A novel approach to quantify out-of-distribution uncertainty in Neural and Universal Differential Equations

Stefano Giampiccolo[1,2], Giovanni Iacca[2], Luca Marchetti[1,3]

[1]Fondazione The Microsoft Research-University of Trento Centre for Computational and Systems Biology (COSBI), Rovereto, Trento, Italy.
[2]Department of Information Engineering and Computer Science (DISI), University of Trento, Povo, Trento, Italy.
[3]Department of Cellular, Computational and Integrative Biology (CIBIO), University of Trento, Povo, Trento, Italy.

Contributing authors: giampiccolo@cosbi.eu; marchetti@cosbi.eu;

Abstract

Dynamical systems play a central role across the quantitative sciences, offering a powerful mathematical framework to describe, analyze, and predict the evolution of complex processes over time. In Systems Biology, dynamical systems provide a foundation for modeling and predicting the intricate behaviors of biological systems. Recent advances in data-driven approaches, such as Neural Ordinary Differential Equations (NODEs) and Universal Differential Equations (UDEs), have enabled the development of models that are either fully or partially data-driven. Integrating data-driven components into dynamical systems amplifies the challenge of generalization beyond training data, highlighting the need for robust methods to quantify uncertainty in scenarios not encountered during training. In this work, we investigate the reliability of uncertainty quantification (UQ) based on ensembles of models in the reconstruction of dynamical systems. We show that standard ensembles (i.e., models trained independently with different random initializations) risk producing overconfident predictions in previously unseen scenarios, as the models in the ensemble tend to exhibit similar behaviors. To address this issue, we propose a novel ensemble construction method for NODEs and UDEs that fosters diversity in the reconstructed vector field across models within specific regions of the state space, while maintaining explicit control over the fit on the training set. We evaluate our method on synthetic test cases derived from three models commonly used as benchmarks for data-driven reconstruction

of dynamical systems: the Lotka-Volterra model, the damped harmonic oscillator, and the Lorenz system. Our results demonstrate that the proposed method enhances the reliability of UQ in previously unseen scenarios with respect to standard ensembles, providing a more robust framework for the emerging field of fully or partially data-driven reconstruction of dynamical systems.

1 Introduction

The mathematization of scientific disciplines has led to the widespread use of dynamical systems across a broad spectrum of fields, including Systems Biology. Systems biology focuses on the quantitative analysis of dynamic interactions among the components of a biological system, with the goal of understanding the behavior of the system as a whole [1]. Within this context, dynamical systems constitute a fundamental mathematical tool, providing a rigorous framework to describe, analyze, and predict the complex behaviors of biological systems over time [2]. By employing dynamical systems, researchers can investigate the interactions among biological entities, uncovering the mechanisms underlying various processes and predicting scenarios that have not been experimentally observed. Mathematically, dynamical systems are codified by systems of Ordinary Differential Equations (ODEs), which express the rate of change of a vector of variables $\mathbf{y} \in \mathbb{R}^n$ with respect to time as a function of $\mathbf{y}$ and t:

$$\frac{d\mathbf{y}}{dt} = f(\mathbf{y}).$$

Traditionally, the governing function f is *mechanistic* [3], encoding mathematically the theoretical knowledge about the biological system to be modeled. Models synthesized in this way are physically plausible (i.e., they comply with conservation laws and other known governing principles) and are typically interpretable. However, the *mechanistic* approach requires a detailed understanding of the system to be modeled and often relies on strong assumptions that may not be satisfied in real-world scenarios [4].

In recent decades, the growing availability of measured data has fostered the development of *data-driven* approaches to model dynamical systems [5]. Unlike the traditional *mechanistic* approach, these methods do not rely on the physical expertise about the system, but try to infer models directly from observational data. Various methodologies have been proposed, including symbolic regression [6], autoregressive models [7], recurrent neural networks (NNs) [8–10], and Neural Ordinary Differential Equations (NODEs) [11]. Among these, NODEs have emerged as particularly promising, as they retain the continuous-time structure of classical differential equations while substituting the explicit form of the governing function f with an NN:

$$\frac{d\mathbf{y}}{dt} = f_{NN}(\mathbf{y}, \boldsymbol{\theta}_{NN}), \tag{1}$$

where f_{NN} represents a NN and $\boldsymbol{\theta}_{NN}$ the vector of NN parameters. Thanks to the universal approximation theorem for NNs [12], it is theoretically guaranteed that, given

sufficient capacity, f_{NN} can approximate any continuous vector field f on a compact subset of $\mathbb{R}^n$—corresponding to the state space of the system—to arbitrary accuracy. Moreover, NODEs can be naturally extended to incorporate partial mechanistic knowledge through the framework of *Universal Differential Equations* (UDEs) [13]. In this formulation, the vector field can be decomposed as

$$\frac{d\mathbf{y}}{dt} = f_M(\mathbf{y}, \boldsymbol{\theta}_M) + f_{NN}(\mathbf{y}, \boldsymbol{\theta}_{NN}), \tag{2}$$

where f_M encodes partially known physical laws or domain-specific insights, $\boldsymbol{\theta}_M$ denotes the (possibly unknown) parameters associated with f_M, f_{NN} is a neural network that models the unknown residual dynamics, and $\boldsymbol{\theta}_{NN}$ is the corresponding vector of neural network parameters. UDEs and NODEs are increasingly being adopted as modeling frameworks in systems biology, see e.g. [14–17].

Despite the promise shown by NODEs and UDEs, several challenges persist in the data-driven reconstruction of dynamical systems [4]. A central issue is ensuring that models provide reliable information when applied to novel, previously unseen regimes [4, 18]. This issue is not specific to dynamical systems; rather, it reflects a broader limitation in Machine Learning (ML), where model performance often degrades when evaluated on domains that differ from those encountered during training [19]. Formally, the statistical background of ML relies on the assumption that both training and test data are drawn from the same underlying distribution. When this assumption is violated, the test data are considered *out-of-distribution* (OOD) [20], and model predictions typically become significantly less reliable. This represents a major challenge for the applicability of NODEs and UDEs in Systems Biology, as computational models in this field are often employed to simulate new experiments under previously unseen conditions, predict the outcomes of unobserved biological processes, and generate new testable hypotheses [21]. In such cases, uncertainty quantification (UQ) plays a pivotal role as it allows us to systematically estimate the confidence in model outputs and identify when predictions should—or should not—be trusted [22]. UQ refers to the set of techniques used to measure and represent the uncertainty inherent in model predictions, often by modeling the distribution, variance, or confidence intervals of possible outcomes. Broadly, UQ distinguishes between two types of uncertainty: on one side, aleatoric uncertainty, which arises from the intrinsic randomness in the training data; on the other side, epistemic uncertainty, which stems from limited knowledge of the system to be modeled [23]. The latter cause of uncertainty becomes particularly important when the model is applied in OOD contexts [23]. As such, UQ methods that reliably quantify the epistemic uncertainty in OOD conditions are increasingly recognized as a key requirement in the data-driven reconstruction of dynamical systems [4].

Preliminary studies have begun to explore UQ methods for NODEs and UDEs. Dandekar *et al.* [24] and Ott *et al.* [25] investigate the extension of classical Bayesian techniques [26] to the NODE and UDE frameworks. Takeishi *et al.* [27] propose incorporating regularizers into UQ methods to ensure the identifiability of the mechanistic component in a UDE. Schmid *et al.* [28] compare various Bayesian and ensemble-based approaches for UDEs, offering a formalization of UQ based on maximum likelihood

theory. However, none of these studies specifically focus on epistemic uncertainty, as the evaluation of the reliability of the quantified epistemic uncertainty is typically based only on isolated instances of new trajectories (with initial points perturbed relative to the training data) or, in the UDE context, by comparing the reconstructed distribution of the mechanistic component parameter values with the ground truth. To the best of our knowledge, no study has comprehensively investigated the reliability of epistemic uncertainty quantified on the reconstructed vector field of the dynamical system in regions different from the ones encountered during the training or from trajectories predicted by the model using initial states different from those encountered during training.

In this work, we focus on UQ based on standard ensembles (i.e., models trained independently with different random initializations), originally introduced for ML tasks unrelated to dynamical system modeling [29]. We have selected this approach because, despite its simplicity, it has been consistently shown to provide more reliable and practically useful epistemic uncertainty estimates compared to alternative techniques in standard regression and classification machine learning settings [22, 30–32]. Our aim is to investigate the reliability of the epistemic UQ provided by ensemble methods in OOD scenarios in the context of reconstructing dynamical systems using NODEs and UDEs. Consistent with observations in other ML applications [33–35], we find that in OOD dynamical system modeling contexts, often the models of standard ensembles exhibit a similar behavior: this leads to overconfident predictions, with the quantified uncertainty failing to account for the entire epistemic uncertainty in OOD scenarios. To address this challenge, we propose a novel method for constructing ensembles of NODEs and UDEs that fosters diversity in the reconstructed vector field across models in OOD regions of the state space, while maintaining explicit control over the loss function values of the models within the ensemble on the training trajectories. Our method builds on the assumption of *mode connectivity*, which posits that independently trained NODEs/UDEs can be connected via minimum-loss paths in parameter space. Inspired by gradient descent on Riemann manifolds [36], we route these paths to identify models that minimize training loss while maximizing OOD disagreement. Notably, mode connectivity has been validated for standard neural networks [37–39], but it has not yet been studied in the context of NODEs or UDEs. In this work, we provide the first formal proof of its extension to NODEs and discuss its potential validity in the UDE setting.

We evaluate our method and compare it to standard ensemble-based UQ using synthetic benchmarks previously employed in UQ studies for particular types of NODEs [40] and for data-driven dynamical system recovery [11, 13, 41, 42], derived from three qualitatively distinct dynamical systems: (1) the Lotka-Volterra system, i.e., a non-chaotic, nonlinear system with periodic orbits; (2) a damped harmonic oscillator, i.e., a linear, non-chaotic system with a fixed-point global attractor; and (3) the Lorenz system, i.e, a nonlinear, chaotic system.

2 Results

In this section, we begin by outlining the test cases and the metrics used to evaluate the reliability of epistemic UQ in OOD conditions (Section 2.1). We then introduce a motivating example, showing that standard ensembles risk being overconfident under OOD conditions in the context of dynamical systems reconstruction (Section 2.2). Next, we present our proposed method for constructing model ensembles, explicitly designed to maximize disagreement outside the training domain (Section 2.3). Finally, we evaluate the performance of our method across the different test cases, considering different levels of incorporated mechanistic prior knowledge (Section 2.4).

2.1 Test cases and metrics to evaluate uncertainty quantification

2.1.1 Test cases

For our analyses, we have selected three distinct dynamical systems. Although relatively simple, their vector fields exhibit qualitatively different characteristics that pose varying challenges for data-driven reconstruction. The first system is the Lotka-Volterra predator-prey model, defined by the two-dimensional ODE system:

$$\begin{cases} \frac{dx}{dt} = \alpha x - \beta xy \\ \frac{dy}{dt} = \delta xy - \gamma y \end{cases} \tag{3}$$

For parameters $\alpha, \beta, \gamma, \delta \in \mathbb{R}^+$, this system is non-chaotic and exhibits periodic trajectories. It has two fixed points at $(0, 0)$ and $(\gamma/\delta, \alpha/\beta)$ [43] and no global attractor. The second system is a nonlinear damped oscillator [40], defined by the two-dimensional ODE system:

$$\begin{cases} \frac{dx}{dt} = -\alpha x^3 - \beta y^3 \\ \frac{dy}{dt} = \beta x^3 - \alpha y^3 \end{cases} \tag{4}$$

With $\alpha, \beta \in \mathbb{R}^+$, this system is also non-chaotic and features spiral trajectories converging to a unique global attractor at the origin. The third system is the Lorenz system [44], originally developed to model atmospheric convection and now often studied as a canonical example of chaotic dynamics [45]. It is defined by:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(r - z) - y \\ \frac{dz}{dt} = xy - bz \end{cases} \tag{5}$$

For appropriate parameter choices, this system exhibits sensitive dependence on initial conditions, two unstable fixed points, and the emergence of the well-known Lorenz attractor [46].

For each dynamical system, we generated *in-silico* a training dataset by discretely sampling distinct trajectories obtained by simulating the corresponding ODE system. Details on the selected parameters, initial conditions, time intervals, and sampling

rates are provided in Section 4.1. Since the focus of this study is on epistemic UQ, no additional aleatoric noise has been added to the training trajectories.

2.1.2 Metrics

The above-mentioned setup has been designed to evaluate the ability of an ensemble of models to quantify the uncertainty inherent in reconstructing a ground-truth dynamical system under OOD conditions. In this context, we evaluate ensemble-based UQ through the analysis of the coverage properties of the 0.95-prediction intervals on model output [47], which represent the range within which the ground-truth is expected to fall with 95% probability.

Broadly speaking, given a ground truth model $g : A \to \mathbb{R}^n$ and a function $Pred^\alpha$ that associates to each element in A a prediction interval $Pred^\alpha(a)$ of level α on $g(a)$, we define the coverage of the prediction interval function $Pred^\alpha$ in a as the binary-valued function:

$$C(Pred^\alpha, a) = \begin{cases} 1 & \text{if } g(a) \in Pred^\alpha(a) \\ 0 & \text{otherwise.} \end{cases}$$

This function indicates whether the prediction interval at a contains the ground truth value. For a set of points $\{a_1, \ldots, a_l\} \subseteq A$, the coverage proportion (CP) of the prediction interval $Pred^\alpha$ over the set is the fraction of points for which the prediction interval covers the ground truth, and is defined as:

$$CP(Pred^\alpha, \{a_1, \ldots, a_l\}) = \frac{1}{l} \sum_{i=1}^{l} C(Pred^\alpha, a_i)$$

For clarity, we will omit the function inputs when there is no risk of confusion regarding how the prediction intervals are computed and where the CP is evaluated. When the prediction interval function is derived through stochastic processes—such as in ensembles of NNs where the training of each model involves randomness—we generate the ensemble m times, yielding m distinct prediction interval functions, $Pred_1^\alpha, \ldots, Pred_m^\alpha$. The mean coverage at a point a is then computed as the average coverage across these functions. Similarly, the mean CP over a set $\{a_1, \ldots, a_l\}$ is defined as the average of the CP values obtained from the individual prediction interval functions. The mean CP should converge to the nominal level α [47]; if it converges to a lower value, the prediction interval function is said to be overconfident, meaning it systematically underestimates the true uncertainty and produces intervals that are too narrow to achieve the desired coverage.

We evaluate the coverage properties of 0.95-prediction intervals on two key outputs in dynamical system reconstruction:

- **Reconstructed vector field**: Since NNs in NODEs/UDEs predict (a component of) the vector field governing the dynamical system, we first analyze the coverage properties of the prediction intervals on the vector fields. We then select a region

of the state space extending beyond the training trajectories and examine whether the prediction intervals cover the true vector field values in that region.

- **Reconstructed trajectories**: While the vector field defines the instantaneous rate of change at any point in the phase space, the ultimate goal of NODEs/UDEs is to reconstruct the full temporal dynamics of the system. To assess the coverage properties of the prediction intervals on the dynamics, we generate a set of ground-truth trajectories not seen during training. These trajectories are obtained by integrating the ground truth system forward over the time interval $[0, T]$ from perturbed versions of the initial conditions used for the training trajectories. Let $(t_1, \ldots, t_k)$ denote a regular sampling of the time interval $[0, T]$. For each generated trajectory, we sample the ground-truth system state at these time points and then evaluate the coverage of the prediction intervals on the ground truth in these points, with the prediction intervals generated by the ensemble of models reconstructing the original dynamical systems in either a purely or partially data-driven manner.

Details on the OOD scenarios considered for both vector fields and trajectories, as well as on the computation of prediction intervals given an ensemble of models, are provided in Section 4.2.

2.2 Motivation case

In this section, we show that in the context of dynamical system reconstruction, standard ensembles risk being overconfident when quantifying uncertainty under OOD conditions—an issue previously observed in other ML domains [33–35]. We consider the task of modeling, in a purely data-driven way, the three test case dynamical systems described in Section 2.1. In this setting, we employ pure NODEs, defined as in Eq. (1), with $\mathbf{y} \in \mathbb{R}^n$, with $n = 2$ for the Lotka-Volterra and Damped Oscillator systems, and $n = 3$ for the Lorenz system.

Our goal is to evaluate the coverage properties of the prediction intervals obtained with standard ensembles both on the vector fields and on the system trajectories. To this end, we train NODE ensembles on the three training trajectories detailed in Section 4.1, by only varying the random initialization of $\boldsymbol{\theta}_{NN}$, and evaluate the coverage performances using the metrics introduced in the previous section. The training procedure of the NODEs is detailed in Section 4.3. The NN f_{NN} used for all the test cases consists of an input layer, two hidden layers with 32 nodes each employing the *gelu* activation function, and an output layer, where the dimensionality of the respective dynamical system determines input and output dimensions. We note that this architecture is sufficiently expressive to accurately approximate the ground-truth vector fields in the considered OOD scenarios (see Supplementary Section S1). Therefore, it is theoretically possible to obtain a parameterization of this architecture that closely approximates the ground truth even under OOD conditions. For each test case, to account for the variability arising from the stochasticity of NN training, we train 10 ensembles on the training set, with each ensemble comprising 5 NNs. The quality of the fit on the training trajectories is discussed in Supplementary Section S2. The influence of considering different training datasets and of the ensemble size—an inherently

arbitrary modeling choice— will be discussed at the end of this section. The coverage properties of the resulting 0.95-prediction intervals are reported in Fig. 1.

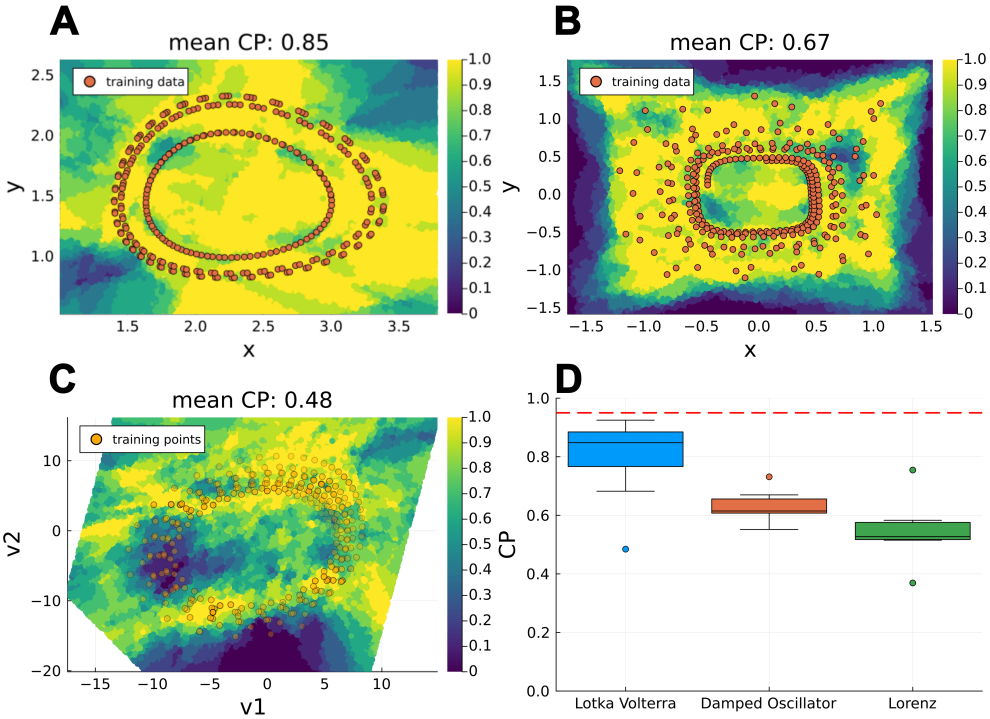

Fig. 1: Coverage of 0.95-prediction intervals with standard ensembles for data-driven reconstruction of dynamical systems. Panels A, B, and C display heatmaps of the mean coverage on the vector field, computed from 10 independently trained ensembles, for the Lotka-Volterra, Damped Oscillator, and Lorenz systems, respectively. Each panel title indicates the resulting mean CP (theoretical nominal coverage level of 0.95). Orange overlays represent the points of the training trajectories, providing spatial context. For the Lorenz system, the mean coverage is visualized on a two-dimensional plane derived via linear regression on the Lorenz attractor (see Supplementary Section S4 for details); here, training trajectory points are color-scaled based on their orthogonal distance to the plane. Panel D presents the distributions of CPs over reconstructed trajectories across the 10 ensembles trained for the different test cases. The dashed red line represents the theoretical nominal coverage level of 0.95.

We begin by analyzing the coverage of prediction intervals on the vector field. In the Damped Oscillator and Lorenz test cases, the mean CP indicates that the prediction intervals are overconfident in the region considered: the mean CP values are substantially below the nominal value of 0.95, and the CP distributions are significantly lower than 0.95 (Wilcoxon signed-rank test, $p < 0.001$ for both cases). In both systems, the mean coverage heatmaps reveal extensive regions in the state space where

all trained ensembles produce prediction intervals that consistently fail to cover the ground truth. The situation differs for the Lotka–Volterra case, where the mean CP is closer to the nominal value of 0.95, although the CP distribution remains significantly lower than this value (Wilcoxon signed-rank test, $p = 0.004$). This result depends on the region of the state space analyzed: when the region is extended, the overconfidence of the prediction intervals becomes more evident here as well, consistent with the other test cases (see Supplementary Section S3).

The coverage of the prediction intervals on the test trajectories is consistent with the results observed on the vector field. Specifically, it confirms the overconfidence of the prediction intervals for the Damped Oscillator and Lorenz test cases, with mean CP values of 0.63 and 0.59, respectively—well below the nominal value of 0.95—and CP distributions significantly lower than 0.95 (Wilcoxon signed-rank test, $p < 0.001$ for both cases). Figure 2 shows two example test trajectories for which the coverage of the prediction intervals is low. In the Lotka–Volterra test case, the mean CP is 0.80, closer to the nominal value with respect to the other test-cases, with a CP distribution significantly lower than this value (Wilcoxon signed-rank test, $p < 0.001$).

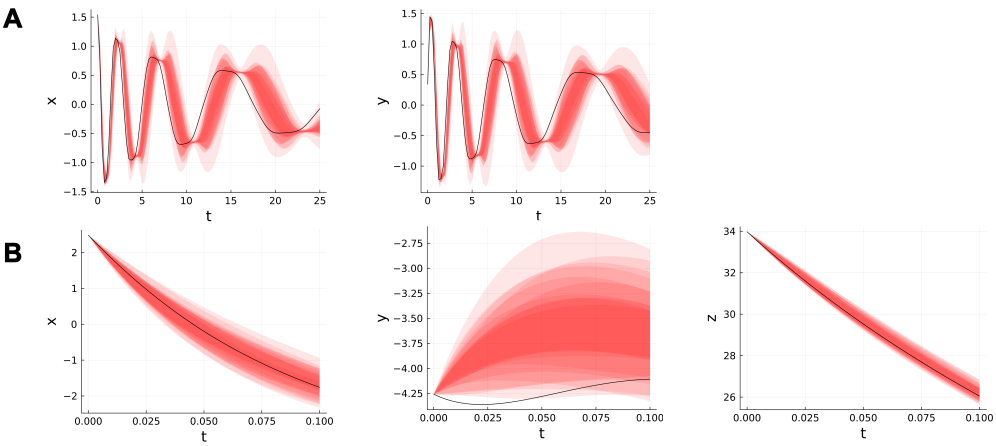

Fig. 2: **Examples of prediction intervals failing to cover the ground-truth trajectories with standard ensembles.** Panel A shows a test trajectory (solid black line) of the Damped Oscillator system, where none of the points are covered by the 0.95 prediction intervals (shaded red regions) produced by the trained standard ensembles. Panel B shows a test trajectory (solid black line) of the Lorenz system, where the y-variable values are covered by only one out of the thirty 0.95 prediction intervals produced by the trained standard ensembles (shaded red regions).

We conclude the analysis by investigating the effect of using different training datasets and varying ensemble sizes on the properties of the resulting prediction intervals in this scenario. On one hand, ensembles trained on different training datasets were considered in Supplementary Section S5: apart from a slight improvement in the mean CP of the Lotka–Volterra case, the results are consistent with those presented in this section. On the other hand, one might expect that increasing the ensemble size

introduces greater variability in the outputs, potentially leading to wider and more conservative prediction intervals. The impact of ensemble size on uncertainty quantification under OOD conditions has previously been explored by Ovadia et al.[37], who, in standard ML settings (regression and classification), found that ensembles of 5 models often provide a sufficient trade-off between performance and computational cost, with diminishing returns beyond this point. Here, we test whether a similar hypothesis holds in the context of dynamical system reconstruction. To this end, we trained ensembles of sizes 5, 10, 20, and 30, with 5 ensembles for each size, and compared the resulting CP distributions for both vector fields and trajectories. The results, presented in Fig.3, show that increasing the ensemble size does not produce a clear upward trend in CP for either trajectories or vector fields in any of the test cases. This suggests that enlarging the ensemble does not substantially increase variability in OOD predictions. A more detailed analysis of vector field coverage as a function of ensemble size is provided in Supplementary Section S6.

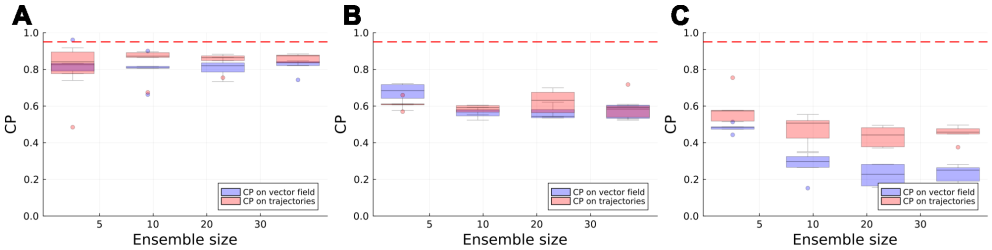

Fig. 3: **CP on vector fields and trajectories with varying standard ensemble sizes for a purely data-driven reconstruction of dynamical systems.** Panels A, B, and C display the CP distributions—shown in red for vector fields and in blue for trajectories—under OOD scenarios for the Lotka-Volterra, Damped Oscillator, and Lorenz test cases, respectively, considering different sizes of ensembles. The dashed red line represents the theoretical nominal coverage level of 0.95.

2.3 Maximizing OOD disagreement exploring loss contours

To mitigate the risk of overconfidence observed in standard ensembles under OOD conditions in Section 2.2, we propose a method for training ensembles with Maximized OOD Disagreement (MOD). The goal of this approach is to induce greater variability in the OOD behavior of the models within the ensemble.

We denote here a generic UDE model defined as in Eq (2); this formulation includes the pure NODE case as a special instance when $f_M = \mathbf{0}$. For convenience, supposing that $\boldsymbol{\theta}_M \in \mathbb{R}^k$ and $\boldsymbol{\theta}_{NN} \in \mathbb{R}^l$, we denote by $\boldsymbol{\theta}$ the concatenated parameter vector comprising both mechanistic and NN parameters, i.e., $\boldsymbol{\theta} = (\boldsymbol{\theta}_M, \boldsymbol{\theta}_{NN}) \in \mathbb{R}^{k+l}$ and with $g(\boldsymbol{\theta}, \mathbf{y})$ the RHS of Eq. (2). We further indicate with $\mathcal{L}(\boldsymbol{\theta})$ the loss function that compares the predictions of the UDE with given experimental data. Training a UDE model involves minimizing the function $\mathcal{L}(\boldsymbol{\theta})$ over $\mathbb{R}^{k+l}$. In general, due to the presence of the data-driven component f_{NN} and the potential non-identifiability of

some mechanistic parameters $\boldsymbol{\theta}^M$ [48], the minimum is not uniquely defined, but it is a generic subset

$$\mathbf{B} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}) \subseteq \mathbb{R}^{k+l}.$$

In these terms, assuming for simplicity only epistemic uncertainty, finding an ensemble of p models that fit the training set while exhibiting high disagreement under OOD conditions corresponds to finding $\mathbf{b_1}, \ldots, \mathbf{b_p} \in \mathbb{R}^{k+l}$ such that

$$\{\mathbf{b_1}, \ldots, \mathbf{b_p}\} \subset \mathbf{B},$$

and such that the models corresponding to these parameterizations display high disagreement in OOD conditions (our specific measure of disagreement is detailed later in this section). Our approach is based on the following assumption:

Assumption (Mode Connectivity for UDEs and NODEs). $\mathbf{B}$ is a connected subset of $\mathbb{R}^{k+l}$, meaning that given any two points $\mathbf{b_1}, \mathbf{b_2} \in \mathbf{B}$, there exists a continuous path connecting them that remains entirely within $\mathbf{B}$.

This hypothesis has been validated in various settings for standard neural networks [37–39]. To the best of our knowledge, however, it has not yet been formally established for either NODEs or UDEs. In Supplementary Section S7, we extend one of the formal proofs for NNs [49] to the cases of NODEs and UDEs with fixed mechanistic parameters, under hypotheses similar to those used for NNs. Although in UDEs with unknown mechanistic parameters this property may not always hold, preliminary evidence from traditional mechanistic ODE models used in Systems Biology [50] suggests that the hypothesis remains plausible in this setting as well. In Supplementary Section S7, we discuss cases in which the proof can be extended to this scenario, as well as rare instances in Systems Biology models where the mathematical structure of the mechanistic component prevents the assumption from holding.

Given this property of the loss landscape, if it is possible to move between two minimum-loss parameterizations along a continuous path, the core idea of our method is to exploit this connectivity. Specifically, starting from a minimum-loss parameterization, we aim to explore the set of parameters with minimal loss by following such paths, to discover other parameterizations that retain low training error while exhibiting markedly different behavior under OOD conditions. To formalize this approach, we must first (a) explicitly define a quantitative measure of OOD disagreement among models, and (b) describe how to maximize this disagreement while remaining within the set of minimum-loss solutions.

a. Measure of disagreement among UDEs. Given a point in the state space $\overline{\mathbf{y}} \in \mathbb{R}^n$, we measure the disagreement of an ensemble of models defined by the parameterizations $\mathbf{b_1}, \ldots, \mathbf{b_p}$ as follows:

- If $p \geq n$, we consider the generalized variance [51] of the vectors $g(\mathbf{b_1}, \overline{\mathbf{y}}), \ldots, g(\mathbf{b_p}, \overline{\mathbf{y}})$, i.e., the scalar quantity

$$\mathrm{dis}_{\overline{\mathbf{y}}}(\mathbf{b_1}, \ldots, \mathbf{b_p}) = \det\Big(\mathrm{cov}\left(g(\mathbf{b_1}, \overline{\mathbf{y}}), \ldots, g(\mathbf{b_p}, \overline{\mathbf{y}})\right)\Big), \tag{6}$$

that, intuitively, measures the overall variability of the vectors $g(\mathbf{b_1}, \overline{\mathbf{y}}), \ldots, g(\mathbf{b_p}, \overline{\mathbf{y}})$ [51].

- If $1 < p < n$, since the determinant in Eq. (6) is zero, we consider the sum of the variances along each axis of the space, that is:

$$\mathrm{dis}_{\overline{\mathbf{y}}}(\mathbf{b_1}, \ldots, \mathbf{b_p}) = \mathrm{Tr}\left(\mathrm{cov}(g(\mathbf{b_1}, \overline{\mathbf{y}}), \ldots, g(\mathbf{b_p}, \overline{\mathbf{y}}))\right),$$

  where $\mathrm{Tr}(\cdot)$ denotes the trace operator, i.e., the sum of the diagonal entries of the covariance matrix.

We can summarize our disagreement measure as:

$$\mathrm{dis}_{\overline{\mathbf{y}}}(\mathbf{b_1}, \ldots, \mathbf{b_p}) = \begin{cases} \det\left(\mathrm{cov}(g(\mathbf{b_1}, \overline{\mathbf{y}}), \ldots, g(\mathbf{b_p}, \overline{\mathbf{y}}))\right) & \text{if } p \geq n, \\ \mathrm{Tr}\left(\mathrm{cov}(g\mathbf{b_1}, \overline{\mathbf{y}}), \ldots, g(\mathbf{b_p}, \overline{\mathbf{y}}))\right) & \text{if } 1 < p < n. \end{cases}$$

For a finite set of points $\mathbf{S} = \{\overline{\mathbf{y}}_1, \ldots, \overline{\mathbf{y}}_m\}$, the disagreement of the models on $\mathbf{S}$, indicated with $\mathrm{dis}_{\mathbf{S}}(\mathbf{b_1}, \ldots, \mathbf{b_p})$, is computed as the average disagreement across points in $\mathbf{S}$.

b. How to maximize the disagreement while maintaining minimum loss. Our approach is inspired by constrained optimization on Riemannian manifolds [36]. Since we do not know whether $\mathbf{B}$ is itself a Riemannian manifold (although mode connectivity results for standard NNs suggest that, in several cases, the set of minimum-loss solutions forms a manifold [52]), we motivate our method intuitively as follows. Suppose we have an ensemble of p parameterizations $\mathbf{b_1}, \ldots, \mathbf{b_p} \in \mathbb{R}^{k+l}$ minimizing the loss function, i.e. $\mathbf{b_o} \in \mathbf{B}$ for each $o = 1, \ldots, p$. Let $\mathbf{S} \subset \mathbb{R}^n$ be a finite set of points in the state space of the model. Now assume that $\mathbf{b_1}, \ldots, \mathbf{b_{p-1}}$ are fixed, and we seek to locally move $\mathbf{b_p}$ in order to increase the disagreement among the parameterizations $\mathbf{b_1}, \ldots, \mathbf{b_p}$ on the set $\mathbf{S}$, while maintaining $\mathbf{b_p} \in B$. If our goal was solely to locally increase the disagreement among the parameterizations in $\mathbf{S}$, the natural update direction for $\mathbf{b_p}$ would be given by the gradient:

$$\mathbf{v} = \nabla_{\boldsymbol{\theta}}\, \mathrm{dis}_{\mathbf{S}}(\mathbf{b_1}, \ldots, \mathbf{b_{p-1}}, \boldsymbol{\theta})\big|_{\boldsymbol{\theta}=\mathbf{b_p}}.$$

However, an update of $\mathbf{b_p}$ along the direction $\mathbf{v}$ does not necessarily result in a point that remains within $\mathbf{B}$. On the other hand, denoting with H the Hessian matrix of $\mathcal{L}$ evaluated at $\mathbf{b_p}$, the null space H^0 of H—i.e., the eigenspace corresponding to zero eigenvalues, defines the subspace of directions along which the loss does not increase [48]. Therefore, the projection of $\mathbf{v}$ onto H^0 locally identifies the direction along which disagreement can be maximally increased without leaving the minimum-loss set $\mathbf{B}$. Our approach proceeds by locally updating $\mathbf{b_p}$ along this projected direction at each step.

Overall, our algorithm proceeds as follows (a visual description is provided in Fig. 4). Given a parameterization of the model trained using standard procedures, and a finite set of points $\mathbf{S} \subset \mathbb{R}^n$ in the state space, possibly located outside the training domain, where we aim to maximize the ensemble disagreement, we construct the

ensemble sequentially. The first model $\mathbf{b_1}$ is obtained by minimizing the loss function without considering disagreement. Then, each model $\mathbf{b_p}$, with $2 \leq p \leq n$, is initialized as $\mathbf{b_1}$ and trained for a fixed number of epochs maximizing the disagreement with the previously trained models $\mathbf{b_1}, \ldots, \mathbf{b_{p-1}}$ on the selected points $\mathbf{S}$, while simultaneously preserving minimum loss on the training data. For details on how the Hessian of the cost function is computed and how the constraint optimization is performed, we refer the reader to Section 4.4.

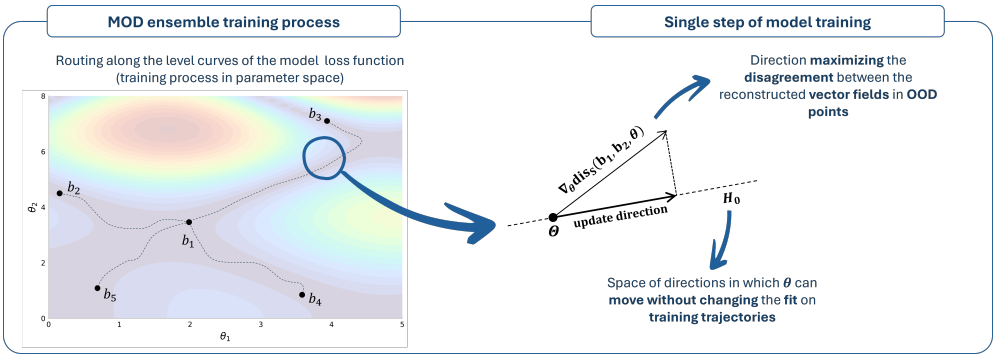

Fig. 4: Visual description of the algorithm for training MOD ensembles of UDEs/NODEs. Starting from a base model $\mathbf{b_1}$, trained using standard procedures, the algorithm constructs the ensemble sequentially. Each ensemble member is initialized from $\mathbf{b_1}$ and trained by following level curves of the loss function, rather than minimizing it directly. The training objective is to maximize disagreement with previously trained models on the reconstructed vector fields, specifically at selected OOD points, thus promoting diversity within the ensemble.

2.4 Results on test cases

We now evaluate the effectiveness of the algorithm described in Section 2.3 in mitigating overconfidence in OOD regions in the context of dynamical system reconstruction. The analysis is carried out for each test case described in Section 2.1, considering three different scenarios that reflect varying levels of prior mechanistic knowledge about the system being modeled:

- **full reconstruction**: in the first scenario, we assume no prior knowledge and perform a fully data-driven system reconstruction (Section 2.4.1);
- **partial reconstruction with known mechanistic parameters**: the second scenario incorporates partial mechanistic knowledge, assuming that the parameters of the mechanistic component of the model are known (Section 2.4.2);
- **partial reconstruction with unknown mechanistic parameters**: in the third scenario, we assume partial structural knowledge, where both the mechanistic parameters and the NN parameters are estimated from data (Section 2.4.3).

For each scenario, 10 standard ensembles of 5 models are trained using a fixed architecture, with the only variation being the random initialization of the NNs (and the mechanistic parameters in the third scenario) on the training dataset described in Section 4.1. Then, for each standard ensemble, one model is randomly selected and used as the initialization point b_1 for our algorithm to construct a corresponding 5-model MOD ensemble. This setup allows for a pairwise comparison between the results of standard ensembles and those of MOD ensembles. Since the training of NODEs and UDEs is susceptible to convergence to local minima of the cost function, an acceptance criterion is required when constructing standard ensembles. Specifically, we define an arbitrary threshold ϵ_{acc} on the cost function which is used both to determine whether a trained model is accepted or rejected in standard ensembles and as a tolerance level on the cost function in our algorithm. In most cases, we adopt a value of 10^{-3}; however, in the context of the first scenario, we demonstrate that the algorithm can also be successfully executed with a more stringent threshold of 10^{-4}, if necessary.

To run the algorithm to train a MOD ensemble, it is necessary to select points in the state space where the generalized variance of the ensemble predictions is to be maximized. For this purpose, we selected regular grids of points within the OOD regions defined for the analysis. For further details on the algorithmic implementation and settings used to run the MOD ensemble training, the reader is referred to Section 4.5.

2.4.1 Full reconstruction

In the first scenario, we consider a purely data-driven setting, where all models are implemented as pure NODEs, as in the motivational case exposed in Section 2.2. The architecture of the NN consists of an input layer, two hidden layers with 32 nodes each using the *gelu* activation function, and an output layer, where the input and output dimensions depend on the dimensionality of the dynamical system. For each test case, we train 10 standard ensembles and the corresponding 10 MOD ensembles. The training fit of all 50 NODEs comprising the standard ensembles, as well as the 50 NODEs in the MOD ensembles, is analyzed in Supplementary Section S8.

We begin by analyzing the coverage of 0.95-prediction intervals on the vector field obtained with the two approaches (standard and MOD ensembles). The CP results are reported in Table 1, whereas the mean coverage heatmaps in the considered regions of the state space are shown in Fig. 5.

The results indicate that MOD ensembles yield mean CP values significantly closer to the nominal value of 0.95 than standard ensembles in all test cases (one-sided Wilcoxon signed-rank test, $p < 0.001$). As in Section 2.2, in the Lotka–Volterra test case, standard ensembles produce prediction intervals with a mean CP close to the nominal value, which becomes even closer when MOD ensembles are applied. In the Damped Oscillator and Lorenz test cases, the prediction intervals produced by standard ensembles are distinctly overconfident, and this problem is mitigated by applying MOD ensembles. In the Damped Oscillator test case, the mean CP reaches a value slightly above the nominal value, and the mean coverage heatmaps show that our approach successfully mitigates the overconfidence exhibited by standard ensembles across entire regions of the state space. In the chaotic Lorenz system, although the mean CP obtained with MOD ensembles remains below the nominal value, it is still

	Lotka-Volterra	**Damped Oscillator**	**Lorenz**
Standard	0.850 ± 0.024	0.669 ± 0.016	0.458 ± 0.010
MOD	**0.986 ± 0.006** (*)	**0.993 ± 0.003** (**)	**0.760 ± 0.040** (**)

Table 1: **Comparison of the CP of** 0.95**-prediction intervals on the vector field (full reconstruction scenario).** Mean CP of 0.95-prediction intervals on the vector field within a selected region of the state space. The results are presented as the mean ± SEM across 10 ensembles. Statistical significance was assessed using the Wilcoxon signed-rank test on absolute deviations from 0.95 between the two distributions of CP values. Asterisks indicate the following significance levels: * $p \leq 0.05$, ** $p \leq 0.01$, and *** $p \leq 0.001$. The values shown in bold are those closest to the theoretical value.

closer to the nominal value than that achieved by standard ensembles. The increase in CP can be primarily attributed to an increase in the generalized variance of the ensemble predictions, particularly in regions of the state space that are distant from the training trajectory points, where standard ensembles tend to exhibit overconfidence, as discussed in Supplementary Section S9. Notably, the effect of MOD ensemble on mean CP is not confined to the selected region of the state space—where the algorithm was explicitly trained to maximize the generalized variance—but also extends to a broader region, as analyzed in Supplementary Section S10.

The results for the coverage of the prediction intervals on trajectories are consistent with those observed on the vector field and are summarized in Table 2. In the Lotka–Volterra system, the improvement brought by MOD ensembles is statistically less pronounced, as standard ensembles already yield mean CP values relatively close to the nominal value of 0.95. In contrast, for the Damped Oscillator and Lorenz systems, the prediction intervals obtained with MOD ensembles produce mean CP values significantly closer to the nominal value than those from standard ensembles. An example illustrating how our algorithm improves the coverage for the trajectories previously presented in the motivation case is shown in Fig. 6.

	Lotka-Volterra	**Damped Oscillator**	**Lorenz**
Standard	0.800 ± 0.042	0.627 ± 0.016	0.546 ± 0.030
MOD	**0.926 ± 0.019** (*)	**0.954 ± 0.012** (**)	**0.738 ± 0.030** (**)

Table 2: **Comparison of the CP of** 0.95**-prediction intervals on the trajectories (full reconstruction scenario).** Mean CP of 0.95-prediction intervals on trajectories on the selected points of test trajectories. The results are presented as the mean ± SEM across 10 ensembles. Statistical significance was assessed using the Wilcoxon signed-rank test on absolute deviations from 0.95 between the two distributions of CP values. Asterisks indicate the following significance levels: * $p \leq 0.05$, ** $p \leq 0.01$, and *** $p \leq 0.001$. The values shown in bold are those closest to the theoretical value.

In this first scenario, we examined the effect of varying the tolerance threshold in the cost function used to train the MOD ensembles, as well as the impact of considering

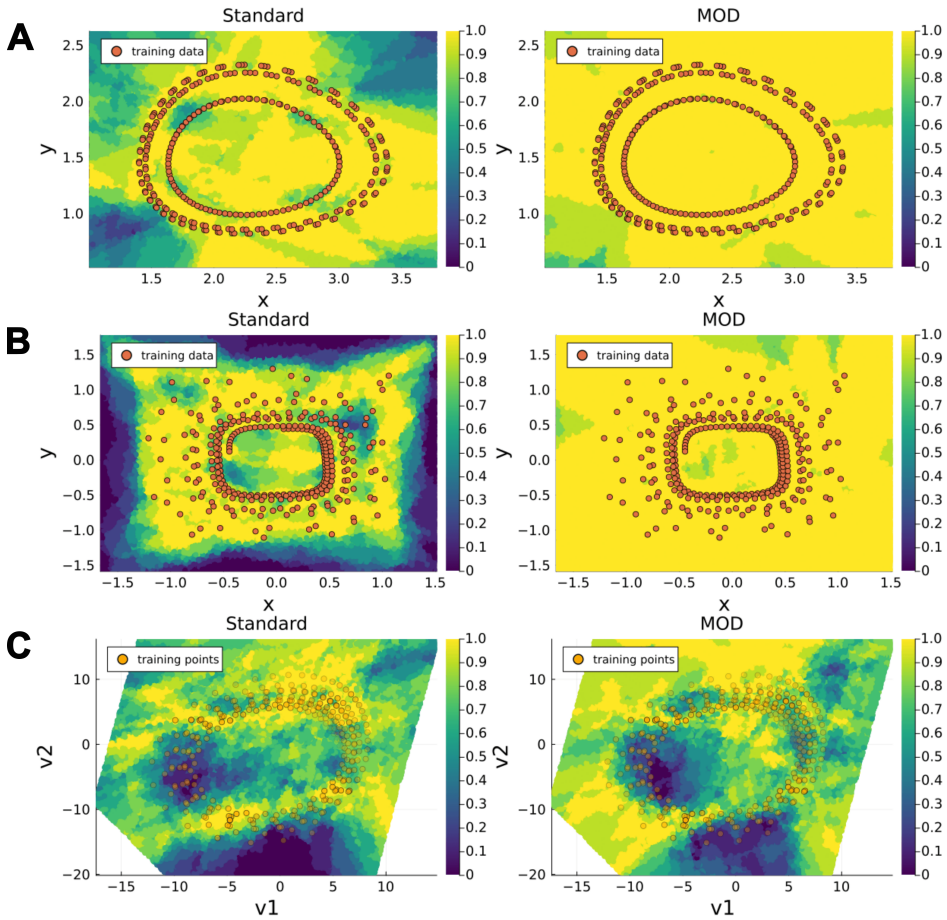

Fig. 5: Comparison of 0.95-prediction interval coverage on state space: Standard vs. MOD Ensembles (full reconstruction scenario). Panels A, B, and C display heatmaps of the mean coverage (computed across 10 ensembles) of 0.95-prediction intervals on the state space for the Lotka-Volterra, Damped Oscillator, and Lorenz systems, respectively. For each system, the left subpanel shows results obtained with standard ensembles, while the right subpanel shows results obtained with MOD ensembles. Orange overlays represent the points from the training trajectories, providing spatial context. For the Lorenz system, the mean coverage is visualized on a two-dimensional plane derived via linear regression on the Lorenz attractor (see Supplementary Section S4 for details); here, the transparency of training trajectory points indicates their orthogonal distance from the training trajectory points to the plane.

different training datasets (as done in Section 2.2 for standard ensembles). Regarding the tolerance threshold, in this section it was fixed at 10^{-3}. Although this is the same threshold used as the acceptance criterion for NODEs in the standard ensembles, the models obtained through the MOD ensemble training algorithm exhibited a slightly worse fit on the training trajectories in the Lotka–Volterra and Damped Oscillator

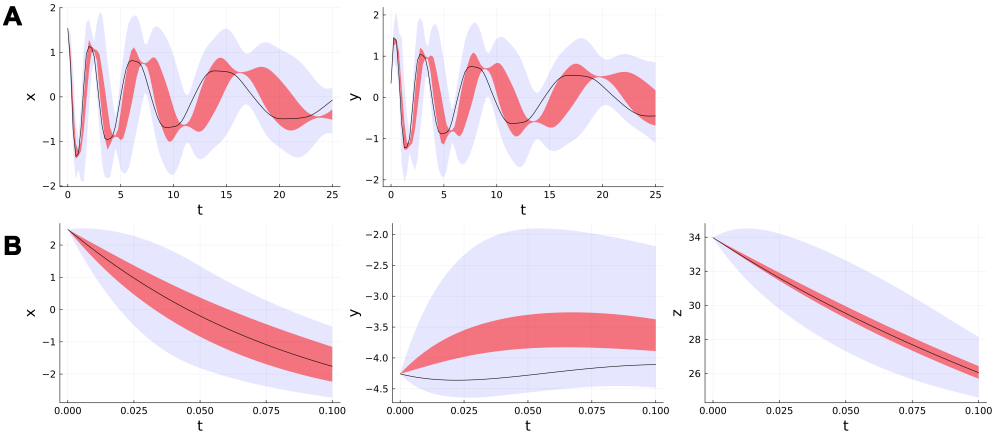

Fig. 6: **Comparison between prediction interval on trajectories obtained with a standard ensemble and a corresponding MOD ensemble (full reconstruction scenario).** Panels A and B illustrate two trajectories for which the trained standard ensembles produce overconfident 0.95-prediction intervals in the Damped Oscillator and Lorenz test cases, respectively, whereas the corresponding MOD ensemble covers the ground truth. The shaded red regions denote the 0.95-prediction intervals obtained with the standard ensemble, the shaded blue regions denote the 0.95-prediction intervals obtained with the MOD ensemble, and the solid black line represents the corresponding ground-truth trajectory.

test cases (see Supplementary Section S8). To demonstrate that the increased OOD disagreement is not merely a consequence of degraded in-distribution performance, we repeated the MOD ensemble training for these two test cases using a stricter threshold of 10^{-4}. The results, presented in Supplementary Section S11, confirm that comparable performance can also be achieved with this lower threshold. Furthermore, the results of MOD ensembles trained on different datasets, reported in Supplementary Section S12, are consistent with those presented here and further underscore the generality of our approach.

2.4.2 Partial reconstruction with known mechanistic parameters

In the second scenario, we assume partial knowledge of the dynamical system, with only a portion of the system derivatives requiring reconstruction through data-driven methods.

Specifically, for the Lotka–Volterra system, we assume no knowledge of the interaction terms βxy and γxy in Eq. (3), and we replace them with an NN, leading to the following formulation:

$$\begin{cases} \frac{dx}{dt} = \alpha x - f_{NN}(x,y)[1] \\ \frac{dy}{dt} = f_{NN}(x,y)[2] - \gamma y \end{cases} \tag{7}$$

where $f_{NN} : \mathbb{R}^2 \to \mathbb{R}^2$ is an NN that takes (x, y) as input and outputs the approximated interaction terms.

In the Damped Oscillator test case, we assume the nonlinear coupling terms $-\beta y^3$ in the first equation and βx^3 in the second equation of Eq. (4) to be unknown, and

we replace them with an NN, yielding:

$$\begin{cases} \frac{dx}{dt} = -\alpha x^3 - f_{NN}(x,y)[1] \\ \frac{dy}{dt} = f_{NN}(x,y)[2] - \alpha y^3 \end{cases} \tag{8}$$

where, $f_{NN} : \mathbb{R}^2 \to \mathbb{R}^2$ takes (x, y) as input and approximates the unknown dynamics.

In the Lorenz system, we assume that the third equation in Eq. (5) is entirely unknown and substitute it with an NN, while keeping the first two equations intact:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(r - z) - y \\ \frac{dz}{dt} = f_{NN}(x, y, z) \end{cases} \tag{9}$$

where $f_{NN} : \mathbb{R}^3 \to \mathbb{R}$ is an NN that takes (x, y, z) as input and outputs the reconstructed component of the dynamics.

In this scenario, for every test case, we assume to know the values of the parameters of the known portion of the system. The architecture of the NN used in this scenario is consistent with the one used in the first scenario, comprising two hidden layers, each containing 32 neurons, and employing the *gelu* activation function. In the case of the Lorenz system, however, the output layer consists of a single neuron, reflecting the dimensionality of the portion of the derivatives to be approximated. As in the other scenarios, for each test case, we train 10 standard ensembles and 10 corresponding MOD ensembles. The fit of the resulting models on the training trajectories is reported in Supplementary Section 13.

The results in terms of CP for the prediction intervals on the vector fields are presented in Table 3 and are consistent with those observed in the first scenario. For the Lotka–Volterra system, the standard ensembles already produce prediction intervals with a mean CP close to the nominal value of 0.95. When MOD ensembles are applied, the mean CP slightly exceeds the nominal value; in this case, the standard ensembles yield a mean CP closer to 0.95, although the absolute deviations are less than 0.05 in both cases. In contrast, MOD ensembles lead to prediction intervals with mean CP values distinctly closer to the nominal value in the Damped Oscillator and Lorenz test cases, where standard ensembles produce overconfident prediction intervals, with entire regions of the state space in which the prediction intervals fail to cover the ground truth for all trained standard ensembles (Fig. 7). As in the first scenario, the increases in mean CP on the vector field are associated with an increase in the generalized variance of the ensemble members in the state space (see Supplementary Section S14). Moreover, these results persist when the evaluation region is extended beyond the considered portion of the state space, as shown in Supplementary Section S15.

The coverage of the prediction intervals on trajectories is closely aligned with the behavior observed on the vector field (Table 4). For the Lotka–Volterra system, standard ensembles already produce prediction intervals with a mean CP close to the nominal value, and MOD ensembles result in a change that is not statistically significant. In contrast, in the Damped Oscillator and Lorenz test cases, the mean

	Lotka-Volterra	**Damped Oscillator**	**Lorenz**
Standard	**0.941 ± 0.018** (**)	0.620 ± 0.014	0.566 ± 0.036
MOD	0.998 ± 0.001	**0.997 ± 0.001** (**)	**0.909 ± 0.016** (**)

Table 3: **Comparison of the CP of** 0.95**-prediction intervals on the vector field (partial reconstruction with known mechanistic parameters scenario).** Mean CP of 0.95-prediction intervals on the vector field within a selected region of the state space. The results are presented as the mean ± SEM across 10 ensembles. Statistical significance was assessed using the Wilcoxon signed-rank test on absolute deviations from 0.95 between the two distributions of CP values. Asterisks indicate the following significance levels: * $p \leq 0.05$, ** $p \leq 0.01$, and *** $p \leq 0.001$. The values shown in bold are those closest to the theoretical value.

CP of the prediction intervals generated by MOD ensembles is distinctly closer to the nominal value, exceeding 0.94 in both cases.

	Lotka-Volterra	**Damped Oscillator**	**Lorenz**
Standard	**0.922 ± 0.010**	0.454 ± 0.027	0.811 ± 0.035
MOD	0.982 ± 0.002	**0.946 ± 0.011** (**)	**0.950 ± 0.014** (**)

Table 4: **Comparison of the CP of** 0.95**-prediction intervals on the trajectories (partial reconstruction with known mechanistic parameters scenario).** Mean CP of 0.95-prediction intervals on trajectories on the selected points of test trajectories. The results are presented as the mean ± SEM across 10 ensembles. Statistical significance was assessed using the Wilcoxon signed-rank test on absolute deviations from 0.95 between the two distributions of CP values. Asterisks indicate the following significance levels: * $p \leq 0.05$, ** $p \leq 0.01$, and *** $p \leq 0.001$. The values shown in bold are those closest to the theoretical value.

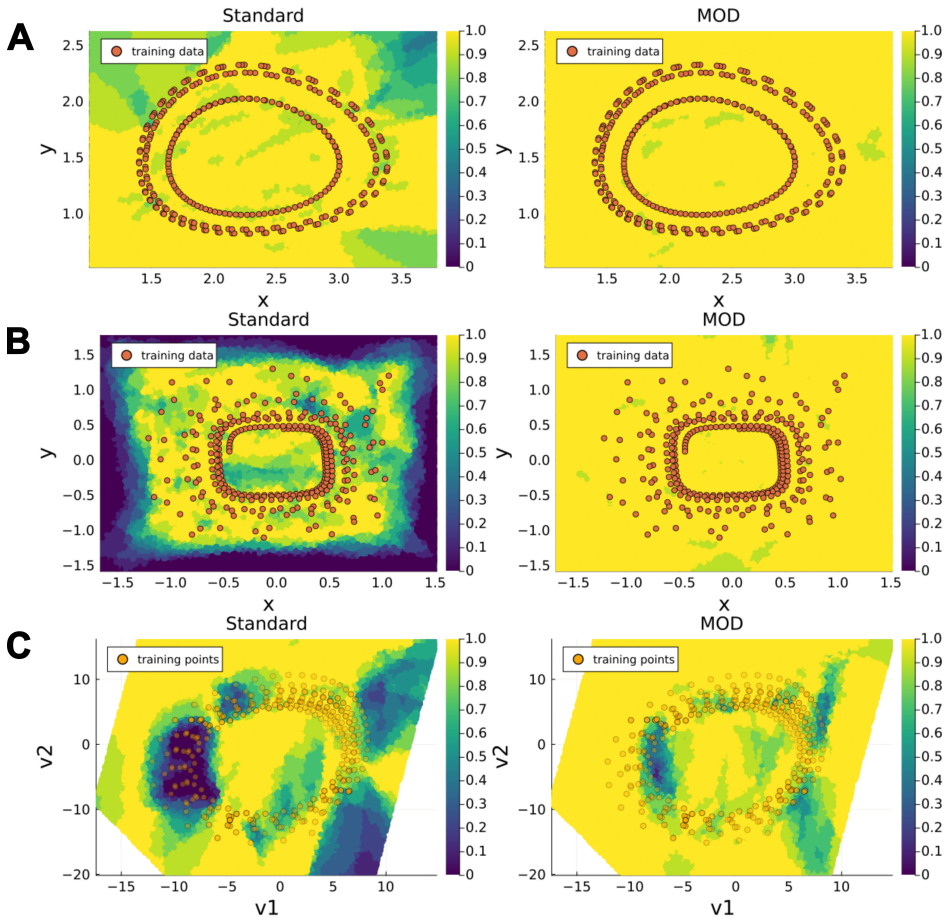

Fig. 7: **Comparison of** 0.95**-prediction interval coverage on state space: Standard vs. MOD Ensembles (partial reconstruction with known mechanistic parameters scenario).** Panels A, B, and C display heatmaps of the mean coverage (computed across 10 ensembles) of 0.95-prediction intervals on the state space for the Lotka-Volterra, Damped Oscillator, and Lorenz systems, respectively. For each system, the left subpanel shows results obtained with standard ensembles, while the right subpanel shows results obtained with MOD ensembles. Orange overlays represent the points from the training trajectories, providing spatial context. For the Lorenz system, the mean coverage is visualized on a two-dimensional plane derived via linear regression on the Lorenz attractor (see Supplementary Section S4 for details); here, the transparency of training trajectory points indicates their orthogonal distance from the training trajectory points to the plane.

2.4.3 Partial reconstruction with unknown mechanistic parameters

In the third scenario, we assume the same prior knowledge about the dynamical systems as in the second scenario. However, in this case, we do not assume prior knowledge

of the parameters of the known mechanistic portions of the model. Instead, we estimate these parameters alongside the NN parameters. Specifically, we estimate the parameters α and γ for the Lotka-Volterra test case, α for the Damped Oscillator test case, and σ and r for the Lorenz test case. The mechanistic parameters were initialized within a boundary range between 0.5 and 2 times the ground truth values (reported in Section 4.1). The architecture used for the NN is identical to the one used in the first two scenarios. As in the other scenarios, we train 10 standard ensembles and 10 corresponding MOD ensembles. The fit results on the training trajectories are reported in Supplementary Section S16.

The results for the coverage of the 0.95 prediction intervals on vector fields for standard ensembles and MOD ensembles, in terms of mean CP and mean coverage, are presented in Table 5 and Fig. 8, respectively. In the Lotka–Volterra system, the prediction intervals generated by standard ensembles are not overconfident, and in this case MOD ensembles lead to a mean CP slightly above the nominal value, although in both cases the deviation from 0.95 is less than 0.05. In the Damped Oscillator system, standard ensembles exhibit distinct regions where the prediction intervals are overconfident, resulting in a mean CP of 0.597, while MOD ensembles increase the mean CP to 0.997, bringing it distinctly closer to the nominal value. In the Lorenz system, the prediction intervals from standard ensembles exhibit relatively high coverage compared to the results obtained for the same system in the other scenarios, and the mean CP obtained with MOD ensembles is even closer to 0.95, although the improvement in CP between the model pairs is not statistically significant. As in the other scenarios, the considerations regarding the validity of the results over a broader region of the state space, along with the increase in generalized variance compared to the standard ensembles, remain valid (see Supplementary Sections S17 and S18).

	Lotka-Volterra	**Damped Oscillator**	**Lorenz**
Standard	**0.955 ± 0.015** (**)	0.597 ± 0.014	**0.811 ± 0.054**
MOD	0.999 ± 0.001	**0.997 ± 0.002** (**)	**0.841 ± 0.055**

Table 5: **Comparison of the CP of** 0.95**-prediction intervals on the vector field (partial reconstruction with unknown mechanistic parameters scenario).** Mean CP of 0.95-prediction intervals on the vector field within a selected region of the state space. The results are presented as the mean ± SEM across 10 ensembles. Statistical significance was assessed using the Wilcoxon signed-rank test on absolute deviations from 0.95 between the two distributions of CP values. Asterisks indicate the following significance levels: * $p \leq 0.05$, ** $p \leq 0.01$, and *** $p \leq 0.001$. The values shown in bold are those closest to the theoretical value.

The results for the CP of the 0.95 prediction intervals on trajectories (Table 6) are consistent with those observed on the vector field. In particular, for the Damped Oscillator and Lorenz systems, MOD ensembles mitigate the overconfidence present in the prediction intervals generated by standard ensembles. The improvement is especially pronounced in the Damped Oscillator test case, where the mean CP from standard ensembles is particularly low and is increased to above 0.90 by MOD ensembles.

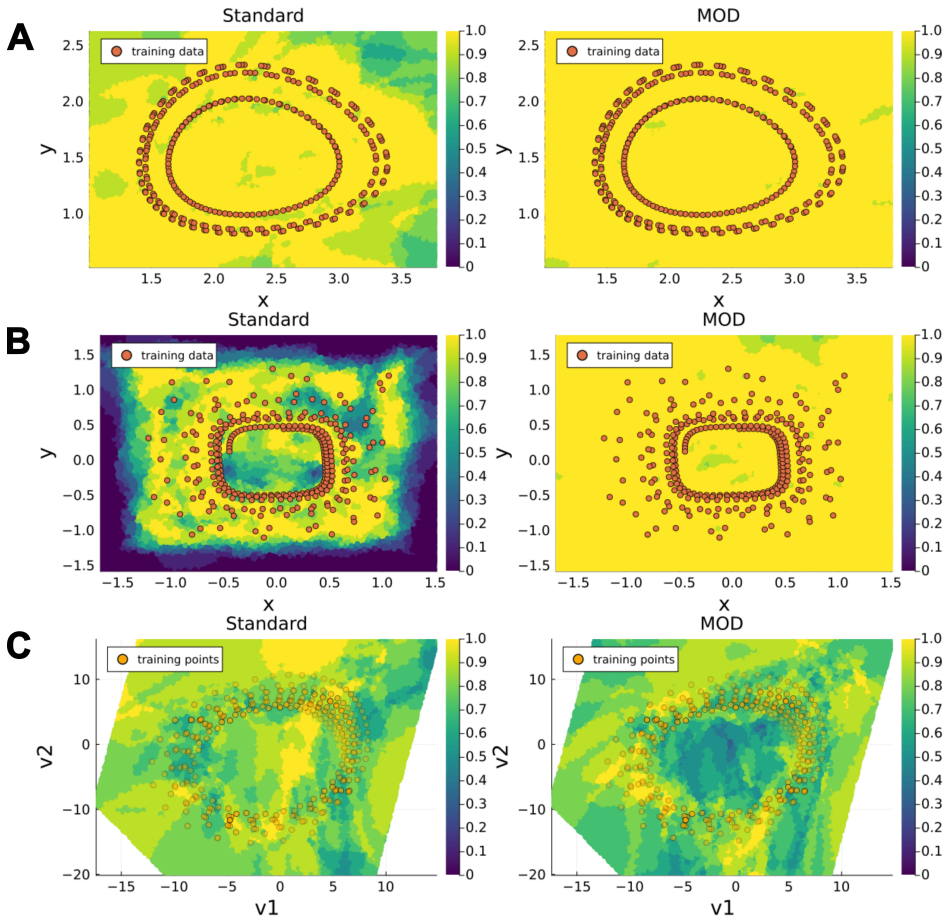

Fig. 8: **Comparison of** 0.95**-prediction interval coverage on state space: Standard vs. MOD Ensembles (partial reconstruction with unknown mechanistic parameters scenario).** Panels A, B, and C display heatmaps of the mean coverage (computed across 10 ensembles) of 0.95-prediction intervals on the state space for the Lotka-Volterra, Damped Oscillator, and Lorenz systems, respectively. For each system, the left subpanel shows results obtained with standard ensembles, while the right subpanel shows results obtained with MOD ensembles. Orange overlays represent the points from the training trajectories, providing spatial context. For the Lorenz system, the mean coverage is visualized on a two-dimensional plane derived via linear regression on the Lorenz attractor (see Supplementary Section S4 for details); here, the transparency of training trajectory points indicates their orthogonal distance from the training trajectory points to the plane.

We conclude this section by examining how mechanistic parameter identifiability affects the performance of our algorithm. In the test cases considered, there are two distinct situations: the mechanistic parameters in the Lotka-Volterra UDE (both α and δ) and the Damped Oscillator UDE (α) are not identifiable, while the mechanistic parameters in the Lorenz UDE (both σ and r) are identifiable. This phenomenon can

	Lotka-Volterra	**Damped Oscillator**	**Lorenz**
Standard	**0.947 ± 0.011** (**)	0.440 ± 0.038	0.813 ± 0.047
MOD	0.987 ± 0.001	**0.905 ± 0.026** (**)	**0.945 ± 0.019**

Table 6: **Comparison of the CP of** 0.95**-prediction intervals on the trajectories (partial reconstruction with unknown mechanistic parameters scenario).** Mean CP of 0.95-prediction intervals on trajectories on the selected points of test trajectories. The results are presented as the mean ± SEM across 10 ensembles. Statistical significance was assessed using the Wilcoxon signed-rank test on absolute deviations from 0.95 between the two distributions of CP values. Asterisks indicate the following significance levels: * $p \leq 0.05$, ** $p \leq 0.01$, and *** $p \leq 0.001$. The values shown in bold are those closest to the theoretical value.

be informally attributed to the universal approximation properties of NNs and can be empirically validated by the scatter of values observed from standard ensemble training or by evaluating the Hessian matrix of the cost function at the minimum points of the parameter space identified by the standard ensemble training, as proposed in [48] (further details are provided in Supplementary Section S19). By examining the trajectories of the mechanistic parameters during the training of MOD ensembles (Fig. 9), we observe that the MOD training algorithm tends to explore different regions of the mechanistic parameter space when the parameters are non-identifiable (Lotka-Volterra and Damped Oscillator systems), whereas it is constrained to maintain the parameters near the initial model values when they are identifiable (Lorenz system).

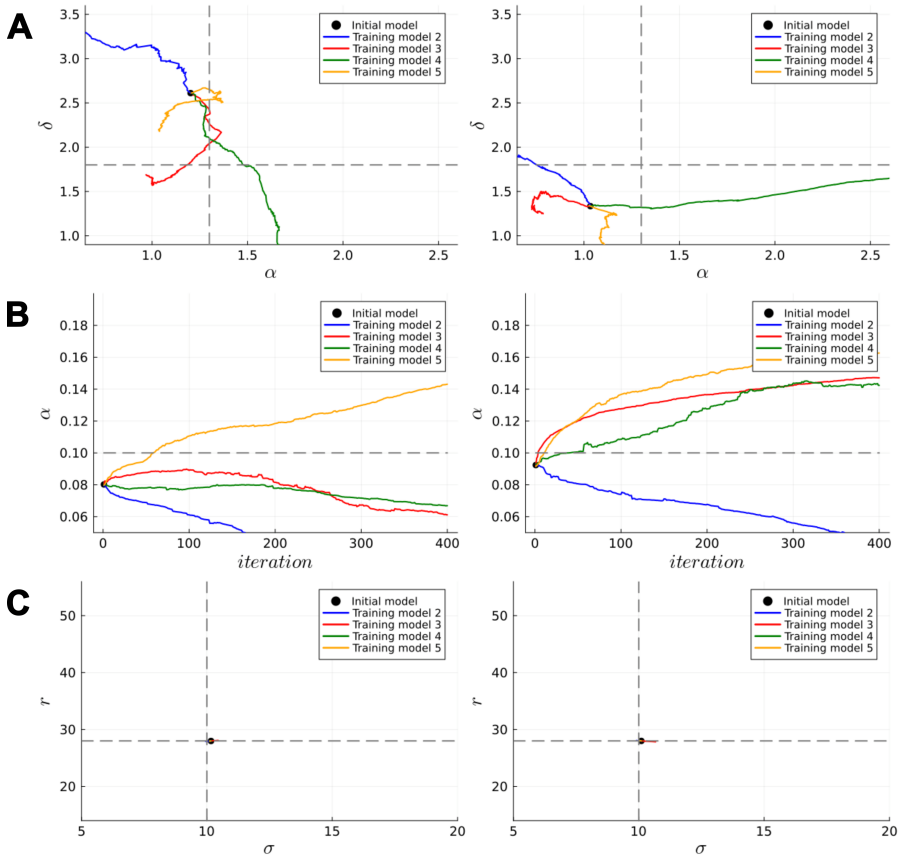

Fig. 9: **Trajectories described by the mechanistic parameters during the training of models in MOD ensembles.** Panels A, B, and C show the trajectories described by the mechanistic parameters during the training of models in the MOD ensembles for the Lotka-Volterra, Damped Oscillator, and Lorenz systems, respectively. For each system, trajectories from two example ensembles are presented. The solid black point represents the mechanistic parameters of the initial model from which the algorithm begins, while the grey dashed lines indicate the ground truth values of the parameters. For the Lotka-Volterra and Lorenz systems, the trajectories are depicted in the two-dimensional space of the mechanistic parameters of the UDEs. For the Damped Oscillator, the trajectory described by the sole mechanistic parameter of the UDE is plotted against the number of iterations. The limits of the plots correspond to the boundaries assumed for the mechanistic parameter values.

3 Discussion

In this work, we tackle the critical challenge of UQ in the data-driven reconstruction of dynamical systems using NODEs and UDEs. We consider two primary outputs of the data-driven reconstruction process: (i) the reconstructed vector fields and (ii) the prediction of temporal dynamics that were not observed during training. Our emphasis is on quantifying epistemic uncertainty both on OOD regions of the state space

and on previously unseen trajectories—a theme that, to the best of our knowledge, has not been systematically addressed in the existing literature. We focus on ensemble methods, which have been shown to provide more reliable and practically useful epistemic uncertainty estimates compared to alternative techniques in standard regression and classification ML settings [22, 30–32]. Using *in silico* test cases, we show that standard ensembles may fail to provide reliable epistemic uncertainty estimates under OOD conditions, due to the similarity of OOD behavior across ensemble members—a phenomenon already observed in traditional ML contexts [33–35]. Motivated by these limitations, we propose a novel ensemble training strategy designed to build ensembles that exhibit different behavior in OOD conditions. Our method involves sequentially training each model in the ensemble while maximizing the disagreement on selected OOD regions of the state space among the ensemble models, explicitly controlling the loss error on the training dataset. The theoretical justification of the method relies on the assumption of single-mode connectivity, an assumption that has been validated in various contexts for both standard NNs and mechanistic systems [37–39, 50], but not yet investigated, to the best of our knowledge, for NODEs and UDEs; we present an initial extension of an existing formal proof for NNs [49] to the cases of NODEs and UDEs with fixed mechanistic parameters, and discuss its potential applicability in the general UDE setting. The proposed approach is evaluated on three *in silico* test cases characterized by different vector field properties. For each test case, we explore different levels of prior physical knowledge about the system: a fully data-driven reconstruction, a partially data-driven reconstruction where parts of the system derivatives are known exactly, and a partially data-driven reconstruction where the derivative structure is known but its parameters must be estimated. Our results show that MOD ensembles significantly improve the reliability of UQ compared to standard ensembles in scenarios where the latter exhibit overconfidence.

The main limitations of this work are: (a) the occurrence of (rare) cases in which the mechanistic component of the model may violate the single-mode connectivity assumption; (b) the exclusion of aleatoric uncertainty; (c) the reliance on an arbitrary threshold for routing the level curves of the loss function; and (d) the dependence on the chosen region of the state space and the lack of a clear stopping criterion for terminating the training of each model. These aspects are examined in more detail in the following paragraphs.

Occurrence of (rare) cases in which the mechanistic component of the model may violate the single-mode connectivity. The assumption of single-mode connectivity in UDEs fails when the mechanistic component of the model contains parameters that are locally identifiable but not globally identifiable. In such instances, the loss function may exhibit minima that are not connected. The rarity and the causes of this phenomenon have been explored in the context of biological mechanistic modeling by Barreiro *et al.* [50], who investigated its occurrence across a range of published models. They identified specific conditions for this phenomenon, such as parameters appearing with even exponents, which can admit both positive and negative values. In these cases, the single-mode connectivity assumption should be replaced with the hypothesis that the set of global minima in the UDE consists of several connected components. This would necessitate an adaptation of the MOD

training algorithm, requiring initialization from multiple, traditionally trained parameterizations of the model, each located in a distinct connected component. Extending the methodology to address such (rare) cases is beyond the scope of the present study.

Epistemic but not aleatoric uncertainty. In the present study, we focus on epistemic uncertainty, which arises from the partial information available about the model being reconstructed. In contrast, aleatoric uncertainty represents another fundamental source of uncertainty, caused by the intrinsic randomness present in the phenomenon being modeled [53]. This distinction explains why random noise has not been added to the training time series in our test cases. In the presence of aleatoric uncertainty, modifications to the algorithm would be necessary, particularly concerning the approximation of the Hessian matrix of the loss function, as the current approximation (see Section 4.4.2) assumes zero residuals [54]. Analyzing the impact of aleatoric uncertainty on the MOD ensemble training algorithm is beyond the scope of this work and is thus left for future research.

Threshold for accepting an ensemble member. The threshold ϵ_{acc} on the cost function used both to select the members to accept in the standard ensemble and to run the algorithm for training MOD ensembles is let as an arbitrary choice of the modeler. We have shown that the MOD ensemble training algorithm can function under different threshold values (Supplementary Section S11), but there is no formal justification for preferring one over another. Our cost function could be straightforwardly adapted to a likelihood-based formulation by assuming that the error model governing the phenomenon is an i.i.d. Gaussian process with a standard deviation proportional to the min-max oscillations of the variables. In mechanistic modeling theory, under this assumption, the threshold on the cost value can be derived from the likelihood ratio test [55]. We did not adopt this statistical criterion due to the non-identifiability of the models under investigation. It is well established that the asymptotic theory underpinning the likelihood ratio test does not apply in cases of non-identifiability [56], which is the case for NODEs and UDEs, at least concerning the NN parameters. While extensions have been proposed to characterize the asymptotic distribution of the likelihood ratio test under specific forms of non-identifiability [57], these have not been considered in this work, and the assessment of a statistically meaningful threshold remains beyond the scope of the present study.

Selection of state space region and stopping criterion. The final major limitation of the current work lies in the arbitrary selection of the state space region in which model disagreement is optimized, as well as the lack of a stopping criterion during the training of individual models. Our results show that maximizing disagreement in a specific region of the state space leads to increased disagreement and improved prediction interval coverage compared to standard ensembles, even across a broader portion of the vector field. However, uncertainty estimation is sensitive to the choice of region, and a single MOD ensemble may be suboptimal when applied to regions of the state space different from where the model disagreement was maximized. Furthermore, in our test cases, the ensemble models were trained for a fixed number of iterations without a convergence criterion or a threshold to stop the training. An analysis of the maximization curve for model disagreement during training is provided in Supplementary Section S20, which reveals key challenges in defining a threshold on the objective

function or a convergence criterion. First, the values assumed by the objective function are dependent on the system considered and on the OOD region in which the disagreement is maximized, thus preventing the introduction of a fixed threshold to stop the training of models in MOD ensembles. Second, no clear maximum in model disagreement may exist, leading to a failure in optimization convergence.

Despite these limitations, the proposed work represents an initial step toward extending traditional UQ methods to address OOD UQ in dynamical system reconstruction. In future work, we aim to extend the method in three key directions. First, we will incorporate aleatoric uncertainty into the training dataset and adapt the Hessian matrix approximation method to handle noise in the training data. This will be achieved either through the use of suggested numerical methods (e.g., [54]) or by employing automatic differentiation techniques. Second, we will focus on incorporating the qualitative characteristics of the vector field into the ensemble training process. The analysis presented in Supplementary Section S21 shows that our training algorithm currently produces models that fail to respect key qualitative features of the vector field, such as periodicity in the Lotka-Volterra case or the existence of a global attractor in the Damped Oscillator case—features that a modeler could inductively deduce. To address this, we plan to introduce these characteristics as soft constraints in the loss function. Finally, we aim to address the sequential nature of our algorithm by introducing parallelization in the training process of the models composing the MOD ensemble. Specifically, we plan to train the ensemble models simultaneously, maximizing the disagreement between them during training.

4 Methods

4.1 Training sets

For each dynamical system considered, three trajectories were generated by numerically integrating the equations defined in Eqs. (3), (4), and (5), using distinct initial conditions in each case. Each trajectory was discretized into 100 equally spaced time intervals, resulting in 100 sample points per trajectory (beyond the initial condition). Consequently, the training dataset for each system comprises 300 data points. The specific initial conditions, integration time intervals, and parameter values used for each dynamical system are provided in Subsections 4.1.1, 4.1.2, and 4.1.3.

4.1.1 Lotka-Volterra

The three ground truth trajectories for the Lotka-Volterra system were obtained by numerically integrating the equations defined in Eq. (3), using the parameter values $\alpha = 1.3$, $\beta = 0.9$, $\gamma = 0.8$, and $\delta = 1.8$, as in [13]. Integration was carried out over the time interval $[0.0, 10.0]$ with the following initial conditions:

$$\begin{aligned}
&\text{Trajectory 1: } x(0) = 3.00, \quad y(0) = 1.50,\\
&\text{Trajectory 2: } x(0) = 3.38, \quad y(0) = 1.38,\\
&\text{Trajectory 3: } x(0) = 3.28, \quad y(0) = 1.58.
\end{aligned}$$

4.1.2 Damped oscillator

The three ground truth trajectories for the Damped oscillator test case were obtained by numerically integrating the system defined in Eq. (4) with the parameters $\alpha = 0.1$ and $\beta = 2.0$, as used in [40]. The integration was performed over the time interval $[0.0, 25.0]$, using the following initial conditions:

$$\begin{aligned}
&\text{Trajectory 1: } x(0) = 1.00, \quad y(0) = 1.00 \\
&\text{Trajectory 2: } x(0) = 1.06, \quad y(0) = 1.20 \\
&\text{Trajectory 3: } x(0) = 0.98, \quad y(0) = 1.09.
\end{aligned}$$

4.1.3 Lorenz

The three ground truth trajectories for the Lorenz test case were generated by numerically integrating the system defined in Eq. (5), using the canonical chaotic parameter values $\sigma = 10.0$, $r = 28.0$, and $b = \frac{8}{3}$, which induce sensitive dependence on initial conditions and complex system dynamics [58]. The integration was performed over the time interval $[0.0, 2.0]$, with the following initial conditions:

$$\begin{aligned}
&\text{Trajectory 1: } x(0) = 1.47, \quad y(0) = -4.44, \quad z(0) = 28.71 \\
&\text{Trajectory 2: } x(0) = 1.43, \quad y(0) = -4.85, \quad z(0) = 30.42 \\
&\text{Trajectory 3: } x(0) = 1.76, \quad y(0) = -4.41, \quad z(0) = 32.08.
\end{aligned}$$

4.2 Metrics

Uncertainty quantification is carried out on two distinct yet related outputs of the model: the vector field and the trajectories generated by the dynamical system. The methods used to construct prediction intervals from an ensemble of models, as well as the evaluation of their OOD accuracy in both scenarios, are described in detail in Subsections 4.2.1 and 4.2.2. In this and the following subsections, we will extend the notation used in Section 2.3. We denote by

$$\frac{d\mathbf{y}}{dt} = f(\mathbf{y}), \tag{10}$$

with $\mathbf{y} \in \mathbb{R}^n$, the ground-truth model to be reconstructed, and by $\mathbf{y}(t; t_0, \mathbf{y}_0)$ the state of the ground-truth system at time t, obtained from the initial condition $(t_0, \mathbf{y}_0)$, with $t > t_0$, given by

$$\mathbf{y}(t; t_0, \mathbf{y}_0) = \mathbf{y}_0 + \int_{t_0}^{t} f(\mathbf{y}(\tau))\, d\tau.$$

We denote by $y^j(t; t_0, \mathbf{y}_0)$ the j-th component of $\mathbf{y}(t; t_0, \mathbf{y}_0)$, i.e., the state of the j-th variable of the system at time t. We consider an ensemble of p UDE models trained to approximate the dynamics of the ground-truth system, that is, p parameterizations $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p \in \mathbb{R}^{k+l}$, each trained to reproduce the training data describing the time evolution of the dynamical system in Eq. 10. For a fixed parameterization $\overline{\boldsymbol{\theta}}$, we denote

by

$$\hat{\mathbf{y}}(t, \overline{\boldsymbol{\theta}}; t_0, \mathbf{y}_0) = \mathbf{y}_0 + \int_{t_0}^{t} g(\mathbf{y}(\tau), \overline{\boldsymbol{\theta}})\, d\tau$$

the prediction of the UDE model at time t, obtained by integrating the system from the initial condition $(t_0, \mathbf{y}_0)$, with $t > t_0$. We further indicate with $\hat{y}^j(t, \overline{\boldsymbol{\theta}}; t_0, \mathbf{y}_0)$ the j-th component of the predicted state.

4.2.1 Uncertainty on vector field

Given a point in the state space $\mathbf{z} \in \mathbb{R}^n$, the prediction intervals for the value of the vector field in $\mathbf{z}$, $f(\mathbf{z})$, are computed by extending to the multivariate setting the procedure proposed for the univariate case in NN ensembles [29]. Fixed a level $\alpha \in (0, 1)$, the α-prediction interval on $f(\mathbf{z})$ is computed, following [59], as the ellipsoidal region:

$$PE^{\alpha}(f(\mathbf{z})) = \left\{ \mathbf{w} \in \mathbb{R}^n \;\middle|\; (\mathbf{w} - \boldsymbol{\mu})^{\top} \boldsymbol{\Sigma}^{-1} (\mathbf{w} - \boldsymbol{\mu}) \leq \chi^2_D(\alpha) \right\},$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the sample mean and covariance matrix of the ensemble predictions $g(\mathbf{z}, \boldsymbol{\theta}_1), \ldots, g(\mathbf{z}, \boldsymbol{\theta}_p)$, respectively. The term $\chi^2_D(\alpha)$ denotes the α-quantile of the chi-squared distribution with D degrees of freedom, corresponding to the number of vector field components being regressed. In our work, we use $\alpha = 0.95$ and set $D = n$ for every test case, except for the Lorenz system in the UDE with fixed physical parameters. In this case, we use $D = 1$ and $n = 3$, given that the estimation process involves parameters from only a single equation in the model.

In each test case, the OOD region where the prediction intervals are evaluated is constructed by symmetrically expanding the bounding box of the training trajectories in the state space by 20% along each coordinate axis (see Fig. 10). The metrics related to coverage of prediction intervals, described in Section 4.2, are computed on 50000 points sampled uniformly from this region.

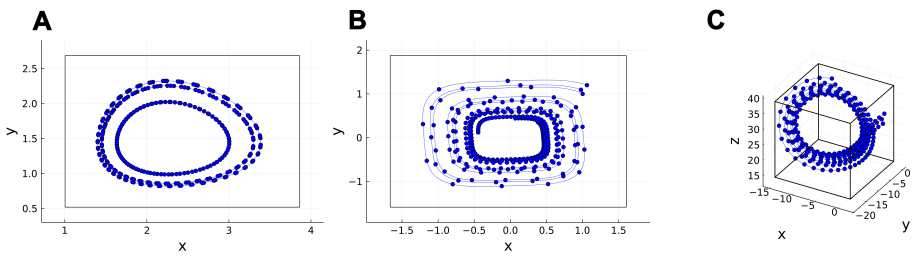

Fig. 10: **Training trajectories and OOD scenarios on vector fields.** Phase portraits of sampled training trajectories for the Lotka-Volterra, Damped Oscillator, and Lorenz dynamical systems are shown in panels A, B, and C, respectively. The black contours indicate the OOD regions used for computing UQ metrics on the vector field.

4.2.2 Uncertainty on trajectories

The prediction intervals on the ground truth trajectories of the dynamical system in Eq. (10) are computed individually for each variable of the dynamical system. For a

fixed $\alpha \in (0,1)$, the α-prediction interval for the j-th variable at time t is computed as

$$PE^{\alpha}\big(y^j(t;t_0,\mathbf{y}_0)\big) = \left\{ w \in \mathbb{R} \;\middle|\; |w-\mu| \leq t_{p-1,\,(1+\alpha)/2} \cdot \sigma \cdot \sqrt{1+\frac{1}{p}} \right\},$$

where μ and σ denote the sample mean and standard deviation of the ensemble predictions $\hat{y}^j(t,\boldsymbol{\theta}_1;t_0,\mathbf{y}_0),\ldots,\hat{y}^j(t,\boldsymbol{\theta}_p;t_0,\mathbf{y}_0)$, and $t_{p-1,\,(1+\alpha)/2}$ is the $(1+\alpha)/2$ quantile of the Student distribution with $p-1$ degrees of freedom [60].

For each dynamical system, we generated 100 test trajectories by randomly selecting one of the training trajectories and perturbing its initial condition. The perturbation was sampled from a normal distribution with mean 0 and coefficient of variation 0.5. The integration time of these test trajectories matches that of the training trajectories for the Lotka-Volterra and Damped Oscillator systems. For the Lorenz system, however, the integration time was reduced to 5% of the training time interval, due to its chaotic dynamics and the numerical instability encountered in the data-driven reconstruction when integrating over longer time horizons (see Fig. 11). From each trajectory, 10^2 points were sampled at regular time intervals. On these points, the metrics described in Section 4.2—related to the coverage of prediction intervals—were computed.

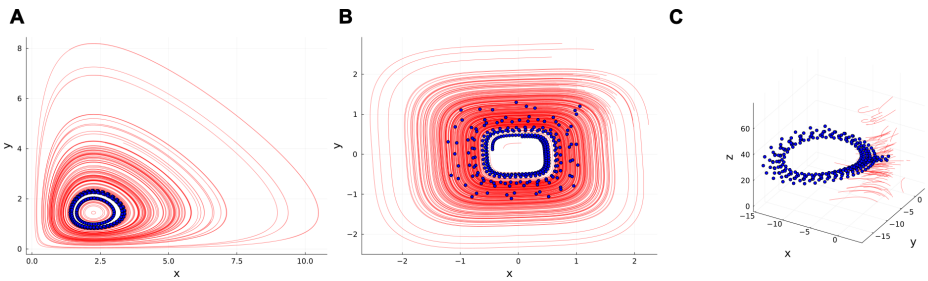

Fig. 11: **Training trajectories and OOD UQ on trajectories.** The trajectories used for UQ on trajectories are shown in red for the Lotka-Volterra, Damped Oscillator, and Lorenz dynamical systems are shown in panels A, B, and C, respectively. The sampled training trajectories are shown in blue.

4.3 Standard ensemble training

This section provides a detailed description of the procedure used to train the models of the standard ensembles. Each model was trained following the same process, outlined in Subsection 4.3.1, with the only variation being the random initialization of the model parameters (the initialization strategy is explained in Subsection 4.3.2). Since training NODEs and UDEs can encounter convergence issues and numerical instabilities, this may result in parameterizations with suboptimal fit to the training data. The criterion used to accept the models is described in Subsection 4.3.3.

4.3.1 Training process

The models comprising the standard ensembles were trained to fit the dynamics of the three training trajectories described in Section 4.1. In this section, we describe the procedure used to train a model on a single trajectory. In the case of simultaneous training on multiple trajectories, the overall cost and validation functions are computed as the sum of the corresponding cost and validation functions evaluated on each individual trajectory. The hyperparameters employed for each test case and scenario were manually tuned, and are provided—along with other algorithmic details—in Supplementary Section S22.

Given a time series of observed system states $\{(t_i, \mathbf{y}_i)\}_{i=0}^{r}$, with $\mathbf{y}_i \in \mathbb{R}^n$, we randomly split the indices $\{1, \dots, r\}$ into disjoint training and validation sets

$$\mathcal{T} = \{T_1, \dots, T_k\} \qquad \text{and} \qquad \mathcal{V} = \{V_1, \dots, V_s\},$$

respectively, according to an 80:20 ratio. Let y_i^j denote the j-th component of the observed state $\mathbf{y}_i$. To account for differences in variable magnitudes, we define Γ^j as the maximum variation of the j-th component across the trajectory:

$$\Gamma^j := \max_{i=1,\dots,r} y_i^j - \min_{i=1,\dots,r} y_i^j.$$

The normalized mean-squared error loss functions for training and validation used are:

$$\mathcal{L}_{\text{train}}(\boldsymbol{\theta}) = \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{n} \left(\frac{\hat{y}^j(t_{T_i}, \boldsymbol{\theta}; t_0, \mathbf{y}_0) - y_{T_i}^j}{\Gamma^j} \right)^2 \tag{11}$$

$$\mathcal{L}_{\text{val}}(\boldsymbol{\theta}) = \frac{1}{s} \sum_{i=1}^{s} \sum_{j=1}^{n} \left(\frac{\hat{y}^j(t_{V_i}, \boldsymbol{\theta}; t_0, \mathbf{y}_0) - y_{V_i}^j}{\Gamma^j} \right)^2 \tag{12}$$

Since minimizing L_{train} directly is often suboptimal due to the potential risks of the training getting stuck in local minima [61], we articulate the training process in two different steps, which are detailed below.

Preliminary step. In the first step, we use the multiple shooting technique [62] to initially optimize the fit of the model to the training set. This involves segmenting the training time series into segments of k points and optimizing the fit of the model on these different segments. The cost function in this case is the weighted sum of the sum of cost functions on the single segments plus a penalty term to promote the continuity of the different segments (we refer to [62] for details). This cost function is optimized with Adam [63] for 1000 epochs. In cases where the optimization is conducted on both the NN parameters and the mechanistic parameters of the system, the mechanistic parameters have been normalized within their boundaries.

Refinement. The result from this initial step is refined in the second step, where the training loss function on the whole trajectory, as defined in Eq. (11), is optimized using the L-BFGS algorithm [64]. This optimization is performed with a maximum

of 500 epochs, utilizing early stopping based on the validation loss to prevent overfitting to the training dataset. Additionally, in cases where optimization involves both NN parameters and mechanistic system parameters, the mechanistic parameters are normalized within their respective bounds to ensure consistent and stable training.

4.3.2 Parameter initialization

The NN parameters are initialized using Xavier initialization [65], while the mechanistic parameters (in the case of a partial data-driven model) are randomly generated within their boundaries.

4.3.3 Acceptance criteria

The optimization results are accepted, meaning the parameters are selected to be part of an ensemble, if

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{r}\sum_{i=1}^{r}\sum_{j=1}^{n}\left(\frac{\hat{y}^j(t_{V_i}, \boldsymbol{\theta}; t_0, \mathbf{y}_0) - y^j_{V_i}}{\Gamma^j}\right)^2 \leq \epsilon_{\text{acc}} \tag{13}$$

where ϵ_{acc} is a tolerance threshold. In the experiments presented in Section 2.2, we set $\epsilon_{\text{acc}} = 10^{-3}$.

4.4 MOD ensemble training

This section provides a detailed description of the algorithm used to train MOD ensembles, which was introduced at a high level in Section 2.3. Subsection 4.4.1 explains how to compute the directions along which the loss remains locally constant, Subsection 4.4.2 presents the algorithm for performing the constrained optimization, and Subsection 4.4.4 describes the complete MOD training procedure.

4.4.1 Computation of the loss function Hessian

The loss function $\mathcal{L}$, computed over the entire training trajectories, is the same as the one used in the acceptance criterion for standard ensemble training, as defined in Eq. (13). We approximate the Hessian matrix H of $\mathcal{L}$ at a point $\overline{\boldsymbol{\theta}}$ using the Gauss-Newton method. Specifically, the approximation is given by:

$$H^{u,t}_{\mathcal{L}}(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{\Gamma_i^2}\frac{\partial y^j}{\partial \theta_k}(t_i, \overline{\boldsymbol{\theta}}, \mathbf{y}_0)\cdot\frac{\partial y^j}{\partial \theta_l}(t_i, \overline{\boldsymbol{\theta}}, \mathbf{y}_0), \quad \text{for } 1 \leq u, t \leq k+l, \tag{14}$$

where θ_l denotes the l-th component of the parameter vector.

When computed numerically (under finite-precision arithmetic), the eigenvalues of $H_{\mathcal{L}}(\boldsymbol{\theta})$ are always strictly positive. To identify directions in the null space $H^0_{\mathcal{L}}(\boldsymbol{\theta})$, we approximate zero eigenvalues by applying a threshold ϵ_{eig}, such that eigenvalues smaller than ϵ are considered null. To improve computational efficiency, we do not compute the Hessian over all n training points $\{(t_i, \mathbf{y}_i)\}_{i=1}^n$, but instead sample a random subset of them.

4.4.2 How to route loss contours

Given $p-1$ models in an ensemble (with $p > 1$), i.e., $p-1$ parameterizations $\mathbf{b}_1, \ldots, \mathbf{b}_{p-1}$, the p-th model parameterization $\mathbf{b}_p$ is initialized at $\mathbf{b}_1$. We first perturb it randomly along the null space of the cost function $H^0_{\mathcal{L}}(\mathbf{b}_1)$, and then perform a fixed number of iterations n_{iter}. In each iteration, as described in Section 2.3, we update the parameters $\mathbf{b}_p$ according to the direction $\mathbf{w}$, defined by the projection of the gradient of the disagreement

$$\nabla_\theta \operatorname{dis}_S(\mathbf{b}_1, \ldots, \mathbf{b}_{p-1}, \theta)\big|_{\theta=\mathbf{b}_p}$$

onto $H^0_{\mathcal{L}}(\mathbf{b}_p)$.

Since the direction $\mathbf{w}$ guarantees no increase in the loss function $\mathcal{L}$ only locally, we introduce a threshold $\epsilon_{\mathcal{L}}$ to define the maximum allowable value of $\mathcal{L}$ for ensemble members. At each parametersTo ensure that optimization remains within this minimum-loss region, we adopt two complementary strategies. First, at each iteration of the MOD training algorithm, the parameters are updated along the direction $\mathbf{w}$ using an adaptive learning rate. Specifically, the learning rate for updating $\mathbf{b}_p$ along $\mathbf{w}$ is dynamically chosen so that it is as large as possible without causing the loss to increase by more than a prescribed threshold ϵ_{step}. This adaptive rule allows us to control the learning rate at each iteration to remain on the level sets of $\mathcal{L}$. Second, if after an update, even with a minimum learning rate, the value $\mathcal{L}(\mathbf{b}_p)$ exceeds the threshold ϵ_{acc}, we perform a short optimization phase to explicitly minimize $\mathcal{L}$, effectively re-projecting $\mathbf{b}_p$ back onto the minimum-loss region. These two mechanisms work in tandem to guide the optimization along the level curves of the loss while maintaining the desired loss constraint.

4.4.3 Maximizing the disagreement over a region

Given a selected region of the state space and a finite set of points S within this region, the definition of model disagreement introduced in Section 2.3 provides a quantitative measure of ensemble disagreement over S. Since the disagreement on S is defined as the average disagreement across all points in the set, directly maximizing this quantity poses a risk: it may lead the optimization to focus disproportionately on a small subset of points where disagreement is already large, without improving diversity across the entire region. To mitigate this issue, at each optimization step, we randomly sample a subset of S to estimate the gradient

$$\nabla_\theta \operatorname{dis}_S(\mathbf{b}_1, \ldots, \mathbf{b}_{p-1}, \theta)\big|_{\theta=\mathbf{b}_p}.$$

4.4.4 Overall algorithm description

The overall algorithm is summarized in Algorithm 1. For each model in the ensemble, we optimize only a subset of the NN parameters in $\boldsymbol{\theta}$, in order to further encourage diversity among ensemble members.

In the case of hybrid models where $\boldsymbol{\theta}$ includes both mechanistic and NN parameters (as in the third scenario in Section 2.4), the mechanistic parameters are normalized with respect to their initial values—i.e., the values optimized in $\mathbf{b}_1$.

Algorithm 1 MOD Ensemble Training Procedure

Require: Training trajectories; ensemble size p; evaluation set of OOD state-space points $S \subset \mathbb{R}^n$; number of steps n_{iter}; threshold for near-null Hessian eigenvalues ϵ_{eig}; loss threshold ϵ_{acc}; maximum allowed loss increase per step ϵ_{step}.

Ensure: Ensemble $\{\mathbf{b}_1, \ldots, \mathbf{b}_p\}$ with maximized OOD disagreement

Train $\mathbf{b}_1$ by minimizing the loss $\mathcal{L}$ on the training trajectories.
for $j = 2$ **to** p **do**
  Select a subset of model parameters $\boldsymbol{\theta}' \subset \boldsymbol{\theta}$ to optimize.
  Approximate Hessian $H \leftarrow \nabla_\theta^2 \mathcal{L}(\mathbf{b}_1)$ via Gauss–Newton.
  Compute the null subspace H_0 of H using eigenvectors with $\lambda \leq \epsilon_{\text{eig}}$.
  Create $\mathbf{b}_1^{\text{pert}}$ by perturbing $\mathbf{b}_1$ along a random unit vector in H_0 (restricted to $\boldsymbol{\theta}'$).
  Initialize $\mathbf{b}_j \leftarrow \mathbf{b}_1^{\text{pert}}$.
  for $t = 1$ **to** n_{iter} **do**
    Sample a random subset of OOD points $S' \subset S$.
    Compute gradient $v \leftarrow \nabla_\theta \operatorname{dis}_{S'}(\mathbf{b}_1, \ldots, \mathbf{b}_{j-1}, \theta)\big|_{\theta=\mathbf{b}_j}$.
    Approximate Hessian $H \leftarrow \nabla_\theta^2 \mathcal{L}(\mathbf{b}_j)$ via Gauss–Newton.
    Compute the null subspace H_0 of H using eigenvectors with $\lambda \leq \epsilon_{\text{eig}}$.
    Project v onto H_0: $\tilde{v} \leftarrow \pi_{H_0}(v)$.
    Choose step size η so that $\left|\mathcal{L}(\mathbf{b}_j + \eta\tilde{v}) - \mathcal{L}(\mathbf{b}_j)\right| < \epsilon_{\text{step}}$.
    Update $\mathbf{b}_j \leftarrow \mathbf{b}_j + \eta\,\tilde{v}$ (restricted to $\boldsymbol{\theta}'$).
    if $\mathcal{L}(\mathbf{b}_j) > \epsilon_{\text{acc}}$ **then**
      Reproject $\mathbf{b}_j$ by running a few optimization steps on $\mathcal{L}$ (restricted to $\boldsymbol{\theta}'$).
      if $\mathcal{L}(\mathbf{b}_j) > \epsilon_{\text{acc}}$ **then**
        continue
      else
        break ▷ Proceed to the next outer iteration
      end if
    end if
  end for
end for
return $\{\mathbf{b}_1, \ldots, \mathbf{b}_p\}$

4.5 Algorithmic setup

The results reported in the main text of the paper were obtained by running the MOD ensemble training algorithm with ϵ_{acc} and $n_{\text{iter}} = 800$. The step-size threshold ϵ_{step} was set as $\epsilon_{\text{acc}} \cdot 10^{-1} = 10^{-4}$. At each iteration, the Hessian of the cost function

was computed on a randomly sampled subset of 20 time points, selected from the total 100 time points available for each trajectory. Each ensemble member was trained by optimizing 200 NN parameters, plus mechanistic parameters if applicable. The finite set S, used to maximize model disagreement, was constructed as a regular grid over the OOD region of the state space. Specifically, we used a 10×10 grid for the Lotka–Volterra and the Damped Oscillator test cases, and a $5 \times 5 \times 5$ grid for the Lorenz system. At each iteration, a random subset comprising 25% of the grid points was used to compute the gradient of the disagreement. The parameter projection, applied when the loss on the training trajectory exceeded $\epsilon_{\mathcal{L}}$, was carried out by optimizing the loss function using 20 iterations of gradient descent, followed by 20 iterations of the L-BFGS algorithm. The threshold to identify near-null eigenvectors was fixed at $\epsilon_{eig} = 10^{-1}$.

The results presented in the Supplementary Material, which were obtained with a lower loss threshold, were generated using $\epsilon_{\text{acc}} = 10^{-4}$ and a correspondingly updated step-size threshold $\epsilon_{\text{step}} = 10^{-5}$. In this case, due to the smaller step sizes, the number of training steps was increased to $n_{\text{iter}} = 1200$ for each ensemble member. All the other settings are the same as in the ones reported in the main text.

4.6 Implementation

All the computations have been run on a Debian-based Linux cluster node, with two 24-core Xeon(R) CPU X5650 CPUs and 250 GB of RAM. The code is implemented in Julia v1.10.5 [66], using the SCiML environment [13], a software suite for modeling and simulation that also incorporates ML algorithms.

5 Data availability

No experimental data were used as part of this study. The *in silico* generated datasets are available at https://github.com/cosbi-research/UQNU.

6 Code availability

The scripts to reproduce the results are available at https://github.com/cosbi-research/UQNU.

References

[1] Vera, J., Wolkenhauer, O.: A system biology approach to understand functional activity of cell communication systems. Methods in cell biology **90**, 399–415 (2008)

[2] DiStefano, J.J.: Dynamic Systems Biology Modeling and Simulation, First edition. edn. Elsevier, Academic Press, Amsterdam (2013)

[3] Procopio, A., Cesarelli, G., Donisi, L., Merola, A., Amato, F., Cosentino, C., *et al.*: Combined mechanistic modeling and machine-learning approaches in systems

biology-a systematic literature review. Comput. Methods Programs Biomed. **240**, 107681 (2023)

[4] Yu, R., Wang, R.: Learning dynamical systems from data: An introduction to physics-guided deep learning. Proceedings of the National Academy of Sciences **121**(27), 2311808121 (2024)

[5] Vasiliauskaite, V., Antulov-Fantulin, N.: Generalization of neural network models for complex network dynamics. Communications Physics **7**(1), 348 (2024)

[6] Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proceedings of the National Academy of Sciences **113**(15), 3932–3937 (2016)

[7] Navarro-Moreno, J.: Arma prediction of widely linear systems by using the innovations algorithm. IEEE Transactions on Signal Processing **56**(7), 3061–3068 (2008)

[8] Van Houdt, G., Mosquera, C., Nápoles, G.: A review on the long short-term memory model. Artificial Intelligence Review **53**(8), 5929–5955 (2020)

[9] Funahashi, K.-i., Nakamura, Y.: Approximation of dynamical systems by continuous time recurrent neural networks. Neural networks **6**(6), 801–806 (1993)

[10] Aussem, A.: Dynamical recurrent neural networks towards prediction and modeling of dynamical systems. Neurocomputing **28**(1-3), 207–232 (1999)

[11] Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. Advances in Neural Information Processing Systems **31** (2018)

[12] Scarselli, F., Tsoi, A.C.: Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. Neural networks **11**(1), 15–37 (1998)

[13] Rackauckas, C., Ma, Y., Martensen, J., Warner, C., Zubov, K., Supekar, R., Skinner, D., Ramadhan, A., Edelman, A.: Universal differential equations for scientific machine learning. arXiv preprint arXiv:2001.04385 (2020)

[14] Xiang, J., Qi, B., Cerou, M., Zhao, W., Tang, Q.: Dn-ode: Data-driven neural-ode modeling for breast cancer tumor dynamics and progression-free survivals. Computers in Biology and Medicine **180**, 108876 (2024)

[15] Erbe, R., Stein-O'Brien, G., Fertig, E.J.: Transcriptomic forecasting with neural ordinary differential equations. Patterns **4**(8) (2023)

[16] Hossain, I., Fanfani, V., Fischer, J., Quackenbush, J., Burkholz, R.: Biologically informed neuralodes for genome-wide regulatory dynamics. Genome Biology

25(1), 127 (2024)

[17] Rooij, M., Erdős, B., Riel, N.A., O'Donovan, S.D.: Physiology-informed regularisation enables training of universal differential equation systems for biological applications. PLOS Computational Biology **21**(1), 1012198 (2025)

[18] Göring, N., Hess, F., Brenner, M., Monfared, Z., Durstewitz, D.: Out-of-domain generalization in dynamical systems reconstruction. In: Proceedings of the 41st International Conference on Machine Learning. ICML'24, pp. 16071–16114. JMLR.org, Vienna, Austria (2024)

[19] Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W., Yu, P.S.: Generalizing to unseen domains: A survey on domain generalization. IEEE Transactions on Knowledge and Data Engineering **35**(8), 8052–8072 (2022)

[20] Liu, J., Shen, Z., He, Y., Zhang, X., Xu, R., Yu, H., Cui, P.: Towards out-of-distribution generalization: A survey. arXiv preprint arXiv:2108.13624 (2021)

[21] Puniya, B.L., Verma, M., Damiani, C., Bakr, S., Dräger, A.: Perspectives on computational modeling of biological systems and the significance of the SysMod community. Oxford University Press (2024)

[22] Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., Snoek, J.: Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. Advances in Neural Information Processing Systems **32** (2019)

[23] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U.R., *et al.*: A review of uncertainty quantification in deep learning: Techniques, applications and challenges. Information Fusion **76**, 243–297 (2021)

[24] Dandekar, R., Chung, K., Dixit, V., Tarek, M., Garcia-Valadez, A., Vemula, K.V., Rackauckas, C.: Bayesian neural ordinary differential equations. arXiv preprint arXiv:2012.07244 (2020)

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

[26] Zou, Z., Meng, X., Psaros, A.F., Karniadakis, G.E.: Neuraluq: A comprehensive library for uncertainty quantification in neural differential equations and operators. SIAM Review **66**(1), 161–190 (2024)

[27] Takeishi, N.: Toward Bayesian Deep Grey-box Modeling. In: International Conference on Scientific Computing and Machine Learning, Kyoto (2024)

[28] Schmid, N., Pozo, D., Waegeman, W., Hasenauer, J.: Assessment of uncertainty

quantification in universal differential equations. Philosophical Transactions A **383**(2293), 20240444 (2025)

[29] Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in Neural Information Processing Systems **30** (2017)

[30] Caldeira, J., Nord, B.: Deeply uncertain: comparing methods of uncertainty quantification in deep learning algorithms. Machine Learning: Science and Technology **2**(1), 015002 (2020)

[31] Gustafsson, F.K., Danelljan, M., Schon, T.B.: Evaluating scalable bayesian deep learning methods for robust computer vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 318–319 (2020)

[32] Scalia, G., Grambow, C.A., Pernici, B., Li, Y.-P., Green, W.H.: Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. Journal of Chemical Information and Modeling **60**(6), 2697–2717 (2020)

[33] Van Amersfoort, J., Smith, L., Teh, Y.W., Gal, Y.: Uncertainty estimation using a single deep deterministic neural network. In: International Conference on Machine Learning, pp. 9690–9700 (2020)

[34] Mehrtens, H.A., González, C., Mukhopadhyay, A.: Improving robustness and calibration in ensembles with diversity regularization. In: DAGM German Conference on Pattern Recognition, pp. 36–50 (2022)

[35] Jain, S., Liu, G., Mueller, J., Gifford, D.: Maximizing overall diversity for improved uncertainty estimates in deep ensembles. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 4264–4271 (2020)

[36] Bonnabel, S.: Stochastic gradient descent on riemannian manifolds. IEEE Transactions on Automatic Control **58**(9), 2217–2229 (2013)

[37] Garipov, T., Izmailov, P., Podoprikhin, D., Vetrov, D.P., Wilson, A.G.: Loss surfaces, mode connectivity, and fast ensembling of dnns. Advances in Neural Information Processing Systems **31** (2018)

[38] Draxler, F., Veschgini, K., Salmhofer, M., Hamprecht, F.: Essentially no barriers in neural network energy landscape. In: International Conference on Machine Learning, pp. 1309–1318. PMLR, Stockholm, Sweden (2018)

[39] Daniel Freeman, C., Bruna, J.: Topology and Geometry of Half-rectified Network Optimization. (2017)

[40] Fronk, C., Yun, J., Singh, P., Petzold, L.: Bayesian polynomial neural networks and polynomial neural ordinary differential equations. PLOS Computational Biology **20**(10), 1012414 (2024)

[41] Roesch, E., Rackauckas, C., Stumpf, M.P.: Collocation based training of neural ordinary differential equations. Statistical applications in genetics and molecular biology **20**(2), 37–49 (2021)

[42] Fronk, C., Petzold, L.: Interpretable polynomial neural ordinary differential equations. Chaos: An Interdisciplinary Journal of Nonlinear Science **33**(4) (2023)

[43] Duarte, P., Fernandes, R.L., Oliva, W.M.: Dynamics of the attractor in the lotka–volterra equations. Journal of Differential Equations **149**(1), 143–189 (1998)

[44] Baines, P.G.: Lorenz, e.n. 1963: Deterministic nonperiodic flow. journal of the atmospheric sciences 20, 130–41.1. Progress in Physical Geography: Earth and Environment **32**(4), 475–480 (2008)

[45] Hirsch, M.W., Smale, S., Devaney, R.L.: Differential Equations, Dynamical Systems, and an Introduction to Chaos. Academic press, Amsterdam, Boston, Heidelberg, London, New York, Oxford, Paris, San Diego, San Francisco, Singapore, Sydney, Tokyo (2013)

[46] Williams, R.F.: The structure of lorenz attractors. Publications Mathematiques de l'IHES **50**, 73–99 (1979)

[47] Sluijterman, L., Cator, E., Heskes, T.: How to evaluate uncertainty estimates in machine learning for regression? Neural Networks **173**, 106203 (2024)

[48] Giampiccolo, S., Reali, F., Fochesato, A., Iacca, G., Marchetti, L.: Robust parameter estimation and identifiability analysis with hybrid neural ordinary differential equations in computational biology. NPJ Systems Biology and Applications **10**(1), 139 (2024)

[49] Simsek, B., Ged, F., Jacot, A., Spadaro, F., Hongler, C., Gerstner, W., Brea, J.: Geometry of the loss landscape in overparameterized neural networks: Symmetries and invariances. In: International Conference on Machine Learning, pp. 9722–9732 (2021). PMLR

[50] Barreiro, X.R., Villaverde, A.F.: On the origins and rarity of locally but not globally identifiable parameters in biological modeling. IEEE Access **11**, 65457–65467 (2023)

[51] Wilcox, R.R.: Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy vol. 249. Springer, New York, NY (2001)

[52] Cooper, Y.: The loss landscape of overparameterized neural networks. arXiv

preprint arXiv:1804.10200 (2018)

[53] Hüllermeier, E., Waegeman, W.: Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. Machine learning **110**(3), 457–506 (2021)

[54] Bartholomew-Biggs, M.C.: The estimation of the hessian matrix in nonlinear least squares problems with non-zero residuals. Mathematical programming **12**(1), 67–80 (1977)

[55] Villaverde, A.F., Raimúndez, E., Hasenauer, J., Banga, J.R.: Assessment of prediction uncertainty quantification methods in systems biology. IEEE/ACM transactions on computational biology and bioinformatics **20**(3), 1725–1736 (2022)

[56] Lindsay, B.G.: Mixture models: theory, geometry, and applications. (1995). Ims

[57] Liu, X., Shao, Y.: Asymptotics for likelihood ratio tests under loss of identifiability. The Annals of Statistics **31**(3), 807–832 (2003)

[58] Grigorenko, I., Grigorenko, E.: Chaotic dynamics of the fractional lorenz system. Physical Review Letters **91**(3), 034101 (2003)

[59] Giri, N.C.: Multivariate Statistical Inference. Academic Press, New York, San Francisco, London (1977)

[60] Gourieroux, C., Monfort, A.: Statistics and Econometric Models vol. 1. Cambridge University Press, Cambridge, England (1995)

[61] Kidger, P.: On neural differential equations. arXiv preprint arXiv:2202.02435 (2022)

[62] Turan, E.M., Jäschke, J.: Multiple shooting for training neural differential equations on time series. IEEE Control Systems Letters **6**, 1897–1902 (2021)

[63] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)

[64] Nash, S.G., Nocedal, J.: A numerical study of the limited memory bfgs method and the truncated-newton method for large scale optimization. SIAM Journal on Optimization **1**(3), 358–372 (1991)

[65] Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 249–256 (2010). JMLR Workshop and Conference Proceedings

[66] Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to

numerical computing. SIAM review **59**(1), 65–98 (2017)