

[McGoff et al., 2015b] McGoff, K., Mukherjee, S., and Pillai, N. (2015b). Statistical inference for dynamical systems: A review. *Statistics Surveys*, 9(none).

[McGoff and Nobel, 2020] McGoff, K. and Nobel, A. B. (2020). Empirical risk minimization and complexity of dynamical models. *The Annals of Statistics*, 48(4).

[Norcliffe et al., 2021] Norcliffe, A., Bodnar, C., Day, B., Moss, J., and Liò, P. (2021). Neural ODE processes. *CoRR*, abs/2103.12413.

[Oakley and O'Hagan, 2004] Oakley, J. E. and O'Hagan, A. (2004). Probabilistic sensitivity analysis of complex models: a bayesian approach. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 66(3):751–769.

[Ott et al., 2023] Ott, K., Tiemann, M., and Hennig, P. (2023). Uncertainty and structure in neural ordinary differential equations. *arXiv preprint arXiv:2305.13290*.

[O'Hagan et al., 2013] O'Hagan, A. et al. (2013). Polynomial chaos: A tutorial and critique from a statistician's perspective. *SIAM/ASA J. Uncertainty Quantification*, 20:1–20.

[Papamakarios, 2019] Papamakarios, G. (2019). Neural density estimation and likelihood-free inference. *arXiv preprint arXiv:1910.13233*.

[Papamakarios et al., 2021] Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680.

[Poli et al.,] Poli, M., Massaroli, S., Yamashita, A., Asama, H., Park, J., and Ermon, S. Torchdyn: Implicit models and neural numerical methods in pytorch.

[Ribeiro et al., 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

[Rozet et al., 2022] Rozet, F. et al. (2022). Zuko: Normalizing flows in pytorch.

[Rubanova et al.,] Rubanova, Y., Chen, R. T. Q., and Duvenaud, D. Latent ODEs for Irregularly-Sampled Time Series.

[Saltelli et al., 2004] Saltelli, A., Tarantola, S., Campolongo, F., Ratto, M., et al. (2004). *Sensitivity analysis in practice: a guide to assessing scientific models*, volume 1. Wiley Online Library.

[Santambrogio, 2017] Santambrogio, F. (2017). {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7:87–154.

[Simonyan, 2013] Simonyan, K. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

[Soboĺ, 1993] Soboĺ, I. (1993). Sensitivity estimates for nonlinear mathematical models. *Math. Model. Comput. Exp.*, 1:407.

[Stanley et al., 2024] Stanley, M., Kuusela, M., Byrne, B., and Liu, J. (2024). Technical note: Posterior uncertainty estimation via a Monte Carlo procedure specialized for 4D-Var data assimilation. *Atmospheric Chemistry and Physics*, 24(16):9419–9433.

[Strumbelj and Kononenko, 2010] Strumbelj, E. and Kononenko, I. (2010). An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18.

[Sudret, 2008] Sudret, B. (2008). Global sensitivity analysis using polynomial chaos expansions. *Reliability engineering & system safety*, 93(7):964–979.

[Tang et al., 2022] Tang, K., Wan, X., and Liao, Q. (2022). Adaptive deep density approximation for fokker-planck equations. *Journal of Computational Physics*, 457:111080.

[Wan et al., 2023] Wan, W., Zhang, Y., Bao, C., Dong, B., and Shi, Z. (2023). A scalable deep learning approach for solving high-dimensional dynamic optimal transport. *SIAM Journal on Scientific Computing*, 45(4):B544–B563.

[Winkler et al., 2023] Winkler, C., Worrall, D., Hoogeboom, E., and Welling, M. (2023). Learning likelihoods with conditional normalizing flows.

[Zhao, 2017] Zhao, X.-Q. (2017). *Dynamical Systems in Population Biology*. CMS Books in Mathematics. Springer International Publishing AG, Cham, 2nd ed edition.

A Including Uncertainty In Parameters: Augmented State

To think of parameters as uncertain (e.g., as occurs in a Bayesian Neural Network), we simply derive a Liouville model on the joint space of state and parameters referred to as the augmented state space.

Definition A.1 (Augmented State Space)**.** We define the augmented state space $\mathcal{S}$ as the subset of $\mathbb{R}^d \times \mathbb{R}^p$ given by the Cartesian product

$$\mathcal{S} = \mathcal{X} \times \Theta.$$

Therefore, if $\boldsymbol{x}(t)$ and $\boldsymbol{\theta}(t)$ are the joint state and the joint parameters at time t, respectively, the augmented state at time t, $\boldsymbol{s}(t) \in \mathcal{S}$, is given by the vector $\boldsymbol{s}(t) = (\boldsymbol{x}(t), \boldsymbol{\theta}(t))$.

Since the parameters are stationary, we can describe the evolution of the augmented state by the following coupled system of ODEs,

$$\dot{\boldsymbol{s}} = \boldsymbol{J}(\boldsymbol{s}) \tag{27}$$

where $\boldsymbol{J}(\boldsymbol{x}, \boldsymbol{\theta}) = (\boldsymbol{F}(\boldsymbol{x}, \theta), \boldsymbol{0})$ and,

$$\frac{dx_i}{dt} = f_i(\boldsymbol{x}, \boldsymbol{\theta}) \tag{28a}$$

$$\frac{d\theta_j}{dt} = 0 \tag{28b}$$

for $1 \leq i \leq d$ and $1 \leq j \leq p$ with $\boldsymbol{F} = (f_1, \cdots, f_d)$.

The conceptual framework, theory, and algorithms described within the paper all directly apply to $\mathcal{S}$ replacing $\mathcal{X}$ with marginal or conditional densities utilized whenever specific results are desired over only the state or parameter space.

B Proofs of All Theorems

We prove all theorems here. Many of the equations in these proofs are long and involve several integrals that are, at times, broken up across several lines due to linewidth constraints. Therefore, we generally omit the differentials $d\boldsymbol{x}$ or dt involved with each integral since the variables and order of integration are always clear from the context of the domains over which the various integrals are computed.

The Liouville equation can be derived from measure theoretic results as in [Lasota and Mackey, 1998]. The same equation is valid for the augmented case when parameters are considered stationary.

Let $\boldsymbol{p} = (p_1, \cdots, p_i, \cdots p_N)$ be the parameter vector (model parameters and vectorized statistical parameters). Let p_i be a parameter of interest with $Q(t, \boldsymbol{p})$ denoting the QoI at time t as defined in Definition 3.1. We now derive the sensitivity equation for Q_{p_i} as summarized in Theorem 3.3. This proof is long so we present it in 3 steps to aid readability. In Step 1, we derive the formal adjoint Liouville operator $L^\dagger[\cdot]$. This step utilizes integration by parts, which of course requires the implicit assumption that either $\mathcal{X}$ is bounded or the functions involved decay at a sufficient rate for the appropriate Green's theorem to apply. In Step 2, we derive the evolution equation for the sensitivity of the density to the parameter $s_i = \frac{\partial \varrho}{\partial p_i}$. Finally, in Step 3, we complete the derivation of the sensitivity equation for Q_{p_i}.

Proof of Theorem 3.3. **Step 1: Derive Formal Adjoint Operator $L^\dagger$.** Let $L[\cdot]$ be the Liouville operator given by $L[\cdot] = \frac{\partial [\cdot]}{\partial t} + \text{div}([\cdot]\boldsymbol{F})$. Let us define the following bilinear form on all suitable functions of the state space $\mathcal{X}$ for some fixed $t^* > t_0$,

$$B(v, w) := (v, L[w]) := \int_{t_0}^{t^*} \int_{\mathcal{X}} v \left[\frac{\partial w}{\partial t} + \text{div}(w\boldsymbol{F}) \right], \tag{29}$$

We utilize *integration by parts* to define the formal adjoint of the Liouville operator L w.r.t. to the bilinear form

above.

$$\begin{aligned}(v, L[w]) &= \int_{t_0}^{t^*}\int_{\mathcal{X}} v\left[\frac{\partial w}{\partial t} + \operatorname{div}(w\boldsymbol{F})\right] \\ &= \int_{t_0}^{t^*}\int_{\mathcal{X}}\left[\frac{\partial vw}{\partial t} - w\frac{\partial v}{\partial t}\right] + \int_{t_0}^{t^*}\int_{\mathcal{X}} v\operatorname{div}(w\boldsymbol{F}) \\ &= \int_{t_0}^{t^*}\int_{\mathcal{X}}\left[\frac{\partial (vw)}{\partial t} - w\frac{\partial v}{\partial t}\right] + \int_{t_0}^{t^*}\int_{\partial\mathcal{X}} vw\langle \boldsymbol{F}, \boldsymbol{n}\rangle - \int_{t_0}^{t^*}\int_{\mathcal{X}} w\langle \operatorname{grad} v, \boldsymbol{F}\rangle \\ &= -\int_{t_0}^{t^*}\int_{\mathcal{X}} w\left[\frac{\partial v}{\partial t} + \langle \operatorname{grad} v, \boldsymbol{F}\rangle\right] + \mathrm{BT}, \end{aligned} \tag{30}$$

where $\boldsymbol{n}$ denotes the outward pointing normal vector to $\partial\mathcal{X}$, and the *boundary term* BT is given by

$$\int_{\mathcal{X}} (v(t^*)w(t^*) - v(t_0)w(t_0)) + \int_{t_0}^{t^*}\int_{\partial\mathcal{X}} vw\langle \boldsymbol{F}, \boldsymbol{n}\rangle.$$

Thus, we define $L^\dagger[v]$ as

$$L^\dagger[v] := \left[-\frac{\partial v}{\partial t} - \langle \operatorname{grad} v, \boldsymbol{F}\rangle\right],$$

which yields

$$(v, L[w]) = \left(L^\dagger[v], w\right) + \mathrm{BT}.$$

Step 2: Derive the Evolution Equation of $s_i = \frac{\partial \varrho}{\partial p_i}$. Let $s_i(\boldsymbol{x}, t) = \partial\varrho(\boldsymbol{x}, t)/\partial p_i$. Observe that,

$$\frac{\partial L[\varrho]}{\partial p_i} = \frac{\partial s_i}{\partial t} + \operatorname{div}(s_i\boldsymbol{F}) + \operatorname{div}\left(\varrho\frac{\partial \boldsymbol{F}}{\partial p_i}\right).$$

Thus,

$$\frac{\partial L[\varrho]}{\partial p_i} = L[s_i] + \operatorname{div}\left(\varrho\frac{\partial \boldsymbol{F}}{\partial p_i}\right). \tag{31}$$

Since $L[\varrho] = 0$ (a.e on $\mathcal{X}$), we get the *evolution* equation of sensitivity with suitable initial and boundary conditions (using $\boldsymbol{F}_{p_i}$ to mean $\frac{\partial \boldsymbol{F}}{\partial p_i}$),

$$\frac{\partial s_i}{\partial t} + \operatorname{div}(s_i\boldsymbol{F}) = -\operatorname{div}\left(\varrho\boldsymbol{F}_{p_i}\right) \tag{32}$$

Step 3: Derive the Adjoint-Based Sensitivity Equation to Compute Q_{p_i}. In order to compute sensitivity of Q w.r.t. p_i, we first consider the following scalar equation, h, defined as

$$h(v, \boldsymbol{p}) := Q(t^*, \boldsymbol{p}) - B(v, \varrho(t^*, \boldsymbol{p})), \tag{33}$$

for some suitable function v on the state space (that we are free to choose) since substituting $w = \varrho$ in Equation (29) implies $B(v, \varrho) = 0$. Using the compact notation for partial derivatives and the definition of $B(v, \varrho)$ in Equation (29) yields,

$$\begin{aligned} h_{p_i} &= Q_{p_i}(t^*, \boldsymbol{p}) - \frac{\partial}{\partial p_i} B(v, \varrho(t^*, \boldsymbol{p}) \\ &= Q_{p_i}(t^*, \boldsymbol{p}) - \int_{t_0}^{t^*}\int_{\mathcal{X}} v\frac{\partial L[\varrho]}{\partial p_i} \end{aligned} \tag{34}$$

Using Eq. (31) gives

$$h_{p_i} = Q_{p_i}(t^*, \boldsymbol{p}) - \int_{t_0}^{t^*}\int_{\mathcal{X}} vL[s_i] - \int_0^{t^*}\int_{\mathcal{X}} v\operatorname{div}(\varrho\boldsymbol{F}_{p_i}). \tag{35}$$

Utilizing Eq. (7) in conjunction with the definition of $L^\dagger$, we rewrite this as

$$h_{p_i} = Q_{p_i}(t^*, \boldsymbol{p}) - \int_{t_0}^{t^*}\int_{\mathcal{X}} s_iL^\dagger[v] - \int_{t_0}^{t^*}\int_{\mathcal{X}} v\operatorname{div}(\varrho\boldsymbol{F}_{p_i}) - \mathrm{BT}, \tag{36}$$

where BT is now given by

$$\text{BT} = \int_{\mathcal{X}} (v(t^*)s_i(t^*) - v(t_0)s_i(t_0)) + \int_0^{t^*} \int_{\partial\mathcal{X}} v s_i \langle \boldsymbol{F}, \boldsymbol{n} \rangle. \tag{37}$$

Since

$$Q(t^*, \boldsymbol{p}) = \int_{\mathcal{X}} g(\boldsymbol{x}(t^*)) \varrho(\boldsymbol{x}, t, \boldsymbol{p}),$$

we have

$$Q_{p_i}(t^*, \boldsymbol{p}) = \int_{\mathcal{X}} g(\boldsymbol{x}(t^*)) s_i(t^*). \tag{38}$$

We now observe that since $h = Q - B(v, \varrho)$ and the latter is 0 for all suitable v, we also have $h_{p_i} = Q_{p_i}$. It follows that

$$Q_{p_i}(t^*, \boldsymbol{p}) = \int_{\mathcal{X}} g(\boldsymbol{x}(t^*)) s_i(t^*) - \int_{t_0}^{t^*} \int_{\mathcal{X}} s_i L^\dagger[v] - \int_{t_0}^{t^*} \int_{\mathcal{X}} v \operatorname{div}(\varrho \boldsymbol{F}_{p_i}) - \text{BT} \tag{39}$$

Substituting in the explicit form of BT, we re-organize the terms as

$$\begin{aligned} Q_{p_i}(t^*, \boldsymbol{p}) = & \underbrace{- \int_{t_0}^{t^*} \int_{\mathcal{X}} v \operatorname{div}(\varrho \boldsymbol{F}_{p_i}) + \int_{\mathcal{X}} v(t_0) s_i(t_0)}_{\text{Term } \mathbf{I}} \\ & \underbrace{\int_{\mathcal{X}} (g(\boldsymbol{x}(t^*)) - v(t^*)) s_i(t^*) - \int_{t_0}^{t^*} \int_{\mathcal{X}} s_i L^\dagger[v] - \int_{t_0}^{t^*} \int_{\partial\mathcal{X}} v s_i \langle \boldsymbol{F}, \boldsymbol{n} \rangle}_{\text{Term } \mathbf{II}} \end{aligned} \tag{40}$$

Term **II** is made zero by choosing v as the solution to the following adjoint problem backward in time, i.e.

$$L^\dagger[v] = 0, \tag{41}$$

with initial condition $v(t^*) = g(\boldsymbol{x}(t^*))$ and boundary condition, $v = 0$ on $\partial\mathcal{X}$ for each t in (t_0, t^*). It follows that Term **I** with this same v defines the sensitivity of Q. □

We now prove Theorem 3.5.

Proof of Theorem 3.5. Let $e_\varrho = \varrho - \widehat{\varrho}$. Multiplying both sides of $L^\dagger[v] = 0$ by e_ϱ and following the same steps (in reverse) shown in Eq. (30) for integrating by parts across both state space and time yields,

$$0 = \int_{t_0}^{t^*} \int_{\mathcal{X}} e_\varrho L^\dagger[v] = - \int_{t_0}^{t^*} \int_{\mathcal{X}} v L[e_\varrho] + \int_{\mathcal{X}} \left[v(0) e_\varrho(0) - v(t^*) e_\varrho(t^*) \right]. \tag{42}$$

Above, the boundary conditions for v as shown in the adjoint equation of Eq. (6) eliminate the integral over $\partial\mathcal{X}$ that is present in the BT term within Eq. (30). Using the definitions of the true and approximate QoIs in Eq. (4) and Eq. (8), respectively, along with the fact that $v(t^*) = g(\boldsymbol{x}(t^*))$, we rewrite the above equation as

$$e_Q := \int_{\mathcal{X}} g(x(t^*)) e_\varrho(t^*) = \int_{t_0}^{t^*} \int_{\mathcal{X}} v L[e_\varrho] + \int_{\mathcal{X}} v(0) e_\varrho(0). \tag{43}$$

Finally, using Eq. (2) (which gives $L[\varrho] = 0$) along with the linearity of the operator L gives

$$e_Q = - \int_{t_0}^{t^*} \int_{\mathcal{X}} v L[\widehat{\varrho}(\boldsymbol{x}, t)] d\boldsymbol{x}\, dt + \int_{\mathcal{X}} v(0) e_\varrho(0) \tag{44}$$

The conclusion follows by identifying $e_\varrho(0) = \pi(\boldsymbol{x}) - \varrho(\boldsymbol{x}, t_0)$. □

Proof of Theorem 3.7. The first convergence results in this theorem follow from the classic Weak Law of Large Numbers.

To prove the primary convergence results given in equation 16, we first observe that all of the integrands over the state space $\mathcal{X}$ with respect to the Lebesgue measure (denoted by $d\boldsymbol{x}$) in equation 9 and equation 10 involve the function $v(\boldsymbol{x}, t)$. Multiplying each of the integrands by $\frac{\widehat{\rho}(\cdot,t)}{\widehat{\rho}(\cdot,t)}$ allows us to rewrite each of these integrals as functions involving $w(\cdot, t) := \frac{v(\cdot,t)}{\widehat{\varrho}(\cdot,t)}$ with respect to the probability measure associated with $\widehat{\rho}$. Thus, for example,

$$G_1^n(t) \to \mathbb{E}_{\boldsymbol{x}\sim\widehat{\varrho}(\cdot,t)}[w(\cdot,t)\,\mathrm{div}(\widehat{\varrho}\boldsymbol{F}_{p_i})] = \int_{\mathcal{X}} v(\boldsymbol{x},t)\,\mathrm{div}(\widehat{\varrho}\boldsymbol{F}_{p_i})d\boldsymbol{x}.$$

Then, the boundedness assumption allows us to combine the first convergence results in this theorem with the Dominated Convergence Theorem to obtain equation 16. Thus, e.g.

$$\int_{t_0}^{t^*} G_1^n(t) \to \int_{t_0}^{t^*}\int_{\mathcal{X}} v(\boldsymbol{x},t)\,\mathrm{div}(\widehat{\varrho}\boldsymbol{F}_{p_i})d\boldsymbol{x}.$$

Since

$$G_2^n \to \mathbb{E}_{\boldsymbol{x}\sim\widehat{\varrho}(\cdot,t_0)}\left[w(\cdot,t_0)\frac{\partial\pi}{\partial p_i}\right],$$

combining the two (via the Continuous Mapping Theorem) we get

$$\left(G_2^n - \int_{t_0}^{t^*} G_1^n(t)\right) \to \widehat{Q_{p_i}}(t^*).$$

Exactly the same argument will lead to the other convergence result in equation 16.

□

B.1 Reverse KL

In order to use reverse KL we will show Theorem B.2. We first gather some useful results just to establish a common notation including the formal definition of the *pushforward* measure.

Definition B.1 (Pushforward Measure)**.** Let $\mathcal{X}_1$, equipped with an appropriate σ-algebra, be a measure space with measure μ_1. Let $\mathcal{X}_2$, equipped with an approrpiate σ-algebra, be a measurable space and T be a measurable map from $\mathcal{X}_1$ to $\mathcal{X}_2$. Then, T induces a measure μ_2 on $\mathcal{X}_2$ called the *pushforward* measure of T defined as

$$\mu_2[\cdot] := \mu_1[T^{\mathrm{pre}}[\cdot]],$$

for any measurable subset $[\cdot]$ of $\mathcal{X}_2$. We typically denote the pushforward measure μ_2 as $T_{\sharp\mu_1}$.

Pushforward measures behave nicely w.r.t. integration. In particular, they allow for a simple change of measure to occur. For instance, if f is a real valued integrable function on $\mathcal{X}_2$, then we have the following useful result,

$$\int_{\mathcal{X}_2} f d\mu_2 = \int_{\mathcal{X}_1} f\circ T d\mu_1. \tag{45}$$

In the particular case that T is a diffeomorphism and $\mathcal{X}_1 = \mathcal{X}_2 = \mathbb{R}^d$ and the measure μ_1 is absolutely continuous w.r.t. a probability measure associated with density function $p_{\mathcal{Z}}$, the pushforward measure μ_2 is also absolutely continuous w.r.t. to a probability density function $p_{\mathcal{X}}$ that is given by the standard change of variables formula,

$$\mu_2 \sim p_{\mathcal{X}}(\boldsymbol{x}) = p_{\mathcal{Z}}\left(T^{-1}(\boldsymbol{x})\right)\left|\det J_{T^{-1}}(T^{-1}\boldsymbol{x})\right| = \frac{p_{\mathcal{Z}}\left(T^{-1}(\boldsymbol{x})\right)}{\left|\det J_T(T^{-1}\boldsymbol{x})\right|}. \tag{46}$$

Using the result in Eq. (45), we get the following (called law of the unconscious statistician)

$$\mathbb{E}_{p_{\mathcal{X}}}[f] = \mathbb{E}_{p_{\mathcal{Z}}}[f\circ T] \tag{47}$$

We now prove Theorem B.2.

Theorem B.2 ((Reverse) KL Divergence)**.** *Let $T_{[\phi_{\mathrm{NF}},t]} : \mathcal{X} \to \mathcal{X}$ be a diffeomorphic map parameterized by the time indexed set $\{\phi_{\mathrm{NF}}\}$ whose pushforward density on the standard base density $p_{\mathcal{Z}}(\boldsymbol{z})$ on $\mathbb{R}^d$ is given $\widehat{\varrho}$. Then, for a fixed time $t^* \in (t_0, t_f)$*

$$D_{\mathrm{KL}}\left(\widehat{\varrho}(\boldsymbol{x}, t^*) \,||\, \varrho(\boldsymbol{x}, t^*)\right) := \mathbb{E}_{p_{\mathcal{Z}}}\left[\log p_{\mathcal{Z}} - \log\left|\det J_{T_{[\phi_{\mathrm{NF}},t^*]}}\right| - \log \varrho \circ T_{[\phi_{\mathrm{NF}},t^*]}\right] \tag{48}$$

where $\varrho(T_{[\phi_{NF},t^*]}(\boldsymbol{z}))$ is computed by cycling single backward and forward ODEs in Algorithm 2.

Proof of Theorem B.2. Let t^* be a fixed time in (t_0, t_f). Let $p_{\mathcal{Z}}(\boldsymbol{z})$ be a base density on $\mathbb{R}^d$ and let $\widehat{\varrho}(\boldsymbol{x}, t^*)$ be the corresponding pushforward density of the time indexed normalizing flow $T_{[\phi_{\mathrm{NF}},t]}$ defined on the state space $\mathcal{X}$. Let $\varrho(\boldsymbol{x}, t^*; \gamma, \boldsymbol{\theta})$ be the probability density on $\mathbb{R}^d$ at time t^* given by the Liouville equation. Using the definition of KL divergence we have,

$$D_{\mathrm{KL}}\left(\widehat{\varrho}(\boldsymbol{x}, t^*) \,||\, \varrho(\boldsymbol{x}, t^*)\right) = \mathbb{E}_{\widehat{\varrho}}\left[\log \widehat{\varrho} - \log \varrho\right]. \tag{49}$$

Using Equation (47) we can write the above expectation as the expectation over the base density

$$\mathbb{E}_{\widehat{\varrho}}\left[\log \widehat{\varrho} - \log \varrho\right] = \mathbb{E}_{p_{\mathcal{Z}}}\left[\log \widehat{\varrho} \circ T_{[\phi_{\mathrm{NF}},t]} - \log \varrho \circ T_{[\phi_{\mathrm{NF}},t]}\right]. \tag{50}$$

Using the change of variables for the pushforward density as in Equation (46) with $p_{\mathcal{X}}$ given by $\widehat{\varrho}$ and T given by $T_{[\phi_{\mathrm{NF}},t]}$, we can write

$$\begin{aligned}
(\log \widehat{\varrho} \circ T_{[\phi_{\mathrm{NF}},t]})(\boldsymbol{z}) &= \log\left(\widehat{\varrho}(T_{[\phi_{\mathrm{NF}},t]}(\boldsymbol{z}))\right) \\
&= \log\left(\frac{p_{\mathcal{Z}}\left(T^{-1}_{[\phi_{\mathrm{NF}},t]}(T_{[\phi_{\mathrm{NF}},t]}\boldsymbol{z})\right)}{\left|\det J_T(T^{-1}_{[\phi_{\mathrm{NF}},t]}(T_{[\phi_{\mathrm{NF}},t]}\boldsymbol{z}))\right|}\right) \\
&= \log\left(p_{\mathcal{Z}}(\boldsymbol{z})\right) - \log\left(J_{T_{[\phi_{\mathrm{NF}},t]}}(\boldsymbol{z})\right)
\end{aligned} \tag{51}$$

This gives us Equation (48).

Observe that we can rewrite the Liouville equation Eq. (2) in *non-conservative* form as

$$\frac{\partial \varrho}{\partial t} + \langle \mathrm{grad}\varrho, \boldsymbol{F}\rangle = -\varrho \,\mathrm{div}\, \boldsymbol{F}. \tag{52}$$

Let $\phi(\boldsymbol{x}(t), t)$ denote the flow of $\boldsymbol{x}$ given by the ODE in Eq. (1). By this we mean that if $\boldsymbol{x} = \boldsymbol{x}_0$ at $t = t_0$, then the curve $t \mapsto \boldsymbol{x}(t)$ as $\boldsymbol{x}_0$ evolves according to the ODE in Eq. (1) is the flow $\phi(\boldsymbol{x}(t), t)$ starting from t_0 to some time t. At $t = 0$, we know $\varrho(\boldsymbol{x}_0, t_0)$. This is just the value of the prescribed initial density $\pi(\boldsymbol{x}_0; \gamma, \boldsymbol{\theta})$. As $\boldsymbol{x}_0$ evolves, the value of the density at a given time t is the scalar

$$\ell_\phi(t) = \varrho(\boldsymbol{x}(t), t; \gamma, \boldsymbol{\theta}). \tag{53}$$

It turns out, this evolution is given by the coupled system of ODEs from the characteristics of the Liouville equation. Taking the time derivative of this equation Eq. (53) we get,

$$\begin{aligned}
\frac{d\ell_\phi}{dt} &= \frac{\partial \varrho}{\partial t}\Big|_{\boldsymbol{x}(t)} + \frac{\partial \varrho}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}(t)} \frac{d\boldsymbol{x}}{dt} \\
&= \frac{\partial \varrho}{\partial t}|_{\boldsymbol{x}(t)} + \langle \mathrm{grad}\varrho|_{\boldsymbol{x}(t)}, \boldsymbol{F}\rangle \\
&= -\varrho|_{\boldsymbol{x}(t)} \mathrm{div}\, \boldsymbol{F} \quad \text{using Equation (52)}.
\end{aligned}$$

Since $\varrho|_{\boldsymbol{x}(t)} = \ell_\phi(t)$, we get

$$\frac{d\ell_\phi}{dt} = -\ell_\phi(t) \,\mathrm{div}\, \boldsymbol{F}. \tag{54}$$

Thus, the above equation along with $\dot{\boldsymbol{x}} = \boldsymbol{F}$ gives us Equation (53) in Algorithm 2. In other words if we know an initial state and we follow its evolution we also follow the evolution of $\varrho(\boldsymbol{x}(t), t)$ starting from its

initial likelihood $\pi(\boldsymbol{x}_0;\gamma)$. In order to complete Theorem B.2 we need to find the appropriate initial state for a state vector $(\boldsymbol{x}_i, t^*) = T_{[\phi_{\text{NF}},t^*]}(\boldsymbol{z}_i)$ for some suitable $\boldsymbol{z}_i$ sampled from the standard Gaussian. This is given by backward evolving $(\boldsymbol{x}_i, t^*)$ through the ODE Equation (1). Hence, we call this the cycle algorithm, we first evolve $(\boldsymbol{x}_i, t^*) = T_{[\phi_{\text{NF}},t^*]}(\boldsymbol{z}_i)$ backwards and find its initial state and then evolve its corresponding likelihood forward. □

Algorithm 2 Cycle Algorithm: Compute $\ell_\phi(t^*) = \varrho(\boldsymbol{x}, t^*; \gamma, \boldsymbol{\theta})$

1: Given $\boldsymbol{x}(t^*) = T_{[\phi_{\text{NF}},t^*]}(\boldsymbol{z})$ at t^* for some $\boldsymbol{z} \sim p_{\mathcal{Z}}$
2: Backward time integrate $\dot{\boldsymbol{x}} = \boldsymbol{F}$ with initial condition $\boldsymbol{x}(t = t^*)$ to $\boldsymbol{x}(t = t_0)$ Flow of $\boldsymbol{x}(t^*) : \phi(\boldsymbol{x}(t), t_0)$ for $[t_0, t^*]$
3: Forward time integrate

$$\frac{d\ell_\phi(t)}{dt} = -\ell_\phi(t)\,\text{div}(\boldsymbol{F}), \quad \dot{\boldsymbol{x}} = \boldsymbol{F}, \quad ICs : \ell_\phi(t_0) = \pi(\boldsymbol{x}(t_0);\gamma) \text{ and } \boldsymbol{x}(t = t_0) \tag{55}$$

$\varrho(\boldsymbol{x}, t^*; \gamma, \boldsymbol{\theta}) \leftarrow \ell_\phi(t^*)$

B.2 Computing The Adjoint Solution

In this section we show how to compute the adjoint solution $v(\boldsymbol{x}, t)$ given by *backward* evolution of Eq. (6) $L^\dagger[v] = 0$ starting with $v(\boldsymbol{x}, t = t^*) = g(\boldsymbol{x})$, i.e.

$$\begin{aligned} &\frac{\partial v}{\partial t} + \langle \text{grad}\,[v], \boldsymbol{F}\rangle = 0 \ \text{ on } \mathcal{X} \times (t_0, t^*], \quad v(\boldsymbol{x}, t^*) = g(\boldsymbol{x}(t^*)) \ \text{ on } \mathcal{X}, \\ &\quad v(\cdot, t) = 0 \ \text{ on } \partial\mathcal{X} \times [t_0, t^*). \end{aligned} \tag{56}$$

Let us revisit the *flow* $\phi(\boldsymbol{x}(t), t)$ of $\boldsymbol{x}$ given by the ODE in Eq. (1) described above which is the curve $t \mapsto \boldsymbol{x}(t)$ as $\boldsymbol{x}_0$ evolves according to the ODE in Eq. (1) from t_0 to some time t. If we look at the adjoint solution on this flow, $v(\boldsymbol{x}(\boldsymbol{t}), t)$

$$\begin{aligned} \frac{dv}{dt} &= \frac{\partial v}{\partial t}\Big|_{\boldsymbol{x}(t)} + \frac{\partial v}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}(t)} \frac{d\boldsymbol{x}}{dt} \\ &= \frac{\partial v}{\partial t}\Big|_{\boldsymbol{x}(t)} + \langle \text{grad} v|_{\boldsymbol{x}(t)}, \boldsymbol{F}\rangle \\ &= L^\dagger[v]|_{\boldsymbol{x}(t)} \\ &= 0. \end{aligned}$$

Thus along a flow $v =$ const, that is, $v(\boldsymbol{x}(t), t) = v(\boldsymbol{x}(t^*), t^*) = g(\boldsymbol{x}(t^*))$. Hence, if $(\boldsymbol{x}_i, t_i)$ is a point in $\mathcal{X} \times [t_0, t^*)$, to compute $v(\boldsymbol{x}_i, t_i)$, we first evolve $\boldsymbol{x}_i$ from t_i to t^* using our dynamics Eq. (1) to get $\boldsymbol{x}_i^*$ at t^* and then use $v(\boldsymbol{x}_i, t_i) = g(\boldsymbol{x}_i^*)$.

C Normalizing Flow Architcture

We present the architecture as implemented in our GitHub library. The net consists of alternating Actnorm layers and affine mapping layers (both $\mathcal{X} \times [t_0, t_f] \to \mathcal{Z} \times [0, 1]$ functions) for capturing nonlinear time-dependence. As in [Feng et al., 2021, Kingma and Dhariwal, 2018], an Actnorm layer (for stability) f introduces trainable scale and bias parameters α_i, β_i for the transformation

$$f(x_i, t) = \alpha_i \odot x_i + \beta_i.$$

An affine coupling layer g [Dinh et al., 2017] partitions the spatial component $\mathcal{X}$ of the inputs x into two, $(x^1, x^2) \in \mathcal{X}_1 \times \mathcal{X}_2$. Then $g(x_i, t) = (g_1(x_i, t), g_2(x_i, t))$ where

$$\begin{cases} g_1(x, t) &= x^1 \\ g_2(x, t) &= x^2 \odot (1 + \gamma \tanh(\delta(x^1, t))) + e^\epsilon \odot \tanh(\zeta(x^1, t)), \end{cases}$$

the $\gamma, \epsilon \in \mathbb{R}$ and the $\delta, \zeta \in \mathrm{NN}_k : \mathcal{X} \times [t_0, t_1] \to \mathcal{Z} \times [0, 1]$ a family of k-deep neural networks. Further details on the inversion and differentiation of the layers f and g can be found in [Feng et al., 2021]. Finally, a polynomial spline layer (see [Tang et al., 2022]) is employed after the Actnorm and affine mapping layers.

C.1 Loss function

Draw N_b samples from the base density $p_{\mathcal{Z}\times\mathrm{Unif}([0,1]}$ for $b = 1, \ldots, B$ to comprise training batches $D_b = \{(\boldsymbol{x}_i, t_i) : i = 1, \ldots, N_b\}$. Likewise, draw $N_{\mathrm{init},b}$ samples from π to comprise training batches for the initial condition, $D_{\mathrm{init},b} = \{(\boldsymbol{x}_{\mathrm{init},i}, 0) : i = 1, \ldots, N_{\mathrm{init},b}\}$. For our numerical examples, we focus on the PINN loss function,

$$\mathcal{L}(\phi_{\mathrm{NF}}, \boldsymbol{\lambda}; t^*) := \mathbb{E}_{(\boldsymbol{x},t)\sim\widehat{\rho}(\boldsymbol{x},t^*)} \left[L[\widehat{\varrho}(\boldsymbol{x}, t; \phi_{\mathrm{NF}})]\right]^2 + \underbrace{\mathbb{E}_{\boldsymbol{x}\sim\rho(\boldsymbol{x},0)} \|\widehat{\varrho}(\boldsymbol{x}, 0; \phi_{\mathrm{NF}}) - \pi(\boldsymbol{x}_{\mathrm{init},i})\|_2}_{\mathcal{L}_{\mathrm{init}}}, \text{ or} \tag{57}$$

$$\widehat{\mathcal{L}}(\phi_{\mathrm{NF}}, \boldsymbol{\lambda}; t^*) = \frac{1}{N_b} \sum_{i=1}^{N_b} \left[L[\widehat{\varrho}(\boldsymbol{x}_i, t_i; \phi_{\mathrm{NF}})]\right]^2 + \frac{1}{N_{\mathrm{init},b}} \sum_{i=1}^{N_{\mathrm{init},b}} \|\widehat{\varrho}(\boldsymbol{x}_{\mathrm{init},i}, 0; \phi_{\mathrm{NF}}) - \pi(\boldsymbol{x}_{\mathrm{init},i})\|_2; \tag{58}$$

the term $\mathcal{L}_{\mathrm{init}}$ is now included to ensure that the trained tNF solves the continuity equation for the given initial condition π.

D Results

D.1 Two Moons results

We present more detail on the out of distribution sensitivities calculated for the Two Moons example. Fig. 3 shows the calculated sensitivites for each neuron the first layer (top) and second layer (bottom) for the in distribution data compared to the OOD data rotated at 45° and 90°. While for some neurons, the sensitivity is similar for in distribution and out of distribution data, for others, there are differences in sensitivity.

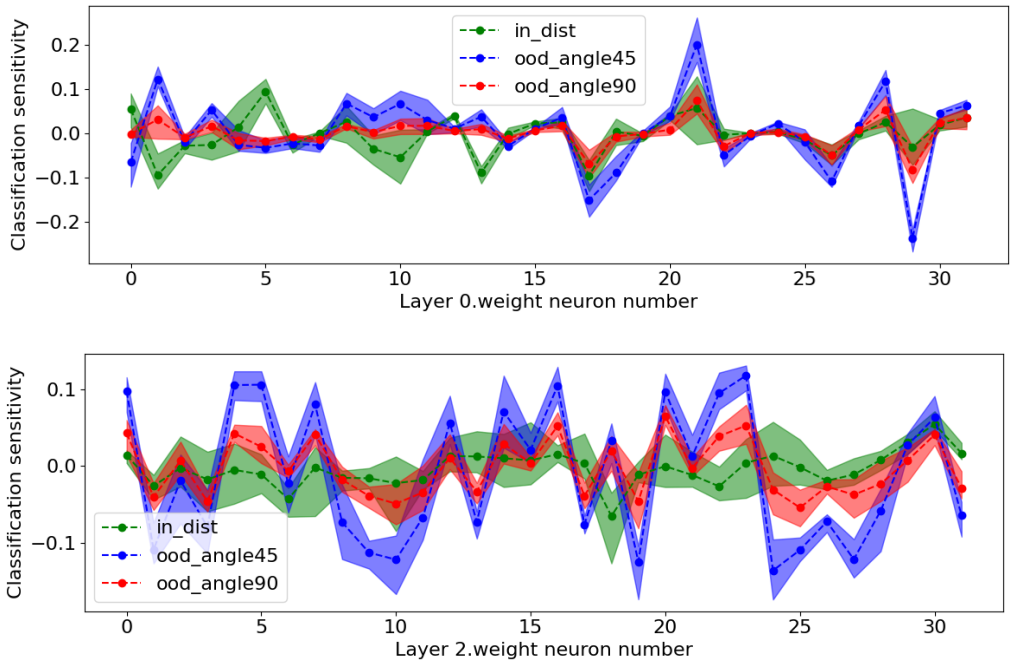

Figure 3: Calculated sensitivities for Two Moons example for first layer (top) and second layer (bottom) of the Neural ODE for in distribution data and OOD data (rotated by 45° and 90°). Dashed lines indicate mean sensitivity values across an ensemble of estimators, while shaded areas indicate plus or minus one standard error. There are clear differences in sensitivity between in-distribution and OOD data wich are particularly pronounced in layer 2.