Geophysical Journal International

Geophys. J. Int. (2026) **244,** 1–21
Advance Access publication 2025 December 16
Research Paper

https://doi.org/10.1093/gji/ggaf518

Data assimilation in machine-learned reduced-order model of chaotic earthquake sequences

Hojjat Kaveh, Jean Philippe Avouac and Andrew M. Stuart

Department of Mechanical and Civil Engineering, California Institute of Technology, 1200 E. California Blvd., Pasadena, CA 91125, USA. E-mail: hkaveh@caltech.edu

Accepted 2025 December 9. Received 2025 November 6; in original form 2025 July 22

SUMMARY

Realistic models of earthquake sequences can be simulated by assuming faults governed by rate-and-state friction embedded in an elastic medium. Exploring the possibility of using such models for earthquake forecasting is challenging due to the difficulty of integrating partial differential equation models with sparse, low-resolution observational data. This paper presents a machine-learning-based reduced-order model (ROM) for earthquake sequences that addresses this limitation. The proposed ROM captures the slow/fast chaotic dynamics of earthquake sequences using a low-dimensional representation, enabling computational efficiency and robustness to high-frequency noise in observational data. The ROM's efficiency facilitates effective data assimilation using the Ensemble Kalman Filter, even with low-resolution, noisy observations. Results demonstrate the ROM's ability to replicate key scaling properties of the sequence—namely the magnitude–frequency, moment–duration, and moment–area relationships—and to estimate the distributions of fault slip rate and state variable, enabling predictions of large events in time and space with uncertainty quantification. These findings underscore the ROM's potential for forecasting and for addressing challenges in inverse problems for nonlinear geophysical systems.

Key words: Seismic cycle; Inverse theory; Machine learning; Earthquake interaction, forecasting, and prediction.

1 INTRODUCTION

Earthquake occurrence arises from complex, nonlinear fault system dynamics, and their predictability remains an open question (I. Main 1996). Recent progress has been made in forecasting the spatial and temporal variations of earthquake rates (E.H. Field *et al.* 2015; D. Dempsey & J. Suckale 2017; H. Kaveh *et al.* 2023) and in developing realistic models of earthquake sequences that simulate faults obeying rate-and-state friction (RSF) laws embedded in an elastic half-space (K. Richards-Dinger & J.H. Dieterich 2012; B.E. Shaw *et al.* 2018). However, the possibility of forecasting individual events using such physics-based models has not been widely explored. This is the question we investigate in this study.

To forecast individual earthquakes, one would need models that are consistent with physical laws and that can be tuned to match historical data within their uncertainties, enabling forecasts of future events with quantified uncertainty. In principle, the frictional properties and state of a fault could be inferred from geodetic and seismological observations, allowing earthquake sequence models to be calibrated against real-world data (S. Barbot *et al.* 2012). However, inferring these model parameters—including fault geometry, frictional heterogeneity and tectonic loading—is an extremely challenging inverse problem. As a first step, we assume that we have access to a physical model capable of producing chaotic sequence of events. Under this assumption, we focus on the more tractable problem of forecasting the next large event in the sequence. This requires a data assimilation framework that incorporates noisy and sparse observations to update the model state and produce probabilistic forecasts.

We begin by considering a system that generates ‘slow’ earthquakes, also known as slow slip events (SSEs) (G. Rogers & H. Dragert 2003). SSEs are episodic slip events that resemble regular earthquakes (S. Michel *et al.* 2019), but are slower and more frequent, resulting in chaotic but potentially more predictable sequences (A. Gualandi *et al.* 2020). However, a major limitation of data such as time-series recorded at geodetic stations is that they do not provide direct information about the stress distribution on the fault—arguably the most critical quantity for forecasting. Moreover, these data are typically sparse and have a low signal-to-noise ratio, which poses a serious challenge: model trajectories that fit the observations within uncertainty can still diverge quickly due to the system's sensitivity to initial conditions.

In our previous study (H. Kaveh *et al.* 2025), we showed that large events can be forecasted due to the self-organization of the stress field resulting from prior ruptures. However, that study did not address the data assimilation problem. This is particularly challenging because the models are governed by high-dimensional, nonlinear partial differential equations (PDEs) (J.R. Rice 1993; N. Lapusta *et al.* 2000), and the available observations are sparse and noisy. For example, in slip inversions, we significantly lose spatial resolution when estimating fault slip from surface displacements, making it difficult to constrain the underlying stress distribution.

Here, we present a machine learning-based reduced-order model (ROM) of earthquake sequences designed to facilitate data assimilation. The ROM takes large-scale features as input and remains robust to the loss of small-scale information, while reproducing both slow/ fast chaotic behaviour. Unlike PDE models, which describe the evolution of full-field variables such as slip rate or the state variable in rate-and-state friction, the ROM captures the dynamics of the earthquake cycle using a low-dimensional vector representation. This dimensionality reduction enables efficient data assimilation by simplifying the integration of observational data. Moreover, since the ROM is machine-learned, it runs orders of magnitude faster than conventional PDE solvers, making it especially suitable for inverse problems and data assimilation tasks (C. Liu *et al.* 2022; R. Maulik *et al.* 2022; H. Mousavi *et al.* 2025).

Recent studies have applied physics-informed neural networks (PINNs) to fault slip monitoring and parameter estimation. For instance, R. Fukushima *et al.* (2023) introduced a PINN framework to simulate slow-slip events in a spring–slider system, simultaneously estimating rate-and-state friction parameters from synthetic observations and predicting future slip evolution. Their study demonstrated that PINNs can accurately reproduce the temporal dynamics of slow slip and recover frictional properties directly from data while incorporating the governing physical laws. C. Rucker & B.A. Erickson (2024) developed a multi-network PINN framework that models elastodynamic fault slip governed by rate-and-state friction, enabling simultaneous solution of forward and inverse problems and inference of depth-dependent friction parameters. The machine-learned ROM developed here is trained on data generated from the full physical model rather than constrained by it. The ROM learns the effective low-dimensional dynamics in a reduced basis, making it well suited for capturing the statistical behaviour of chaotic systems where direct enforcement of the governing equations is often intractable. While PINNs effectively leverage physical constraints to improve data efficiency and generalization, their training can be challenging because of ill-conditioned loss landscapes, a difficulty that is further pronounced in systems exhibiting chaotic dynamics (S. Wang *et al.* 2021; A.S. Krishnapriyan *et al.* 2021; S. Steger *et al.* 2022; P. Rathore *et al.* 2024). The data-driven machine-learned ROM therefore provides a complementary framework for modelling complex, multiscale and chaotic fault slip behaviour while remaining computationally efficient and robust over extended timescales.

Reduced-order modelling techniques have gained significant attention in science and engineering for their ability to efficiently approximate complex physical processes (T. Schneider *et al.* 2021; K. Fukami & K. Taira 2023; H. Mousavi & J.D. Eldredge 2025). They have also been applied in various geophysical contexts, such as modelling turbulent geophysical flows (O. San & R. Maulik 2018) and the thermal structure of subduction zones (G.M. Hobson & D.A. May 2025). In seismology, ROMs have been used for seismic waveform modelling (R. Hawkins *et al.* 2023; T. Nagata *et al.* 2023; J.M. Rekoske *et al.* 2025). The use of ROMs thus appears well-suited to approximating physics-based models of earthquake sequences and history-matching them to observations. However, there is no guarantee that predictability can be achieved with such models. In this study, we explore this question.

Data assimilation techniques, particularly the Ensemble Kalman Filter (EnKF), have been applied to models of earthquake sequences in several previous studies. For example, K. Hirahara & K. Nishikiori (2019) successfully used an EnKF approach to forecast SSEs on a fault governed by RSF laws using surface observations and to estimate model parameters. However, they adopted a simplified setup that produced purely periodic behaviour. Similarly, H.A. Diab-Montero *et al.* (2023) used a forward model with periodic behaviour and a simplified observational operator, both of which may not fully capture the complexities of real earthquake sequences. Building on these studies, our work considers a fault model that generates a complex, chaotic sequence of events. The synthetic data includes a diverse range of events with varying magnitudes and locations along the fault, making both temporal and spatial forecasting a significantly more challenging task.

In this paper, we develop a ROM of earthquake sequences and demonstrate its utility for data assimilation and forecasting. In Section 2, we describe the governing physical model and the procedure for constructing the ROM using Proper orthogonal decomposition (POD), followed by a machine learning framework to approximate its dynamics. We also present our data assimilation setup, including the formulation of the forward and observational models and the implementation of the EnKF. In Section 3.1, we evaluate the ROM's ability to reproduce the long-term statistical properties of the full model—including the magnitude–frequency, moment–duration and moment–area relationships—demonstrate the performance of the EnKF in recovering the system state from sparse and noisy observations and assess the accuracy of event forecasts in both time and space. Finally, in Section 4, we examine the assumptions underlying the approach, explore the limits of predictability, and discuss challenges in applying this framework to more realistic settings. The paper concludes in Section 5 with a summary of key findings and directions for future research.

2 METHODS

2.1 Physical model

The resistance of faults to sliding is described by the laboratory-derived rate-and-state friction law, which has been extensively applied to model earthquake sequences (J.H. Dieterich 1979; A. Ruina 1983; N. Lapusta & Y. Liu 2009). The shear stress on the fault surface, τ, is:

$$\tau = \bar{\sigma}\left(f^* + a\ln\left(\frac{v}{v^*}\right) + b\ln\left(\frac{v^*\theta}{d_{\mathrm{rs}}}\right)\right), \tag{1}$$

where $\tau : \Gamma \times \mathbb{R}^+ \mapsto \mathbb{R}$ is a function of location on the fault surface Γ, and time for all $t > 0$. The variables $v : \Gamma \times \mathbb{R}^+ \mapsto \mathbb{R}^+$ and $\theta : \Gamma \times \mathbb{R}^+ \mapsto \mathbb{R}^+$ denote the slip rate and state variable, respectively. The state variable θ encapsulates the memory of contact on the fault (J.H. Dieterich 1979; A. Ruina 1983). In this equation, $\bar{\sigma}$ represents the effective normal stress, while f^* is the friction coefficient at the reference slip rate v^*. The parameters a, b and d_{rs} are frictional properties, where d_{rs} denotes the characteristic slip distance. We assume that f^*, v^*, d_{rs} and $\bar{\sigma}$ are spatially uniform and temporally constant. However, the frictional parameters a and b are modelled as piecewise constant functions that vary spatially along the fault but remain constant in time, with $a : \Gamma \mapsto \mathbb{R}$ and

$b : \Gamma \mapsto \mathbb{R}$. The sign of $a - b$ determines the fault's frictional behaviour. For $a - b < 0$, the fault is velocity weakening (VW), where an increase in slip rate (v), combined with slip exceeding d_{rs}, reduces the fault strength, enabling earthquake nucleation and rupture acceleration. Conversely, when $a - b > 0$, the fault exhibits velocity strengthening (VS) behaviour, meaning an increase in slip rate enhances the fault strength. Such regions inhibit rupture nucleation and rupture propagation (J.H. Dieterich 1979).

The shear stress rate on the fault is approximated by:

$$\partial_t \tau = \mathcal{L}(v - v_{\mathrm{pl}}) - \kappa \partial_t v, \tag{2}$$

where $\kappa = \mu/2c_s$ represents the radiation damping coefficient, μ is the shear modulus and c_s is the shear wave speed. The term $\kappa \partial_t v$ accounts for energy radiated away as seismic waves, which becomes significant only at high slip rates, and the operator $\mathcal{L}$ is a linear pseudo-differential operator that captures elastostatic stress transfer due to slip (J.R. Rice 1993).

By differentiating eq. (1) with respect to time and using eq. (2), we can eliminate $\partial_t \tau$ and find an evolution law for slip rate (v) on the fault. To close the system, we also need an evolution law for the state variable θ. Various formulations have been proposed for θ's evolution (A. Ruina 1983; J.R. Rice & A.L. Ruina 1983). In this study, we adopt the aging law (A. Ruina 1983), leading to the following closed dynamical system for slip rate on the fault v, and state variable θ:

$$\partial_t v = \left[\kappa + \frac{a\bar{\sigma}}{v}\right]^{-1} \left[\mathcal{L}\big(v - v_{\mathrm{pl}}\big) - b\bar{\sigma}\left(\frac{1}{\theta} - \frac{v}{d_{\mathrm{rs}}}\right)\right], \quad (z, t) \in \Gamma \times (0, \infty), \tag{3a}$$

$$\partial_t \theta = 1 - \frac{v\theta}{d_{\mathrm{rs}}}, \quad (z, t) \in \Gamma \times (0, \infty). \tag{3b}$$

where z denotes the spatial coordinate(s) along the fault surface. For a 1-D fault embedded in a 2-D medium, $\Gamma = [0, L]$, where L is the length of the fault along strike. For a 2-D fault in a 3-D medium, $\Gamma = [0, L] \times [0, D]$, where L and D denote the fault length along strike and downdip width, respectively. In general, the fault surface Γ can be any 1-D or 2-D manifold embedded in the surrounding medium. For simplicity of the forward model, we focus on planar faults in this study. This simplification does not affect the reduced-order modelling or data assimilation framework, which are general and applicable to arbitrary fault geometries. Due to the nonlinearity, the fault slip rate can vary by many orders of magnitude in a very short time, leading to a complex multiscale behaviour. As a result, even in the case of a simple planar fault with a single VW patch, the dynamical system defined in eq. (3a, b) can produce multiscale, periodic or chaotic slip events in both time and space, depending on the model parameters and geometry (S. Barbot 2019). In this study, we focus on parameters that generate chaotic time series, as forecasting periodic slip events has already been addressed in previous studies. The main part of our analysis is conducted using a 2-D fault embedded within a 3-D medium, as illustrated in Fig. 1(a). The corresponding model parameters, detailed in Table 1, produce a complex sequence of SSEs that are spatially and temporally irregular, with magnitudes ranging from 6.1 to 7.3 over 500 yr of simulation, after a 100-yr spin-up period used to remove the influence of the initial condition (Fig. 1).

2.2 Proper orthogonal decomposition (POD)

POD is a linear model reduction technique used to extract dominant patterns from complex data sets, providing an efficient representation of a system's dynamics with reduced computational complexity. It has found wide application in various fields, such as fluid dynamics and geophysical modelling (K. Taira *et al.* 2017; J.M. Rekoske *et al.* 2025), including the modelling of fault slip (A.P. Kositsky & J.P. Avouac 2010; H. Kaveh *et al.* 2025). In the context of this study, POD is applied to data generated from eq. (3a, b). Given that the quantities of interest, v and θ, vary over many orders of magnitude, it is numerically more appropriate to perform model reduction on the logarithms of these variables, $\log_{10} v$ and $\log_{10} \theta$. At time t and spatial location $z \in \Gamma$, let $q(z, t)$ denote either $\log_{10} v(z, t)$ or $\log_{10} \theta(z, t)$. After spatial discretization, the field values at l grid points form the vector $q(t) \in \mathbb{R}^l$. In the POD framework, we aim to find an optimal set of basis functions ϕ_j^q to represent q in a space–time decomposition, as expressed by K. Taira *et al.* (2017):

$$q(z, t) - \bar{q}(z) = \sum_j \alpha^q(j, t)\phi_j^q(z), \quad j \in \mathbb{N} \tag{4}$$

where $\bar{q}(z)$ is the snapshot average taken at non-uniform time intervals of q, and $\phi_j^q(z)$ captures the spatial dependence of the data. Once these basis functions are determined, we can describe the time evolution of the system by computing the temporal coefficients $\alpha^q(j, t)$ for each mode at any time t. The superscript q emphasizes that we must compute these basis functions separately for both $\log_{10} v$ and $\log_{10} \theta$.

When simulating eq. (3a, b), we typically have snapshots of the field q (representing v and θ on a logarithmic scale) that are taken at non-uniform time intervals. This non-uniformity arises due to the fast and slow dynamics inherent in the system. We assume we have r snapshots, where r is sufficiently large, and each snapshot corresponds to a finite-dimensional data vector $q(t_i) \in \mathbb{R}^l$, with l being the number of spatial grid points and with $r > l$. For all field snapshots, we first remove the snapshot average, $\bar{q} = 1/r \sum_{i=1}^{r} q(t_i)$, from each data vector to centre the data. This results in defining a new vector, $w(t_i)^{(q)}$, for each snapshot i:

$$w(t_i)^{(q)} = q(t_i) - \bar{q} \in \mathbb{R}^l, \quad i = 1, 2, \cdots, r. \tag{5}$$

We then construct a matrix $W^{(q)}$:

$$W^{(q)} = \left[w^{(q)}(t_1)\ w^{(q)}(t_2)\ \cdots\ w^{(q)}(t_r)\right] \in \mathbb{R}^{l \times r}, \tag{6}$$

The optimal basis functions for eq. (4) correspond to the eigenvectors of the covariance matrix WW^T, ordered with respect to descending variance defined by the eigenvalue corresponding to a given eigenvector (K. Taira *et al.* 2017). These eigenvectors and eigenvalues are obtained through the singular value decomposition (SVD) of $W^{(q)}$:

$$W^{(q)} = \Phi^{(q)} \Sigma^{(q)} \Psi^{(q)T}, \tag{7}$$

where $\Phi^{(q)} \in \mathbb{R}^{l \times l}$ and $\Psi^{(q)} \in \mathbb{R}^{r \times r}$ are orthogonal matrices and $\Sigma^{(q)} \in \mathbb{R}^{l \times r}$ is a rectangular matrix with non-zero entries ($\sigma_j^{(q)}$) only on its leading diagonal. To reduce computational cost, we employ the reduced SVD, in which, due to the structure of $\Sigma^{(q)}$ (the last $r - l$ columns being zero), only the first l columns of $\Psi^{(q)}$ are computed. The j^{th} column of $\Phi^{(q)}$ represents the eigenvector corresponding to the j^{th} eigenvalue $\lambda_j^{(q)}$, which is computed as:

$$\lambda_j^{(q)} = \frac{1}{(r-1)} \sigma_j^{(q)^2}. \tag{8}$$

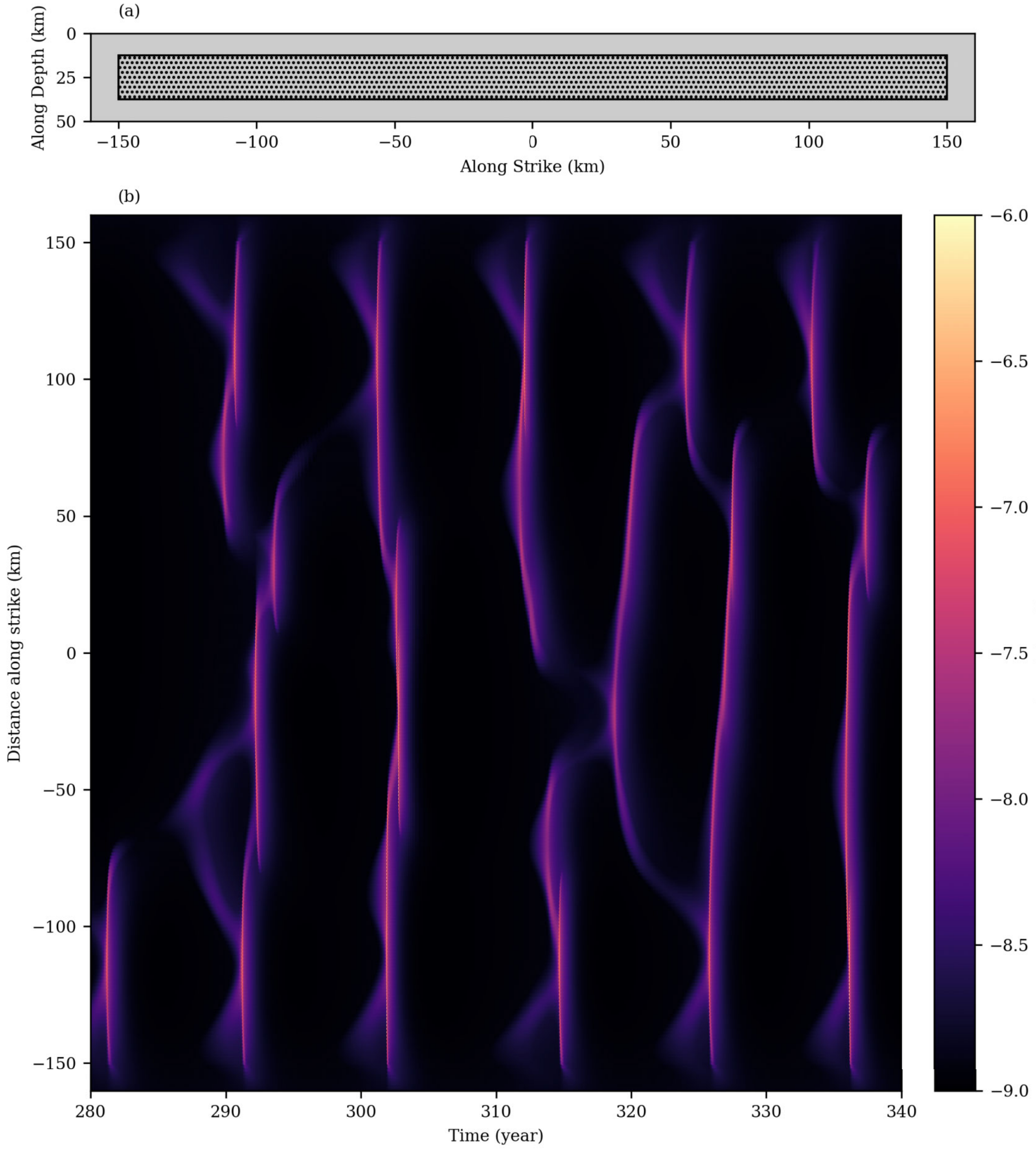

Figure 1. Geometry of the fault and chaotic behaviour of the dynamical model. (a) The geometry of the 2-D fault, with length L along the strike and D along the depth, showing the velocity-weakening (VW) patch (dotted area) embedded within a velocity-strengthening (VS) region. The lengths of the VW patch along the strike and depth are L_{VW} and D_{VW}, respectively. Physical properties are uniform everywhere except for the parameters a and b, which differ between the VW and VS regions (see Table. 1). (b) Maximum slip rate along the fault depth as a function of distance along the strike and time.

The eigenvectors are orthogonal $\langle\phi^q_{j'}, \phi^q_j\rangle = \delta_{j'j}$, where $\langle\cdot,\cdot\rangle$ denotes the inner product in $\mathbb{R}^l$, and $\delta_{j'j}$ is the Kronecker delta function. Given $q(t) \in \mathbb{R}^l$, the temporal coefficients $\alpha^q(j, t)$ can be computed using the orthogonality property of the basis functions ϕ^q_j as follows:

$$\alpha^q(j, t) = \langle q(t) - \bar{q}, \phi^q_j\rangle, \quad 1 \le j \le l. \tag{9}$$

We retain the first n_q eigenvectors, and approximate $q(z, t)$ as:

$$q(z, t) - \bar{q}(z) \approx \sum_{j=1}^{n_q} \alpha^q(j, t)\phi^q_j(z). \tag{10}$$

To determine n_q, we choose it such that the ratio of the sum of the first n_q eigenvalues to the sum of all eigenvalues exceeds a pre-defined threshold. Specifically, we select n_q such that:

$$\frac{\sum_{j=1}^{n_q} \lambda^q_j}{\sum_{j=1}^{l} \lambda^q_j} > 0.9, \tag{11}$$

This ensures that the chosen modes capture at least 90 per cent of the total variance in the data. The dimension n_q should also be chosen to be low enough to facilitate easy tuning of the machine learning model (described in the next section) while remaining sufficiently rich to capture important physical phenomena such as scaling laws. For simplicity of notation, we concatenate all the $\alpha^q(j, t)$ coefficients (for $1 \le j \le n_q$) into a single vector $\alpha^q(t) \in$

Table 1. Parameters used in the model of slow slip events (2-D fault)

Category	Property	Value	Units
Frictional properties	d_{rs}	0.045	m
	a_{VS}	0.019	–
	b_{VS}	0.014	–
	a_{VW}	0.004	–
	b_{WV}	0.014	–
Other physical properties	$\bar{\sigma}$	10	MPa
	μ	30	GPa
	c_s	3.3	km s^{-1}
	ν	0.25	–
Loading quantity	V_{pl}	40	mm yr^{-1}
Geometric quantities	L_{VW}	300	km
	D_{WV}	25	km
	L	320	km
	D	50	km
	Dip angle	17.5	°

$\mathbb{R}^{n_q}$, defined as:

$$\alpha^q(t) = \begin{bmatrix} \alpha^q(1,t) \\ \alpha^q(2,t) \\ \vdots \\ \alpha^q(n_q,t) \end{bmatrix}. \tag{12}$$

The introduced model reduction enables us to represent both $\log_{10} v$ and $\log_{10} \theta$ in low-dimensional spaces, $\mathbb{R}^{n_v}$ and $\mathbb{R}^{n_\theta}$, respectively. Here, n_v represents the number of components retained for the slip rate v, and n_θ denotes the number of components retained for the state variable θ. We set $n_v = n_\theta$, and denote the total number of components as $n = n_v + n_\theta$. For convenience, we use α^v and α^θ to represent the temporal components of the logarithm of the slip rate and the logarithm of the state variable, respectively. In the following section, we employ machine learning to derive a time evolution law for α^v and α^θ, such that, starting from an initial condition, we can simulate a sequence of events without directly solving eq. (3a, b).

2.3 Learning slow/fast dynamics for reduced-order models

In reduced-order modelling using machine learning, we aim to identify an evolution law for $\alpha = (\alpha^v \in \mathbb{R}^{n_v}, \alpha^\theta \in \mathbb{R}^{n_\theta}) \in \mathbb{R}^n$, represented as:

$$\dot{\alpha} = g(\alpha), \quad t \in (0, \infty), \tag{13}$$

where n_v and n_θ are the numbers of retained POD modes for v and θ, respectively, and $g : \mathbb{R}^n \mapsto \mathbb{R}^n$ with $n = n_v + n_\theta$. The dynamical system in eq. (13) is obtained through machine learning and can be integrated numerically at a much lower computational cost in comparison with integration of the full PDE model (eq. 3a, b). Once the function g is learned, using an initial condition, one does not need to use eq. (3a, b) to simulate the sequence of earthquakes.

However, learning the dynamics of eq. (13) presents significant challenges due to the multiscale (slow-fast) and chaotic nature of the underlying system. Typically, eq. (3a, b) is integrated using an adaptive time stepping scheme, where time steps vary from a few seconds to several hours. This variability complicates the learning process for g, as its behaviour reflects the system's dynamics: g outputs small values during slow dynamics and large values during fast dynamics. To overcome these challenges, we propose a methodology tailored for learning chaotic slow-fast dynamical systems. Our approach involves a transformation of the time variable to eliminate the slow/fast behaviour. Instead of using the physical time variable t, we introduce a transformed time variable s, in which the system evolves uniformly:

$$\frac{\mathrm{d}\alpha}{\mathrm{d}t} = \frac{\mathrm{d}\alpha}{\mathrm{d}s}\frac{\mathrm{d}s}{\mathrm{d}t}. \tag{14}$$

While directly learning $\mathrm{d}\alpha/\mathrm{d}t$ is challenging, we decompose this task into learning two separate functions: $g_1 = \mathrm{d}\alpha/\mathrm{d}s$ and $1/g_2 = \mathrm{d}s/\mathrm{d}t$. Using this decomposition, $\mathrm{d}\alpha/\mathrm{d}t$ can be reconstructed as g_1/g_2. In a discrete-time formulation, approximating $\dot{\alpha}$ using forward finite differences yields:

$$\dot{\alpha}(t_i) \approx \frac{\Delta\alpha}{\Delta t} \approx \frac{\alpha(t_i + \Delta t_i) - \alpha(t_i)}{\Delta t_i} = \underbrace{\frac{\alpha(t_i + \Delta t_i) - \alpha(t_i)}{1}}_{g_1} \cdot \underbrace{\frac{1}{\Delta t_i}}_{\frac{1}{g_2}}. \tag{15}$$

Here, t_i represents the i^{th} time step, obtained using an adaptive time stepping scheme with Δt_i as the adaptive physical time step increment, which varies across different timescales. The data set is constructed from numerical solutions of the underlying PDE, producing snapshots of the field at non-uniform time intervals. Projecting these snapshots onto the POD basis generates a non-uniform time-series for $\alpha(t_i)$. Using the formulation in eq. (15), we train separate neural networks to learn g_1 and g_2.

To improve the accuracy of the machine-learned model and help the neural network g_2 better capture the function's variability, we explicitly include $\log_{10} ||v(t_i)||_\infty$ (logarithm of maximum slip rate at t_i) as an additional input. Note that $||v(t_i)||_\infty$ can be directly approximated using $\alpha_v(t_i)$. In addition, we empirically observe that information only from α_v is enough to predict the time step. As a result, we exclude α_θ as input for g_2. This exclusion is advantageous because, as shown later in the data assimilation problem, components of α_θ are estimated with lower accuracy compared to α_v. As a result, the neural networks are defined as $g_1 : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $g_2 : \mathbb{R}^{n_v+1} \mapsto \mathbb{R}^+$. For more details on the structure of these neural networks, how we generate the training data and how we impose dissipative behaviour of g, see Appendix A and B.

Due to the chaotic nature of the system, the machine-learned model (eq. 13) cannot be used for long-term trajectory prediction. Here, long-term trajectory prediction refers to integrating the model beyond the Lyapunov time, where small errors in initial conditions or model representation grow exponentially and lead to divergence from the reference solution. This limitation arises from two factors: (1) projection of the initial condition and state space onto the first few modes introduces errors, and (2) inaccuracies in the machine-learned model are inevitable. These factors, combined with the chaotic dynamics, lead to divergence between the long-term trajectories of the original system and the ROM. Nevertheless, the machine-learned model captures the long-term statistical properties of the system, such as earthquake scaling laws—including the Gutenberg–Richter magnitude–frequency, moment–area and moment–duration relationships—with behaviour similar to the original rate-and-state model. For short-term dynamics, the trajectories of the ROM are designed to remain close to those of the original PDE, making the ROM suitable for sequential data assimilation problems. This is achieved by training the neural network with a loss function that minimizes the mean-squared error of one-step-ahead predictions. The low dimensionality and computational efficiency of the machine-learned model, combined with its ability to leverage large-scale patterns in data while remaining robust to small-scale smoothing, make it particularly advantageous in data assimilation settings, especially when observations are sparse and have lost fine-scale information. In the next section, we describe the data assimilation framework and the forward and observational models used in this paper.

2.4 Data assimilation

Data assimilation is a mathematical framework to combine observational data with numerical models to estimate the state of a dynamic system and improve predictions; typically, this is done sequentially in time, as data is acquired and this is the version of data assimilation we deploy in this paper. Our approach integrates a physics-based, machine-learned model as the backbone of the forecast, distinguishing it from purely data-driven methods developed for laboratory earthquake prediction that rely on empirical relationships between statistical attributes of seismic or acoustic data and the fault state (B. Rouet-Leduc *et al.* 2017, 2019; C. Hulbert *et al.* 2019; J. Lubbers *et al.* 2023). In contrast, the present framework learns a reduced dynamical model constrained by rate-and-state friction laws and assimilates observations to correct for unrepresented effects such as transient phenomena, chaotic dynamics and observational noise.

We assume that observations occur at uniform time intervals, denoted by Δt_{obs}. Our goal is to estimate the state update (α_k) at the kth observation time increment using noisy slip rate observations on the fault, which are further corrupted by a low-pass filter. The low-pass filter mimics the limited resolution provided by surface geodetic measurements. This estimated state is then utilized to forecast large events effectively. For convenience, we use the subscript notation to indicate the observation time increment, that is, $\alpha_k = \alpha(t = k\Delta t_{\text{obs}})$. In general, α_k can represent boundary conditions or uncertain model parameters, but throughout this paper, $\alpha_k \in \mathbb{R}^n$ specifically denotes the temporal POD coefficients of the ROM at $t = k\Delta t_{\text{obs}}$.

To provide a comprehensive understanding of the data assimilation framework, it is essential first to define the forward model and the observation model used in our approach. The forward model represents the physical processes governing the earthquake cycle in the reduced space, while the observation model relates the states of the forward model to observable quantities. In the following sections, we explain these models in detail before outlining the data assimilation formulation.

2.4.1 *Forward and observation model*

In data assimilation, the forward model represents the mathematical or physical system used to predict the evolution of the system's state, while the observation model relates the system's state to measurable quantities by simulating the process of obtaining observations. In eq. (13), we derived an ODE describing the evolution of α. However, for the purpose of data assimilation, since observations are available at discrete time intervals of Δt_{obs}, it is sufficient to construct a solution operator $\psi : \mathbb{R}^n \to \mathbb{R}^n$. This operator takes the state α_k (the value of α at $t = k\Delta t_{\text{obs}}$) as input, solves eq. (13) and outputs α_{k+1}, the state at $t = (k+1)\Delta t_{\text{obs}}$. We assume that the model contains errors arising from various sources, including inaccuracies in the neural network (eq. 13) and truncation of the POD modes (eq. 10). In practice, inaccuracies in the physical model—for example, inaccuracies in eq. (3a, b)—when representing real data can be accounted for as part of the model noise. Additionally, we assume the initial condition is randomly distributed according to a Gaussian distribution with mean zero and covariance matrix C_0. We obtain the following stochastic evolution for the states α_k :

$$\alpha_{k+1} = \psi(\alpha_k) + \xi_k, \quad k \in \mathbb{Z}^+, \tag{16a}$$

$$\alpha_0 \sim \mathcal{N}(0, C_0), \tag{16b}$$

where $\xi = \{\xi_k\}_{k\in\mathbb{N}}$ is an independent and identically distributed (i.i.d.) sequence with $\xi_k \sim \mathcal{N}(0, \Xi)$, where Ξ is empirically estimated by comparing the solutions of eq. (13) and eq. (3a, b) at discrete time steps Δt_{obs}. We use $\mathbb{Z}^+$ to denote the set of non-negative integers, including zero. The covariance matrices from the SVD (eq. 7), specifically Σ^v and Σ^θ, are used to define C_0. Eq. (16a) thus defines our forward model. Due to model inaccuracies and the chaotic nature of the system, eq. (16a) loses information after a finite simulation time. Even in the deterministic case where α_0 is known exactly and $\xi_k = 0$, predictive accuracy deteriorates after a few iterations. This is due to the amplification of numerical errors by the system's sensitivity to initial conditions—an intrinsic feature of chaotic dynamics—although the long-term statistics remain stationary and informative of the original PDE (see Section 2.3). This loss of predictive relevance is governed by the maximum Lyapunov exponent of the system, which measures the rate of divergence of nearby trajectories. These limitations can be mitigated using data assimilation, particularly through the sequential updating of the system state estimates as new observational data becomes available (D. Sanz-Alonso & A.M. Stuart 2015; K. Law *et al.* 2015; D. Sanz-Alonso *et al.* 2023).

We now seek to formulate our observational operator that maps the state vector $\alpha \in \mathbb{R}^n$ to an observable quantity $y \in \mathbb{R}^d$. In realistic settings, observations are obtained at the surface through geodetic measurements, and the corresponding observation operator would involve elastic Green's functions that relate fault slip to surface displacement (K. Hirahara & K. Nishikiori 2019). For simplicity, in this study, we assume that the observational data correspond directly to the slip rate on the fault. To partially account for the lower resolution and spatial smoothing inherent in surface observations, we apply a spatial blurring operator to the slip-rate field before assimilation, providing a more realistic observation scenario.

Since the state variable θ and its POD coefficients are not directly measurable, y consists only of information about the POD coefficients of the slip rate α^v, so we assume $d = n_v$. We formulate the observation model as follows:

$$y_{k+1} = h(\alpha_{k+1}) + \eta_{k+1}, \quad k \in \mathbb{Z}^+, \tag{17}$$

where $h : \mathbb{R}^n \to \mathbb{R}^{n_v}$. The noise sequence $\eta = \{\eta_k\}_{k\in\mathbb{N}}$ is i.i.d., independent of α_0 and ξ and satisfies $\eta_k \sim \mathcal{N}(0, \Pi)$ for $k \in \mathbb{N}$, with Π being a positive definite diagonal matrix. The observation of the slip rate on the fault is primarily contaminated by a low-pass filter. This low-pass filter eliminates high-frequency variations in the slip rate, resulting in a smoothed slip rate. We incorporate the effects of the low-pass filter into the observational operator h, while other sources of noise are represented by η. The observation model h is defined by modelling the low-pass filter using a Gaussian kernel applied to the slip rate measurements. The Gaussian kernel is given by:

$$G(z) = \frac{1}{2\pi\sigma_{\text{kernel}}^2} \exp\left(-\frac{\|z\|^2}{2\sigma_{\text{kernel}}^2}\right), \tag{18}$$

where σ_{kernel} is the standard deviation of the kernel, and $z \in \Gamma$. The low-pass filtered slip rate is obtained using the following transformation:

$$v'_{k+1}(z) = \int_\Gamma v_{k+1}(z') \cdot G(z - z')\, \mathrm{d}z', \tag{19}$$

where v_{k+1} is the slip rate at the $(k+1)$-th observation increment. The slip rate v_{k+1} can be expressed using the POD expansion as:

$$\log_{10}(v_{k+1}) \approx \overline{\log_{10}(v)} + \sum_{j=1}^{n_v} \alpha^v(j, t = (k+1)\Delta t_{\mathrm{obs}})\phi_j^v. \tag{20}$$

Finally, the filtered slip rate v'_{k+1} is projected back onto the POD modes of the slip rate (ϕ_j^v). The j-th element of $h(\alpha_{k+1}) \in \mathbb{R}^{n_v}$ is computed as:

$$h_j(\alpha_{k+1}) = \langle \log_{10} v'_{k+1} - \bar{v}, \phi_j^v \rangle, \quad j = 1, \cdots, n_v. \tag{21}$$

The nonlinear observational operator defined by eqs (18) to (21) converges to a linear operator as $\sigma_{\mathrm{kernel}} \to 0$. In the limiting linear case, the observation operator becomes a matrix that maps the state space to the observation space, given by $\bar{H} = \left[I_{n_v}, 0_{n_\theta}\right] \in \mathbb{R}^{n_v \times n}$, where I_{n_v} is an $n_v \times n_v$ identity matrix and 0_{n_θ} is an $n_v \times n_\theta$ zero matrix. We set the kernel width $\sigma_{\mathrm{kernel}} = 2\,\mathrm{km}$ in this work, consistent with the smoothing scale applied during synthetic observation generation. In real-world applications, the appropriate value of σ_{kernel} depends on factors such as the spatial density of surface observations, the level of regularization used in slip inversion and the signal-to-noise ratio in the data. In future work, σ_{kernel} could be treated as a tunable parameter within the data assimilation framework—either by augmenting the state vector or through hierarchical Bayesian modelling. While we do not pursue this direction here, such approaches may lead to more adaptive and realistic observation models.

2.4.2 Ensemble Kalman filter

In data assimilation, filtering consists of two main steps: the forecast step, where the system's state is predicted at the next observation time using the forward model, and the analysis step, where this prediction is corrected using newly available observational data to refine the state estimate. During the forecast step, the state and its associated uncertainty are predicted based on the system dynamics, resulting in the prior distribution, which represents the state estimate before incorporating new observations. The analysis step updates this prior distribution with new measurements to produce the posterior distribution, reflecting the refined state estimate that incorporates the latest observation data.

Kalman-based filters attempt to optimally combine these steps at each time step to achieve the best possible state estimation (K. Law *et al.* 2015). For a linear forward and observation model, the Kalman filter provides an exact formulation of the posterior distribution of the system state; the resulting model mean is also the minimum variance estimator of the state. For nonlinear dynamical systems or observation models, the assumptions of linearity and Gaussianity are not applicable, necessitating alternative approaches. EnKF addresses these challenges by approximating the nonlinear state evolution and observation functions. EnKF employs an ensemble of state vectors to represent the system's distribution and approximates covariance updates using sample statistics derived from the ensemble, rather than computing them exactly. In the forecast step, the ensemble members are propagated through the nonlinear model to generate the forecast ensemble, denoted by the superscript f. In the analysis step, the ensemble members are updated using observed data, with an approximation of the Kalman gain derived from the ensemble covariance, resulting in the updated ensemble, denoted by the superscript a.

We begin the mathematical formulation of the EnKF with the forecast step. In this step, each ensemble member i is propagated forward in time using the forward model, which updates the state based on the analysis values from the previous time step. This process generates an approximate prior distribution for the state at the next time step. Mathematically, this step is expressed as:

$$\alpha_{k+1}^{f,i} = \psi\left(\alpha_k^{a,i}\right) + \xi_k^i, \quad i = 1, \cdots, m, \tag{22}$$

where $\alpha_{k+1}^{f,i}$ represents the forecasted state of the i-th ensemble member at time increment $k+1$, $\alpha_k^{a,i}$ is the analysis state from the previous time step, $\xi_k^i \sim \mathcal{N}(0, \Xi)$ are i.i.d. model errors associated with the i-th ensemble member and m is the total number of ensemble members.

Next, we move to the analysis step, which aims to refine the forecasted states by incorporating new observational data. This step updates each ensemble member to approximate the posterior distribution. The update process is expressed as:

$$\alpha_{k+1}^{a,i} = \alpha_{k+1}^{f,i} + K_{k+1}\left(y^* - h\left(\alpha_{k+1}^{f,i}\right)\right), \tag{23}$$

where $\alpha_{k+1}^{a,i}$ is the updated analysis state, y^* represents the noisy observed data, h is the observation model, $K_{k+1} \in \mathbb{R}^{n \times n_v}$ is the Kalman gain matrix (to be discussed shortly). In summary, the forecast step uses the analysis values from the previous time step to predict the next state, while the analysis step refines these predictions using new observations, improving the state estimates. To simplify notation, we omit the time subscript (k) from the variables in the rest of this section, as the analysis step does not involve time evolution.

The Kalman gain K plays a crucial role in balancing the influence of new observations against the forecasted state. It determines how much weight to assign to the measurements relative to the predictions, based on the reliability of the observations. Before detailing the computation of K for both linear and nonlinear observation models, we introduce some notations. For a set of m ensemble members ($\alpha^i \in \mathbb{R}^n, i = 1, \cdots, m$), the forecast anomaly matrix $A'^f \in \mathbb{R}^{n \times m}$ is defined as:

$$A'^f = \frac{1}{\sqrt{m-1}}\left[\alpha^{f,1} - \bar{\alpha}^f, \alpha^{f,2} - \bar{\alpha}^f, \cdots, \alpha^{f,m} - \bar{\alpha}^f\right], \tag{24}$$

where $\bar{\alpha}^f = 1/m \sum_{i=1}^m \alpha^{f,i}$. We also define the innovation anomaly matrix $Y'^f \in \mathbb{R}^{n_v \times m}$ with its i^{th} column:

$$Y'^{f,i} = \frac{h(\alpha^{f,i}) - \bar{y}^f - \eta^i + \bar{\eta}}{\sqrt{m-1}}, \quad i = 1, \cdots, m, \tag{25}$$

where $\bar{y}^f = 1/m \sum_{i=1}^m h(\alpha^{f,i})$, and $\bar{\eta}$ is the sample mean of $\eta^i (i = 1, \cdots, m)$ which are drawn i.i.d from observation noise.

For when the observation model is linear, the Kalman gain ($K \in \mathbb{R}^{n \times n_v}$) is identical to its form in the standard Kalman filter and is given by:

$$K = P^f H^\top \left(H P^f H^\top + \Pi\right)^{-1}. \tag{26}$$

The prior covariance matrix P^f represents the uncertainty in the forecasted state. When the model f is nonlinear (as in our study), P^f is approximated by the sample prior covariance matrix, given by $P^f \approx A'^f A'^{f\top}$. For cases where the observation model is nonlinear (which also applies to our study), the tangent linear approximation of the observation operator is utilized (G. Evensen 1994; M. Le Provost & J.D. Eldredge 2021):

$$H(\alpha^{f,i} - \bar{\alpha}^f) \approx h(\alpha^{f,i}) - \bar{\alpha}^f. \tag{27}$$

Then, the Kalman gain is calculated by:

$$K = A'^f Y'^{f\top}(Y'^f Y'^{f\top})^{-1}. \tag{28}$$

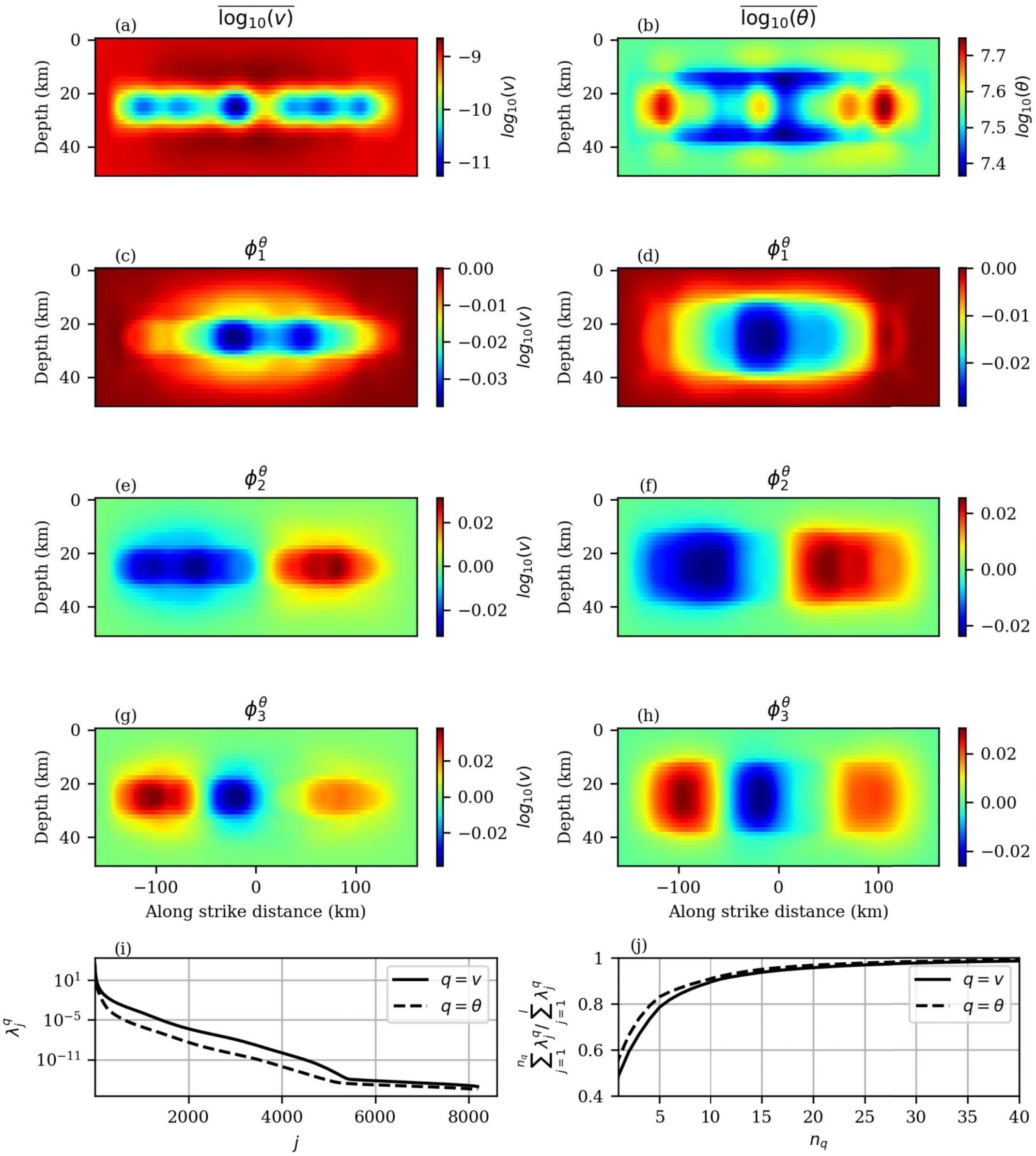

Figure 2. POD analysis of the system's fields (v and θ) and mode variance. (a-b) Snapshot averages of slip rate ($\overline{\log_{10}(v)}$) and state variable ($\overline{\log_{10}(\theta)}$). (c-h) The first three eigenmodes of the slip rate and state variable. (i) Variance of each mode (j) in the singular value decomposition. (j) Ratio of the sum of the first n_q eigenvalues to the sum of all eigenvalues (defined in eq. 11).

3 RESULTS

3.1 Reduced-order model components and scaling properties

In this section, we present the components of the POD and the criteria for selecting the number of modes. We also evaluate the performance of the ROM by comparing its scaling properties—specifically the magnitude–frequency, moment–duration and moment–area relationships—to those of the original PDE. The snapshot averages of the logarithm of the slip rate and state variable, denoted by $\overline{\log_{10}(v)}$ and $\overline{\log_{10}(\theta)}$, are shown in Figs 2(a) and (b). Figs 2(c), (e), (g) display the first three eigenmodes for the slip rate, while Figs 2(d), (f), (h) show the first three eigenmodes for the state variable. As the number of modes increases, the eigenmodes capture progressively finer spatial details.

This observation highlights the importance of model reduction: the first few modes capture the dominant large-scale spatial features while filtering out high-frequency spatial variations. Since observational data typically lack high-frequency resolution, truncating higher-order modes ensures that the ROM remains compatible with realistic observational data while maintaining robustness to inaccuracy in spatially high-frequency data.

Fig. 2(i) illustrates the variance associated with each eigenmode for both the slip rate and state variable. Fig. 2(j) shows the cumulative ratio of the sum of the first n_q eigenvalues to the total sum of all eigenvalues, as defined in eq. (11). We select 20 modes for both v and θ, as this number of modes captures more than 90% of the total variance for each variable. As we will show here, this ROM configuration effectively reproduces the scaling behaviour observed in the original PDE, providing a balance between accuracy and computational efficiency.

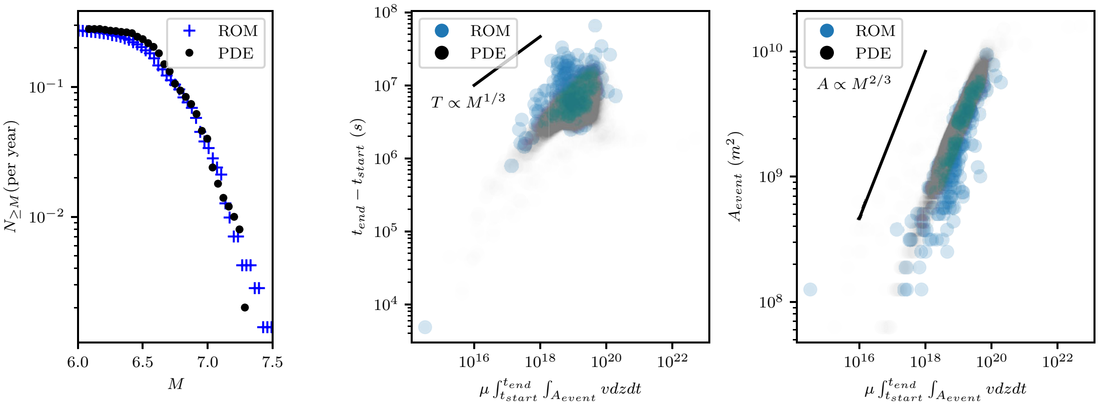

Figure 3. Comparison of scaling properties between the ROM and the original PDE. The left panel shows the number of events exceeding a given magnitude as a function of magnitude. The middle and right panels compare the moment–duration and moment–area scaling laws, respectively, for the original PDE (black) and ROM with $n = 40$.

An event is defined based on the maximum slip rate on the fault, $||v||_\infty$, exceeding a specified threshold v_{thresh}. An event is considered to have started when $||v||_\infty > v_{\text{thresh}}$ and is considered to have ended when $||v||_\infty$ falls below v_{thresh}. Events that occur in spatially distinct regions are treated as separate events, even if they overlap in time. Specifically, if two events are separated by more than l_{thresh} along the strike direction, they are counted as distinct events. The seismic moment of an event is defined as:

$$M = \frac{2}{3}\log_{10}\left(\mu \int_{t_{\text{start}}}^{t_{\text{end}}} \int_{A_{\text{event}}} v(z, t')\, \mathrm{d}z\, \mathrm{d}t'\right) - 6,$$

where t_{start} and t_{end} are the start and end times of the event, determined using $||v||_\infty$ and v_{thresh}, A_{event} is the ruptured area of the event, defined as the region where the slip rate exceeds v_{thresh} between t_{start} and t_{end}, and μ is the shear modulus. If disjoint ruptures are separated by more than l_{thresh}, they are treated as distinct events. For this study, we use a slip-rate threshold of $v_{\text{thresh}} = 5 \times 10^{-8}\,\mathrm{m\,s^{-1}}$, approximately forty times the imposed plate loading rate ($v_{\text{pl}} = 40\,\mathrm{mm\,yr^{-1}} \approx 1.3 \times 10^{-9}\,\mathrm{m\,s^{-1}}$), and a spatial threshold of $l_{\text{thresh}} = 1\,\mathrm{km}$.

Given the system's chaotic behaviour, even small inaccuracies in the ROM can result in significant deviations in the time series compared to the ground truth. However, it is important that the ROM preserves the statistical features of the original system. The blue markers in Fig. 3 illustrate the magnitude-frequency distribution, the moment-duration, and the moment-area scaling relationships for events generated by the long-term evolution of eq. (13). These are compared with events obtained from simulations of the original PDE (eq. 3a, b). The ROM demonstrates a strong capability to replicate the overall statistical properties and scaling behaviours, albeit with slight bias and considerably more variance, when compared to the original PDE. These differences highlight the limitations of the ROM in fully capturing the underlying dynamics; the differences may be attributed to the reduced dimensionality or approximations inherent in the ROM construction.

Recent studies have shown that the inferred scaling exponents for SSEs can depend on the detection level used to define an event (G. Costantino *et al.* 2025). Because the moment–duration and moment–area scaling laws are generally more sensitive to noise than the magnitude–frequency distribution, this sensitivity may explain the relatively larger variability observed in Figs 3(b) and (c).

3.2 Computational cost and efficiency of the machine-learned ROM

One of the main motivations for developing the machine-learned reduced-order model (ROM) is to reduce the computational cost of forward simulations and enable efficient data assimilation. Here, we quantify all major costs associated with (a) generating training data, (b) learning the ROM and (c) performing forward simulations, and we compare them to the original physics-based PDE solver (QDYN).

Full-order model cost. With the physical and geometric setup in Table 1 and Fig. 1, the computational domain is discretized into 32 × 256 grid points along depth and strike, respectively. Simulating 1000 yr of fault evolution using QDYN requires approximately 5,073 s ($\approx$ 1.41 hr) of wall-clock time on a single node with eight OpenMP threads. This corresponds to $\approx$ 11.3 CPU-hr per 1000-yr simulation or 0.0113 CPU-hr per simulated year.

Offline costs (one-time). To construct the ROM basis, we simulate a 600-yr trajectory (discarding the first 100 yr to remove transients), which costs $600 \times 0.0113 = 6.78$ CPU-hours ($\approx$ 0.85 hr of wall-clock time on eight threads). Computing the POD/SVD of the snapshot matrix requires about 0.07 CPU-hr. To generate training data, we perform 100 simulations of 250 yr each, totaling 25 000 yr of synthetic data, which costs $25\,000 \times 0.0113 = 282.5$ CPU-hr ($\approx$ 35.3 hr of wall-clock time on eight threads). Training the ROM on a single GPU takes about 6.07 GPU-hr. All these offline costs are incurred only once.

Online efficiency. After training, the ROM advances the fault state for 1000 yr in approximately 15 s when run on a single GPU, whereas the full-order QDYN model requires 5073 s (11.27 CPU-hr on eight threads) for the same duration. In terms of wall-clock time, the ROM is about 338 times faster, and when measured by total computational cost, this corresponds to an effective reduction of roughly 2700 times compared with the CPU-based PDE solver.

3.3 EnKF

In this subsection, we present the results of estimating the temporal components of the reduced model using our data assimilation framework. This framework employs the ROM as the forward model (eq. 16a) and assumes observational data are available at time intervals of $\Delta t_{\text{obs}} = 5$ d. The observation model is described by eqs (18) to (21), which apply a smoothing kernel and additive noise to a true signal generated by the full PDE model (eq. 3a, b). The true observation data are generated by simulating the original PDE (eq. 3a, b) from a random initial condition, discarding the first few years of data to remove transient behaviour. The data set used for data assimilation is not used in the training of the ROM. To generate the realistic synthetic data, the slip rate snapshots from the PDE simulation are interpolated to produce measurements at increments of Δt_{obs} days. Additionally, a Gaussian low-pass filter (eq. 19) with $\sigma_{\mathrm{kernel}} = 2$ km is applied to the interpolated data to mimic the spatial blurring inherent in slip inversions.

For the observation noise, we assume a diagonal covariance, with each diagonal entry equal to 5×10^{-4} of the variance of the POD modes for the corresponding component. The model error covariance matrix, Ξ, is empirically estimated by comparing long-term simulations of the full model with the ROM. The EnKF implementation employs 80 ensemble members, ensuring a robust statistical representation of the model's uncertainty. This ensemble size was chosen based on empirical tests, which showed improved performance over smaller ensembles while remaining computationally feasible. This configuration is used to assess the ROM's ability to assimilate noisy, spatially smoothed slip rate data and refine its predictions accordingly.

Fig. 4 shows the time-series of the temporal components of the POD in a ROM with $n = 40$. The ensemble members' slip rates closely track the true slip rate, exhibiting small uncertainties. Despite challenges introduced by the low-pass filtering, the EnKF algorithm effectively reconstructs the true slip rate (magenta) from the observation data (green). However, the accuracy for components associated with the state variable θ is significantly lower, with higher uncertainty. This is an expected limitation, as the components of θ are not directly observed, constraining the filter's ability to estimate this variable accurately. Inaccuracies in estimating the leading components of the state variable contribute to inaccuracies in event forecasts and uncertainty quantification, as we will see in the next section.

3.4 Event prediction

The estimates of the coefficients α^v and α^θ can be used to forecast future values of the slip rate v and the state variable θ. Due to the inherent chaotic nature of the system, long-term predictions diverge from the true trajectory. However, short-term predictions remain viable and meaningful within specific horizons. For this study, we use the estimate of the states of the system $t_{\mathrm{est}} = 0.1$ yr before the events and predict it up to $t_{\mathrm{pred}} = 0.4$ yr.

In Fig. 5, we plot the maximum slip rate along the depth of the fault as a function of time and along the strike distance. Each row in the figure corresponds to a distinct event, with the first column plotting the true signal. Time is shifted in this figure to be zero when an event starts in the true signal. At t_{pred} before each big event ($M > 6.9$), we use the mean of the ensemble members as the estimate of α^v and α^θ values to reconstruct the initial conditions of the governing model (eq. 3a, b). These reconstructed initial conditions serve as the initial conditions for forecasting the system's evolution. The second column shows the prediction derived from the estimated initial condition using a model with $n = 40$.

Each ensemble member provides a Monte Carlo approximation of the evolving distribution of the system states and can thus be used to quantify forecast uncertainty. The state estimate for each ensemble member is expressed as:

$$q(z,t) - \bar{q}(z) = \sum_{j\geq 1} \alpha^q(j,t)\phi_j^q(z) = \sum_{j=1}^{n_q} \alpha^q(j,t)\phi_j^q(z) + \sum_{j\geq n_q+1} \alpha^q(j,t)\phi_j^q(z), \tag{29}$$

where q denotes either $\log_{10} v$ or $\log_{10}\theta$. For each ensemble member, we replace $\alpha^q(j,t)$ for $1 \leq j \leq n_q$ with its corresponding estimate obtained from data assimilation. For the higher frequency modes ($j \geq n_q + 1$), we sample $\alpha^q(j,t)$ from a normal distribution with zero mean and variance defined in eq. (8).

To evaluate uncertainty, each ensemble member's estimate is computed using eq. (29) at time t_{est}, prior to the onset of the events. We then propagate the model forward up to t_{pred} after the event started. The spatial and temporal predictions associated with each ensemble member are shown in the third and fourth columns of Fig. 5 in blue. In the third column, the vertical axis represents the distance along strike, while the horizontal axis shows $\int_{-t_{\mathrm{est}}}^{t_{\mathrm{pred}}}\int_0^D v(x,y,t)\,\mathrm{d}y\,\mathrm{d}t$, where D is the fault depth. The fourth column illustrates the temporal uncertainty in predicting events, with time on the horizontal axis and $(\int_0^D\int_0^L v(x,y,t)\,\mathrm{d}x\,\mathrm{d}y)$ on the vertical axis.

We further quantify the performance of predictions in time and space using only the average of the ensemble members, but for more events in the data set. We test our method on 10 simulations, each starting from a random initial condition. After removing transient data from these simulations, we perform data assimilation on 55 yr of data. The total data set contains 24 events with magnitudes greater than 6.9. We use the average of the 80 ensemble members as the expected value of α. To evaluate the prediction performance in both time and space, we define four metrics. **True Positive Ratio (TPR)** and **False Positive Ratio (FPR)** quantify temporal prediction accuracy, while **True Positive Extent Ratio (TPER)** and **False Positive Extent Ratio (FPER)** measure spatial accuracy.

The TPR is defined as the ratio of correctly predicted events to the total number of events, where a correctly predicted event, for the purpose of calculating TPR, refers to a case in which the timing of an event is successfully forecast, regardless of its exact spatial location. The FPR quantifies the probability of predicting an event within a time period that does not contain any actual events.

To compute TPR, we estimate the system states at $t_{\mathrm{est}} = 0.1$ yr, a pre-defined time before an event occurs at t_{event}. The system is then simulated up to $(t_{\mathrm{event}} - t_{\mathrm{est}}) + t_{\mathrm{pred}}$, where $t_{\mathrm{pred}} = 0.4$ yr. These values of t_{pred} and t_{est} were chosen to maximize TPR while keeping FPR as low as possible, balancing prediction accuracy and reliability. In our data set, 24 events exceed a magnitude of 6.9, and the algorithm successfully predicts 18 of them, yielding a TPR of 0.75. To further assess the accuracy of these predictions, we introduce the prediction lag, defined as the time difference between the predicted start time of an event and its actual start time in the data set. The histogram of prediction lags, shown in Fig. 6(a), indicates that more than 75 per cent of correctly predicted events have a time lag between −0.1 and 0.1 yr, demonstrating the model's ability to forecast events with temporal precision.

To compute the FPR in our simulation, we randomly sample N^{FPR} instances of t^*, ensuring that no event occurs in the interval $[t^*, t^* + t_{\mathrm{pred}}]$ within the data set. We then use the estimate of the system states at t^*, simulate the model until $t^* + t_{\mathrm{pred}}$, and check whether an event is falsely predicted within $[t^*, t^* + t_{\mathrm{pred}}]$.

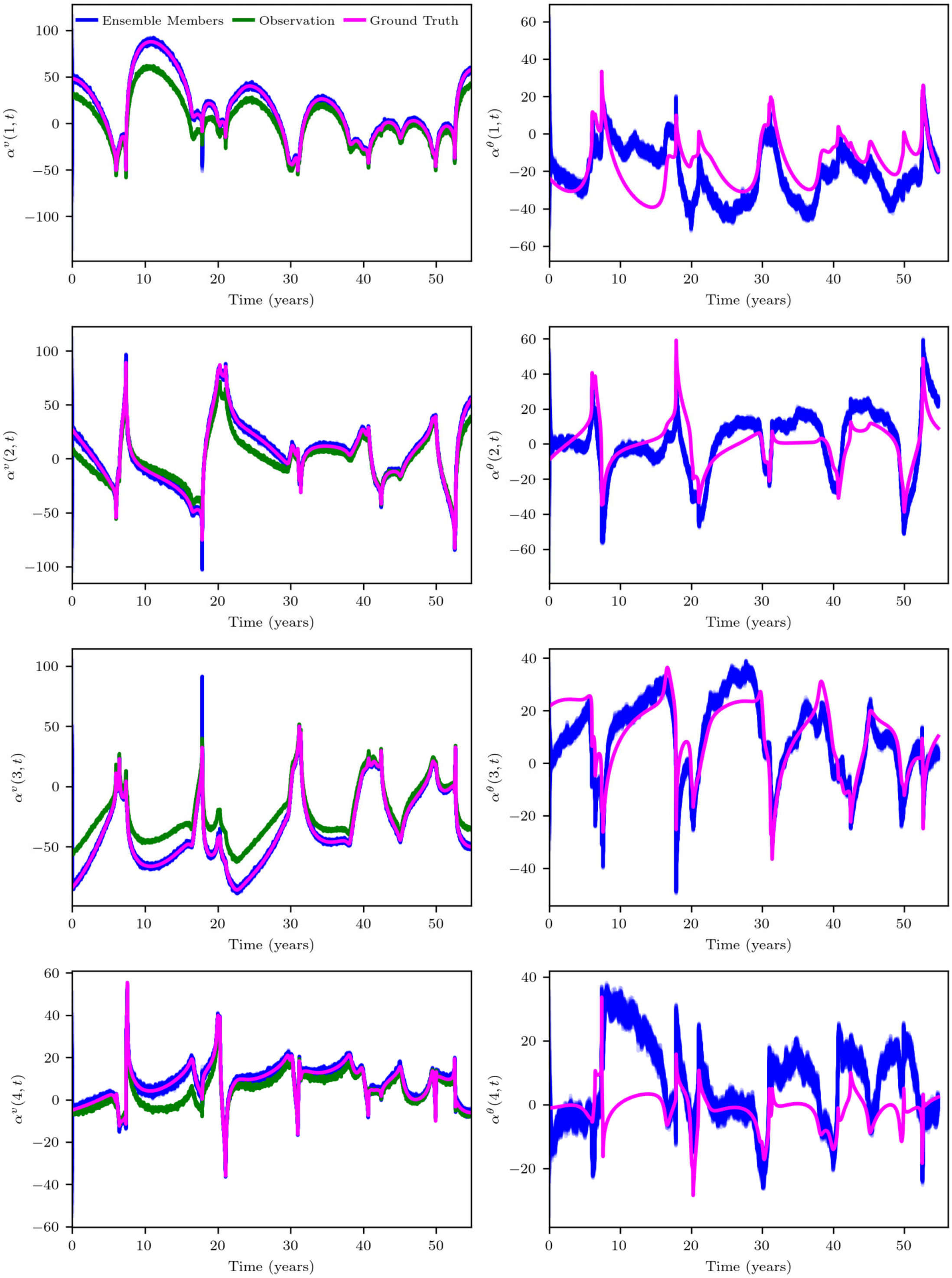

Figure 4. Performance of estimating the temporal components of POD for slip rate ($\alpha^v(i, t)$) and state variable ($\alpha^\theta(i, t)$) using a ROM with $n = 40$ as the forward model. The magenta lines represent the true components of the slip rate and state variable. Blue lines indicate the ensemble members, while green lines show the observed components of the slip rate. Note that no green lines are present in the second column, as the state variable is unobservable.

The FPR is then defined as the ratio of the number of intervals in which at least one false event is predicted to the total number of sampled instances N^{FPR}. Setting $N^{\text{FPR}} = 100$, we obtain an FPR of 0.13.

We compare the performance of our temporal prediction with that of a homogeneous Poisson process. The event rate λ of the Poisson process is empirically estimated as the inverse of the average interevent time over a long simulation. Based on approximately 2500 events, the average interevent time is 3.5 yr, yielding an estimated rate of $\lambda = 1/3.5 = 0.29$ events per year. For a Poisson process, the TPR corresponds to the probability of predicting at least one event in the interval $[t_{\text{event}} - t_{\text{est}}, t_{\text{event}} - t_{\text{est}} + t_{\text{pred}}]$. This probability

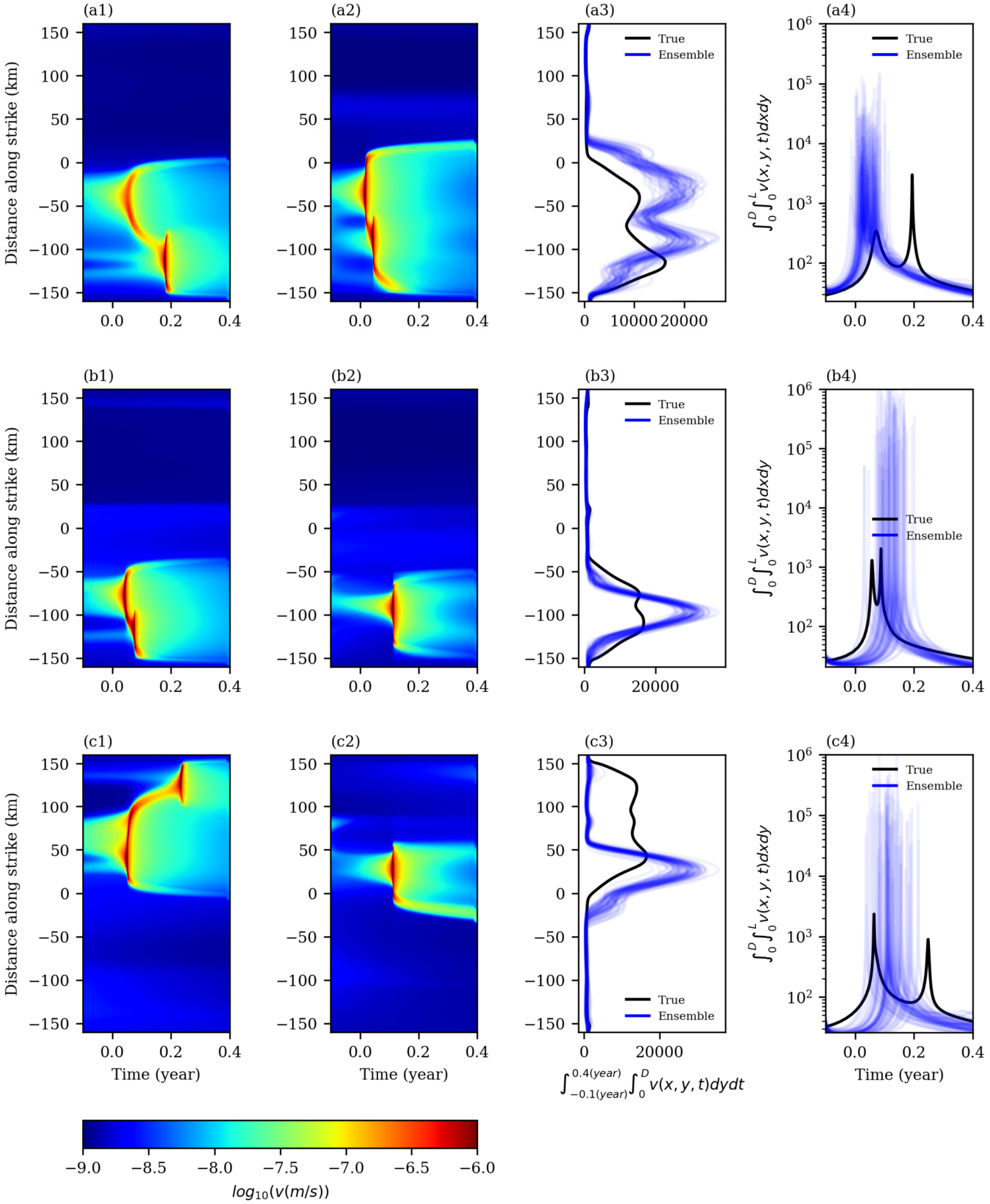

Figure 5. Spatiotemporal evolution of events in true data and predictions for events with $M > 6.9$ with uncertainty quantification. The first column (a1–e1) shows the true maximum slip rate along depth, plotted as a function of position along the strike and time. The second column (a2–e2) presents predictions obtained 0.1 yr before an event starts, based on the estimated slip rate and state variable using the EnKF with a ROM with $n = 40$. The time is shifted to zero at the moment when an event starts in the true signal. The third column shows spatial prediction uncertainty by plotting the slip rate integrated from −0.1 to 0.4 yr over the fault depth for each ensemble member (in blue). The fourth column shows temporal prediction uncertainty by plotting the slip rate integrated over both the depth and strike of the fault.

is given by:

$$P^{\text{Poisson}}(k \geq 1 | \Delta t = t_{\text{pred}}) = 1 - P^{\text{Poisson}}(k = 0 | \Delta t = t_{\text{pred}}), \quad (30)$$

where $P^{\text{Poisson}}(k | \Delta t = t_{\text{pred}})$ denotes the probability of predicting exactly k events in the time interval $\Delta t = t_{\text{pred}}$ using a Poisson process with a rate of $\lambda = 0.29$ events per year. By the properties of a Poisson process, this probability is given by:

$$P^{\text{Poisson}}(k | \Delta t = t_{\text{pred}}) = \frac{(\lambda t_{\text{pred}})^k e^{-\lambda t_{\text{pred}}}}{k!}. \quad (31)$$

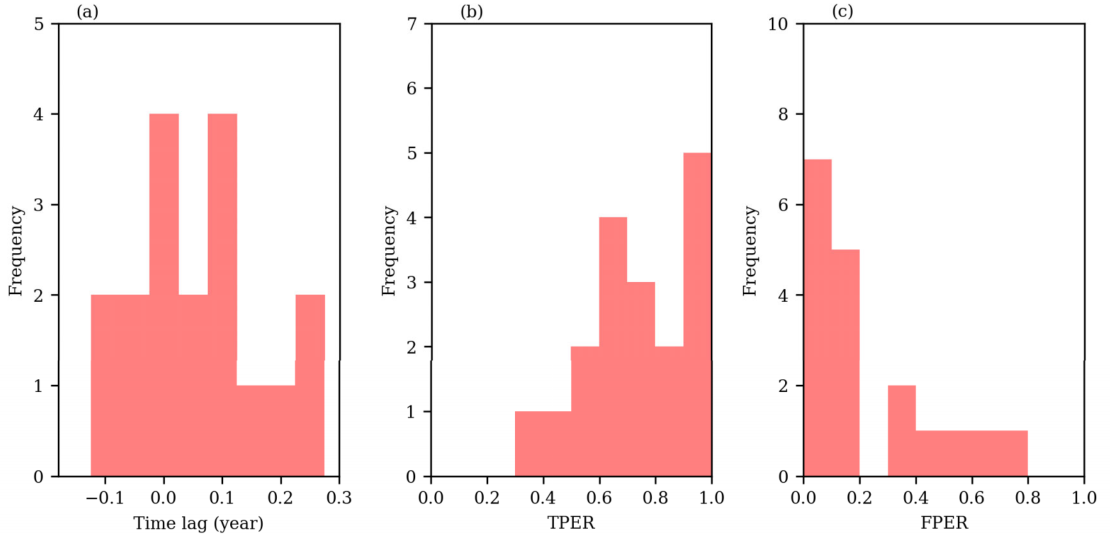

Figure 6. Quantifying the prediction performance in time and space. Histogram of prediction time lag (a), true positive extent ratio (TPER) (b) and false positive extent ratio (FPER) (c).

Thus, the TPR of a Poisson process simplifies to:

$$TPR_{\text{Poisson}} = 1 - e^{-\lambda t_{\text{pred}}}. \tag{32}$$

Similarly, $P^{\text{Poisson}}(k \geq 1 | \Delta t = t_{\text{pred}})$ can be used to compute the FPR. Since FPR is defined as the probability of predicting an event within the interval $[t^*, t^* + t_{\text{pred}}]$ when no event actually occurs in that period, it is also given by:

$$\text{FPR}_{\text{Poisson}} = 1 - e^{-\lambda t_{\text{pred}}}. \tag{33}$$

For small values of t_{pred}, the Poisson process yields an FPR close to zero, which is desirable. However, this comes at the cost of an extremely low TPR, meaning it rarely predicts events. Specifically, for $t_{\text{pred}} = 0.4$ yr and $\lambda = 0.29$ events per year, the Poisson process achieves both a TPR and FPR of 0.11. While the FPR is slightly lower than in our simulations, the TPR is significantly smaller, highlighting the improved predictive performance of our approach.

The data set exhibits complexity not only in time but also in space, as rupture patterns do not repeat periodically. Consequently, spatial prediction is also crucial. Here, we evaluate the spatial prediction performance of events that have been correctly predicted in time, assuming that ruptures fully extend through the depth of the fault. This assumption is valid given the elongated fault geometry and the tendency of ruptures to saturate the depth. Our focus is on prediction performance along the strike direction of the fault. To quantify spatial accuracy, we define two key metrics: the True positive extent ratio (TPER) and the false positive extent ratio (FPER).

The **True Positive Extent Ratio (TPER)** quantifies the proportion of the fault's along-strike extent that both ruptured in the true data and was correctly predicted to rupture. It is defined as

$$\begin{aligned}\text{TPER} &= P(\text{Rupture in prediction} \mid \text{Rupture in true data}) \\ &= \frac{\text{Length of correctly predicted rupture extent}}{\text{Length of true rupture extent}},\end{aligned}$$

or equivalently:

$$\text{TPER} = \frac{|E_{\text{overlap}}|}{|E_{\text{true}}|},$$

where $E_{\text{overlap}} = E_{\text{true}} \cap E_{\text{pred}}$ is the extent of the fault that both ruptured in the true data and was predicted to rupture, E_{true} represents the extent of the fault that ruptured in the true data and E_{pred} represents the extent of the fault predicted to rupture. Intuitively, the TPER quantifies the fraction of the true rupture extent that is successfully captured by the prediction. A TPER of 1 indicates perfect prediction, in a sense that all of the true ruptured area is inside the predicted rupture extent. Lower values of TPER suggest that parts of the true rupture were missed in the prediction.

The **False Positive Extent Ratio (FPER)** evaluates the proportion of the predicted rupture extent that does not correspond to a true rupture. It is defined as

$$\begin{aligned}\text{FPER} &= P(\text{Rupture in prediction} \mid \text{No rupture in true data}) \\ &= \frac{\text{Length of falsely predicted rupture extent}}{\text{Length of unruptured extent in true data}},\end{aligned}$$

or equivalently:

$$\text{FPER} = \frac{|E_{\text{pred}} \setminus E_{\text{true}}|}{|E_{\text{fault}} \setminus E_{\text{true}}|},$$

where $|E_{\text{pred}} \setminus E_{\text{true}}|$ is the extent of the fault predicted to rupture but not ruptured in the true data, and $|E_{\text{fault}} \setminus E_{\text{true}}|$ represents the extent of the fault that did not rupture in the true data, with E_{fault} denoting the total fault length. Intuitively, a lower FPER, ideally zero, indicates that the prediction avoids predicting ruptures in regions where they do not occur. Higher FPER values suggest overprediction and less reliable forecasts.

When TPER equals 1, the model successfully predicts 100 per cent of the rupture extent, and when FPER equals 0, the model perfectly avoids predicting ruptures in unruptured regions. Together, these metrics provide a comprehensive evaluation of the spatial prediction performance, balancing the ability to capture true ruptures with minimizing false alarms. TPER focuses on sensitivity to true events, while FPER emphasizes specificity in avoiding false positives. Using both metrics ensures a nuanced assessment of spatial forecast performance, capturing both accuracy and reliability.

Figs 6(b) and (c) illustrate the spatial performance of event predictions. Fig. 6(b) presents the histogram of TPER, showing that more than 77 per cent of events have a TPER greater than 0.6. Fig. 6(c) displays the histogram of FPER, indicating that over 66 per cent of predicted events have an FPER less than 0.2. These

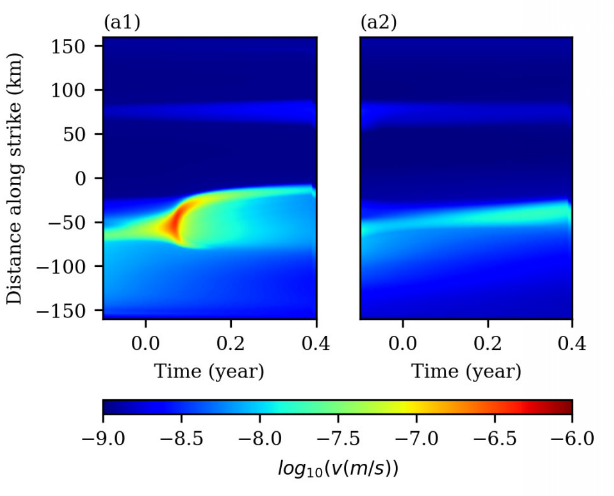

Figure 7. The method fails to predict some small events. (a1) The true maximum slip rate along the depth as a function of time and distance along the strike, showing a small partial rupture with magnitude 6.7. (a2) The corresponding prediction signal, which does not include an event. The prediction is based on the estimation of a model with $n = 40$, at 0.1 yr before the event begins. The figure is aligned such that time = 0 corresponds to the start of the event in the true data.

results demonstrate that the predictions are not only spatially accurate in capturing true ruptures but also effective in minimizing false predictions.

Figs 6(b) and (c) illustrate the spatial performance of event predictions. Fig. 6(b) presents the histogram of TPER, showing that more than 77 per cent of events have a TPER greater than 0.6, indicating that most predicted ruptures capture the majority of the true rupture area. Fig. 6(c) displays the histogram of FPER, showing that over 66 per cent of predicted events have an FPER less than 0.2, suggesting limited overprediction of rupture extent. These results demonstrate that the predictions are spatially consistent with true rupture patterns and effectively minimize false spatial activations.

4 DISCUSSION

4.1 Validity of assumptions and methodological limitations

In this subsection, we examine the core assumptions underlying our framework and discuss their implications, along with some methodological limitations that affect its broader applicability to more realistic earthquake cycle problems. First, we employ a quasi-dynamic approximation for stress transfer on the fault, which neglects wave-mediated effects. This assumption is reasonable for our simulations, as they primarily focus on SSEs where dynamic effects are minimal. However, for faster processes, such as dynamic ruptures, this approximation may no longer be valid, and a fully dynamic model would be necessary.

Additionally, we assume that the model described by eq. (3a, b) represents the ‘true’ system. While this assumption simplifies the analysis and provides a controlled framework for exploring estimation methods, it introduces limitations when applied to real-world scenarios. The physical processes governing the earthquake cycle are inherently complex and not fully captured by this simplified model. Future research could investigate the robustness of our methods under model misspecification or in the presence of additional physical processes, such as fluid migration or inelastic deformation.

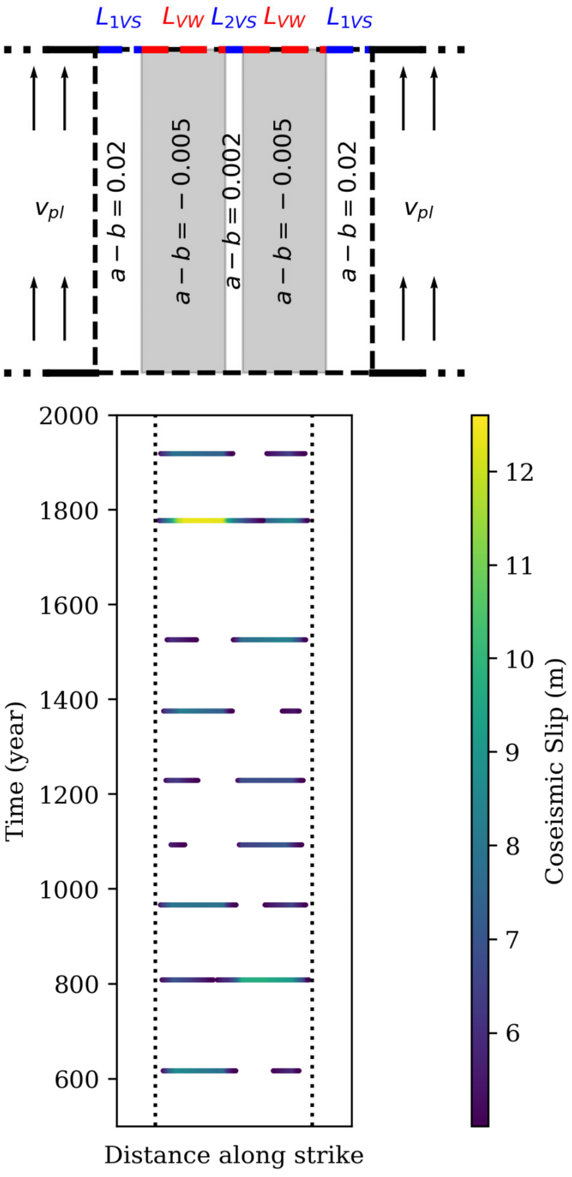

Figure 8. Geometry of a 1-D fault and chaotic earthquake sequences. (a) Fault geometry and spatial distribution of $a - b$ for a 2-D model used to generate earthquake sequences. (b) Coseismic slip along the strike direction over time, thresholded to display only slip greater than 5 m. Vertical dashed lines indicate the locations where $a - b$ transitions from 0.02 to −0.005. This simulation is performed with $d_{\mathrm{rs}} = 12$ mm.

Another key assumption is that the model parameters, such as those describing rate-and-state friction, are perfectly known. This is a significant simplification, as estimating these parameters from observational data remains a major challenge, particularly for earthquakes. Earthquake data is often sparse and noisy, unlike the comparatively richer data sets available for SSEs. This assumption limits the immediate applicability of our method to real-world problems. One possible approach to address this limitation would be to integrate model parameters as inputs into the neural network framework.

Table 2. Parameters used in the model of earthquake (1-D fault)

Category	Property	Value	Units
Frictional properties	d_{rs}	6, 9, 12, 15	mm
	b_{VS1}	−0.01	–
	b_{VW}	0.015	–
	b_{VS2}	0.008	–
	a	0.01	–
Other physical properties	$\bar{\sigma}$	50	MPa
	μ	30	GPa
	c_s	3.3	km s^{-1}
	ν	0.25	–
Loading quantity	V_{pl}	50	mm yr^{-1}
Geometric quantities	L_{VS1}	40	km
	L_{VW}	72.5	km
	L_{VS2}	15	km

This adjustment could allow the model to account for parameter variations dynamically, albeit at a significant computational cost.

Extending our reduced-order modelling framework to simulate fast earthquake ruptures presents additional challenges beyond those encountered with SSEs. Earthquakes exhibit stronger multiscale dynamics in both space and time, with rupture processes unfolding over seconds to minutes. Capturing these dynamics in a machine-learned ROM requires substantially more training data. In our framework, the inputs to the machine learning algorithm are non-uniform-in-time-series of the temporal coefficients associated with the POD modes. For earthquakes, resolving the rupture dynamics requires many more snapshots, due to the necessity of taking significantly smaller time steps during fast events. This leads to a major increase in the size of the training set required to faithfully learn the system's evolution, posing some computational challenges.

Another challenge in extending this framework to fast earthquake simulations lies in the temporal resolution of data assimilation. In the current approach, data assimilation is performed using a uniform time step of five days, which is appropriate for capturing the evolution of slow events. This resolution allows for accurate tracking of SSE dynamics without excessive computational burden. However, earthquake ruptures occur on much shorter timescales and require finer temporal resolution to be accurately resolved and assimilated. Applying uniform fine time steps throughout the simulation would result in a substantial increase in computational cost. Furthermore, to efficiently and accurately capture the rapid dynamics of earthquakes, implementing data assimilation with adaptive time stepping may become necessary.

Moreover, while our ROM successfully reproduces key scaling relationships observed in the full PDE simulations, this outcome emerges without explicitly constraining the machine learning algorithm to preserve such statistical properties. Although we interpret the preservation of these scaling behaviours as a strength of the model, it is important to note that the ROM was not engineered to achieve this outcome. Matching the long-term statistical features of a high-dimensional chaotic attractor remains a challenge in the machine learning and dynamical systems communities (T. Schneider *et al.* 2021; Z. Li *et al.* 2022; J. Park *et al.* 2025). Thus, the success in reproducing these statistics, while encouraging, may not generalize across systems or parameter regimes without further theoretical understanding or architectural constraints. These considerations highlight important limitations of the current method and underscore the need for further methodological development before it can be applied to realistic earthquake problems.

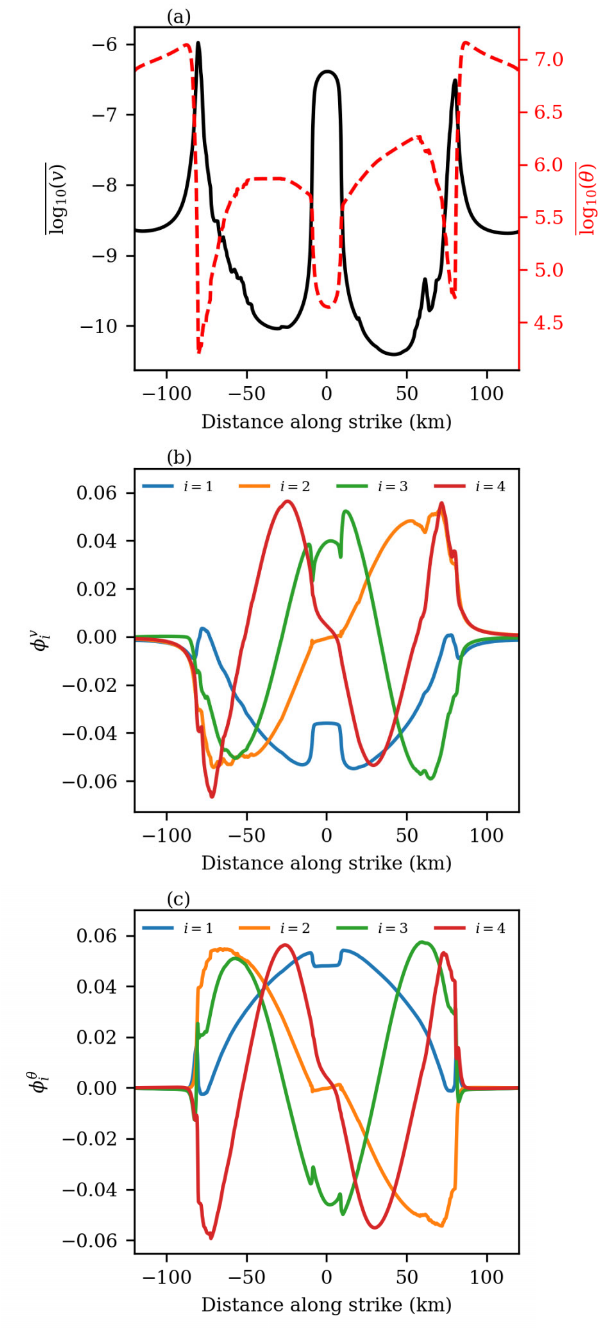

Figure 9. POD analysis of the system's fields (v and θ) in a model of a 1-D fault that produces a chaotic sequence of earthquakes. (a) Snapshot averages of the slip rate ($\overline{\log_{10}(v)}$) and state variable ($\overline{\log_{10}(\theta)}$). (b) The first four eigenmodes for the slip rate (ϕ_i^v). (c) The first four eigenmodes for the state variable (ϕ_i^θ). The POD is applied on a model with $d_{\mathrm{rs}} = 12$ mm.

4.2 Observability

In theory, it is not always necessary or even expected to recover the full state of a system, particularly for unobservable components. This relates to the concept of *observability* in dynamical systems. A system is considered observable if the observed variables can be used to reconstruct all the states of the system. For chaotic dynamical systems, this concept extends to *chaos synchronization*, where synchronization occurs when partial observations of a chaotic system can be used to recover the unobserved states. If this is

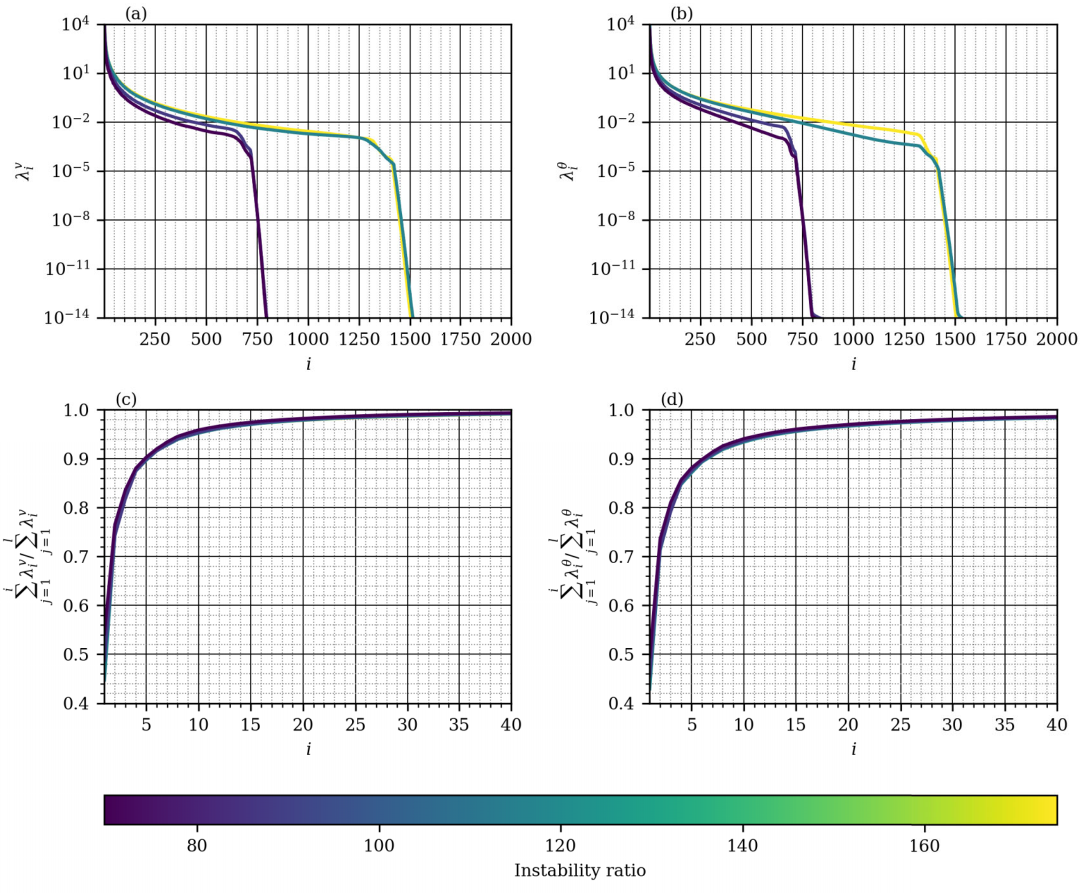

Figure 10. Effect of instability ratio on the reducibility. (a, b) Variance of each mode (i) in the singular value decomposition for the slip rate (a) and the state variable (b). (c, d) The ratio of the sum of the first i eigenvalues to the total sum of all eigenvalues (as defined in eq. 11) for the slip rate (c) and the state variable (d). All panels correspond to simulations with different instability ratios.

possible, the system is said to be *synchronizable* (L.M. Pecora & T.L. Carroll 1990).

Our findings suggest that the original full-scale model, although chaotic, might be synchronizable when only the slip rate (v) is observed. Specifically, if two simulators are governed by the same equations and model parameters but have different initial values of the state variable (θ), the second system can synchronize with the first by using the observed slip rate from the first system. Even starting from different initial conditions for θ, the state variable in both systems would eventually synchronize over time.

This observation is physically reasonable because the state variable (θ) acts as a memory of the contact state in the system. When the slip rate is imposed, the state variable eventually converges to the same value in both systems, as the system 'forgets' its initial condition and adjusts to follow the imposed slip rate.

This observation has important implications beyond the reduced-order modelling framework. It suggests that observing only the slip rate may be sufficient to recover the state variable, from which one can infer stress on the fault. This finding is broadly consistent with results from machine-learning forecasts of laboratory earthquakes, which identified slip rate as the dominant predictor of imminent failure and the key quantity controlling radiated seismic energy (B. Rouet-Leduc *et al.* 2017; C. Hulbert *et al.* 2019; D.C. Bolton *et al.* 2020). If confirmed more generally, this relationship would have significant practical implications for data assimilation and time-dependent earthquake forecasting.

Several open questions naturally arise from this observation. For instance, how much history of slip rate data is required to recover the state variable accurately? In other words, what is the synchronization time needed for the system to converge when only the slip rate is observed? Understanding the required synchronization horizon is essential for designing data assimilation systems that rely on partial observations. These questions motivate future work on observability and chaos synchronization in complex fault models.

4.3 Prediction of small events

Real observational data typically undergoes smoothing, which suppresses high-frequency spatial information. To emulate this characteristic, our synthetic observed slip rate is also smoothed, resulting in similar limitations in spatial resolution. The ROM used in our approach is constructed to capture the dominant large-scale structures represented by the leading POD modes. Moreover, while the slip rate coefficients $\alpha^v(i, t)$ for higher modes can often be recovered, the accuracy of the state variable coefficients $\alpha^\theta(i, t)$ generally degrades as the mode number i increases.

Because of these limitations, accurately representing and forecasting smaller events in the true signal—for example, those with moment magnitude below 6.9—becomes particularly challenging. The cutoff magnitude of 6.9 was chosen based on our observation that the forecast performance degrades notably for events smaller than this threshold. Fig. 7 illustrates one such small event that the method fails to predict. While some degree of predictability exists for events below this threshold, this example highlights a specific failure to capture the underlying small-scale processes. This limitation aligns with recent studies showing that small earthquakes are inherently harder to forecast due to their sensitivity to small-scale stress heterogeneities (P. Venegas-Aravena & D. Zaccagnino 2025).

4.4 Effect of instability ratio on reducibility

The instability ratio, defined as the ratio of the fault size to the nucleation size, plays a critical role in determining fault behaviour (Y. Liu & J.R. Rice 2007). It is given by L/h_{ra}, where L is the length of the fault, and $h_{\mathrm{ra}} = \frac{2\mu' d_{\mathrm{rs}} b}{\pi\sigma(b_{\mathrm{VW}}-a_{\mathrm{VW}})^2}$ is the nucleation size (A.M. Rubin & J.P. Ampuero 2005). μ' is the shear modulus (μ) for antiplane shear and $\mu/(1-\nu)$ for plane strain, where ν is the Poisson ratio. Although the relationship between the instability ratio and event complexity is not strictly monotonic, higher instability ratios generally correlate with increased rupture complexity. When the instability ratio is small and close to one, events are more likely to exhibit lower maximum slip rates and primarily produce SSEs—a sequence of slip events characterized by smaller maximum slip rates compared to earthquakes. Our simulations with a 2-D fault have thus operated within this regime. An important question that arises is how representative the leading POD modes remain of the system's overall behaviour as the instability ratio increases.

In this part, we answer this question in the context of a 1-D fault that generates fast earthquakes. To investigate, we systematically vary the nucleation size by modifying the characteristic slip distance (d_{rs}) and applying SVD to the resulting data set. This analysis enables us to investigate how the eigenvalues of the modes evolve in response to changes in the instability ratio.

We model a finite 1-D fault embedded in an elastic medium, using the same model as in eq. (3a, b), while varying the characteristic slip distance (d_{rs}) to modify the instability ratio. The fault geometry, incorporating heterogeneous material properties, is illustrated in Fig. 8. The physical parameters for this case study are similar to those in (M.Y. Thomas *et al.* 2014), with d_{rs} varied to explore the effects of differing instability ratios. A summary of the physical properties is provided in Table. 2. The coseismic slip above 5 (m) from the year 500 to 2000 for a simulation with $d_{\mathrm{rs}} = 12$ (mm) is shown in Fig. 8.

For each d_{rs} value, we run the forward model and record the same number of snapshots (70 000) of the slip rate and state variable. SVD is then applied to the snapshots of these fields. The components of POD are plotted in Fig. 9. Qualitatively, the eigenmodes exhibit distinct patterns corresponding to different stages of the earthquake cycle. For example, some modes capture ruptures localized within a single VW zone, while others represent ruptures that penetrate the central VS zone and produce large slips in both VW zones. Interestingly, as the mode number increases, the spatial frequency captured by the modes also increases. This trend aligns with the observations in Fig. 2 (for simulation of SSEs), highlighting that POD consistently identifies modes that first capture the dominant large-scale spatial structures before progressing to finer details.

This result is significant because the observational data typically available for real faults have limited spatial resolution and lack information about small-scale spatial processes. The robustness of POD in prioritizing large-scale structures suggests that constructing a ROM based on the projection of fields onto the POD modes is particularly advantageous. Such a ROM takes input primarily from the large-scale structures, making it compatible with the coarse, low-resolution data that are realistically accessible while still preserving the essential dynamics of the system.

The variances of the eigenmodes for the slip rate (λ^{v}) and the state variable (λ^{θ}) are plotted in Figs 10(a) and (b), corresponding to different values of d_{rs} (and thus, different instability ratios). As the instability ratio increases, there is a slight increase in the eigenvalues of the higher modes. However, this effect is relatively small and, to leading order, we do not see significant changes in the eigenvalues with an increase in the nucleation size. This is more apparent in Figs 10(c) and (d). The bottom panels show the ratio of the cumulative variance of the first i modes to the total variance across all modes. This ratio remains nearly identical for all instability ratios studied here. These results demonstrate that, to first order, the leading eigenmodes retain their relative statistical importance even as the instability ratio increases.

A key question arising from these results concerns the predictability of the nucleation phase. At first glance, Figs 10(c) and (d) might suggest that increasing the instability ratio does not significantly affect the predictability of the nucleation phase. However, this interpretation should be made with caution. The analysis only shows that the relative variance contained in the leading modes remains similar; it does not rule out the possibility that low-variance modes, which are not prominent in PCA, could still play a critical role in the dynamics. In analogous situations in fluid dynamics, rare low-energy modes in the proper orthogonal decomposition of Kolmogorov flow have been shown to underlie the occurrence of intermittent, burst-like events (M. Farazmand & T.P. Sapsis 2017). A similar mechanism could, in principle, contribute to variability during the nucleation phase of fault slip.

5 CONCLUSION

This study presents a machine-learned ROM developed to simulate chaotic multiscale sequences of slip events. By integrating POD with machine learning, the ROM efficiently captures the dominant dynamics of the earthquake cycle. The reduced dimensionality allows for significantly faster computations compared to full PDE models, while preserving essential scaling laws and statistical features. Our results demonstrate that the ROM replicates long-term statistical properties of the sequence—such as magnitude–frequency, moment–duration and moment–area scaling relations—consistently with full-scale PDE simulations. The ROM emphasizes large-scale structures in the slip rate and state variable fields, consistent with the coarse resolution of realistic observational data sets. This makes it particularly suitable for earthquake forecasting applications, where small-scale features are typically unresolvable due to smoothing in inversion processes.

The study also highlights the successful integration of the EnKF within the ROM framework to estimate the temporal components of POD from sparse and noisy observational data. While some inaccuracy persists in the reconstruction of state variable components, ensemble-averaged forecasts reliably predict the timing and location of large events.

Nevertheless, several limitations remain. The ROM is currently applied to synthetic SSEs, and extending the method to simulate realistic fast earthquake ruptures presents additional challenges. Earthquakes exhibit more pronounced multiscale behaviour in both time and space, with rapid rupture dynamics occurring over seconds to minutes. Capturing such fast dynamics may require modifications to the ROM architecture and data assimilation strategy. These issues underscore the need for future development before the method can be applied to dynamic earthquake modelling.

Furthermore, the quasi-dynamic approximation used in this study, while suitable for SSEs, may not adequately represent the physics of rapid rupture, motivating a transition to fully dynamic models. The assumption of perfectly known model parameters also simplifies the analysis but limits real-world applicability. Incorporating parameter estimation into the ROM—potentially by extending the neural network to learn parameter dependencies—could improve realism, though at a higher computational cost. Finally, the ROM's focus on dominant modes limits its ability to capture small-scale features.

In summary, this work introduces a robust and efficient framework for modelling multiscale chaotic sequence of events, demonstrating the potential of combining physics-informed machine learning with data assimilation for advancing earthquake forecasting. While the current implementation is validated in synthetic settings, its scalability, efficiency and compatibility with realistic observational data offer a promising pathway toward practical applications in seismology.

ACKNOWLEDGMENTS

The authors would like to thank Oliver Dunbar, Hanieh Mousavi, Dave May and Alice-Agnes Gabriel for insightful discussions, which greatly contributed to the development of this work. The authors, HK and J-PA, acknowledge the support of the National Science Foundation (NSF) through the Industry–University Collaborative Research Center for Geomechanics and Mitigation of Geohazards (Award No. 1822214). AMS is supported by a Department of Defense Vannevar Bush Faculty Fellowship and through the ONR MURI on Data-Driven Closure Relations N00014-23-1-2654. The majority of the simulations presented in this study were performed using the Resnick High-Performance Computing Center at the California Institute of Technology.

DATA AVAILABILITY

The forward model simulations in this study are conducted using the open-source QDYN software (Y. Luo *et al.* 2017), available at https://github.com/ydluo/qdyn. The codes to generate synthetic data sets used for training, along with the machine learning and data assimilation scripts developed for this study, are available at https://github.com/hojjatks/DA-in-ROM-of-earthquake.

REFERENCES

Barbot, S., 2019. Slow-slip, slow earthquakes, period-two cycles, full and partial ruptures, and deterministic chaos in a single asperity fault, *Tectonophysics,* **768,** 228171.

Barbot, S., Lapusta, N. & Avouac, J.P., 2012. Under the hood of the earthquake machine: toward predictive modeling of the seismic cycle, *Science,* **336**(6082), 707–710.

Bolton, D.C., Shreedharan, S., Rivière, J. & Marone, C., 2020. Acoustic energy release during the laboratory seismic cycle: insights on laboratory earthquake precursors and prediction, *J. geophys. Res.: Solid Earth,* **125**(8), e2019JB018975.

Costantino, G., Radiguet, M., Yousfi, Z.E. & Socquet, A., 2025. A continuum of slow slip events in the Cascadia subduction zone illuminated by high-resolution deep-learning denoising, *Authorea Preprints*.

Dempsey, D. & Suckale, J., 2017. Physics-based forecasting of induced seismicity at Groningen gas field, the Netherlands, *Geophys. Res. Lett.,* **44**(15), 7773–7782.

Diab-Montero, H.A., Li, M., van Dinther, Y. & Vossepoel, F.C., 2023. Estimating the occurrence of slow slip events and earthquakes with an ensemble Kalman filter, *Geophys. J. Int.,* **234**(3), 1701–1721.

Dieterich, J.H., 1979. Modeling of rock friction: 1. Experimental results and constitutive equations, *J. geophys. Res.: Solid Earth,* **84**(B5), 2161–2168.

Evensen, G., 1994. Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics, *J. geophys. Res.: Oceans,* **99**(C5), 10143–10162.

Farazmand, M. & Sapsis, T.P., 2017. A variational approach to probing extreme events in turbulent dynamical systems, *Sci. Adv.,* **3**(9), e1701533.

Field, E.H. *et al.*, 2015. Long-term time-dependent probabilities for the third Uniform California Earthquake Rupture Forecast (UCERF3), *Bull. seism. Soc. Am.,* **105**(2A), 511–543.

Fukami, K. & Taira, K., 2023. Grasping extreme aerodynamics on a low-dimensional manifold, *Nat. Commun.,* **14**(1), 6480.

Fukushima, R., Kano, M. & Hirahara, K., 2023. Physics-informed neural networks for fault slip monitoring: simulation, frictional parameter estimation, and prediction on slow slip events in a spring-slider system, *J. geophys. Res.: Solid Earth,* **128**(12), e2023JB027384.

Gualandi, A., Avouac, J.P., Michel, S. & Faranda, D., 2020. The predictable chaos of slow earthquakes, *Sci. Adv.,* **6**(27), eaaz5548.

Hawkins, R., Khalid, M.H., Smetana, K. & Trampert, J., 2023. Model order reduction for seismic waveform modelling: inspiration from normal modes, *Geophys. J. Int.,* **234**(3), 2255–2283.

Hirahara, K. & Nishikiori, K., 2019. Estimation of frictional properties and slip evolution on a long-term slow slip event fault with the ensemble Kalman filter: numerical experiments, *Geophys. J. Int.,* **219**(3), 2074–2096.

Hobson, G.M. & May, D.A., 2025. Sensitivity analysis of the thermal structure within subduction zones using reduced-order modeling, *Geochemistry, Geophysics, Geosystems,* **26,** e2024GC011937.

Hulbert, C., Rouet-Leduc, B., Johnson, P.A., Ren, C.X., Rivière, J., Bolton, D.C. & Marone, C., 2019. Similarity of fast and slow earthquakes illuminated by machine learning, *Nat. Geosci.,* **12**(1), 69–74.

Kaveh, H., Batlle, P., Acosta, M., Kulkarni, P., Bourne, S.J. & Avouac, J.P., 2023. Induced seismicity forecasting with uncertainty quantification: application to the groningen gas field, *Seismol. Res. Lett.,* **95**(2A), 773–790.

Kaveh, H., Avouac, J.P. & Stuart, A.M., 2025. Spatiotemporal forecast of extreme events in a chaotic model of slow slip events, *J. geophys. Int.,* **240**(2), 870–885.

Kositsky, A.P. & Avouac, J.P., 2010. Inverting geodetic time series with a principal component analysis-based inversion method, *J. geophys. Res.: Solid Earth,* **115**(B3). doi: 10.1029/2009JB006535

Krishnapriyan, A.S., Gholami, A., Zhe, S., Kirby, R.M. & Mahoney, M.W., 2021. Characterizing possible failure modes in physics-informed neural networks, in *NeurIPS 2021 · Thirty-Fifth Annual Conference on Neural Information Processing Systems,* Virtual-only Conference. [cs].

Lapusta, N. & Liu, Y., 2009. Three-dimensional boundary integral modeling of spontaneous earthquake sequences and aseismic slip, *J. geophys. Res.: Solid Earth,* **114**(B9).

Lapusta, N., Rice, J.R., Ben-Zion, Y. & Zheng, G., 2000. Elastodynamic analysis for slow tectonic loading with spontaneous rupture episodes on faults with rate- and state-dependent friction, *J. geophys. Res.: Solid Earth,* **105**(B10), 23765–23789.

Law, K., Stuart, A. & Zygalakis, K., 2015. *Data Assimilation: A Mathematical Introduction,* vol. **62** of **Texts in Applied Mathematics**, Springer International Publishing.

Le Provost, M. & Eldredge, J.D., 2021. Ensemble Kalman filter for vortex models of disturbed aerodynamic flows, *Phys. Rev. Fluids,* **6**(5), 050506.

Li, M., Jain, S. & Haller, G., 2023. Model reduction for constrained mechanical systems via spectral submanifolds, *Nonlin. Dyn.,* **111**(10), 8881–8911.

Li, Z., Liu-Schiaffini, M., Kovachki, N., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A. & Anandkumar, A., 2022. Learning dissipative dynamics in chaotic systems, in *The Thirty-Sixth Annual*

Conference on Neural Information Processing Systems, Hybrid Conference. [cs, math].

Liu, C., Fu, R., Xiao, D., Stefanescu, R., Sharma, P., Zhu, C., Sun, S. & Wang, C., 2022. EnKF data-driven reduced order assimilation system, *Eng. Anal. Bound. Elements,* **139,** 46–55.

Liu, Y. & Rice, J.R., 2007. Spontaneous and triggered aseismic deformation transients in a subduction fault model, *J. geophys. Res.: Solid Earth,* **112**(B9). doi: 10.1029/2007JB004930.

Lubbers, J., Loewen, M., Wallace, K., Coombs, M. & Addison, J., 2023. Probabilistic Source Classification of Large Tephra Producing Eruptions Using Supervised Machine Learning: An Example From the Alaska-Aleutian Arc, *Geochem. Geophys. Geosyst.,* **24**(11), e2023GC011037.

Luo, Y., Ampuero, J.P., Galvez, P., Ende, M.v.d. & Idini, B., 2017. QDYN: a Quasi-DYNamic earthquake simulator (v1.1). Zenodo. doi: 10.5281/zenodo.322459

Main, I., 1996. Statistical physics, seismogenesis, and seismic hazard, *Rev. Geophys.,* **34**(4), 433–462.

Maulik, R. *et al.*, 2022. Efficient high-dimensional variational data assimilation with machine-learned reduced-order models, *Geosci. Model Dev.,* **15**(8), 3433–3445.

Michel, S., Gualandi, A. & Avouac, J.P., 2019. Similar scaling laws for earthquakes and Cascadia slow-slip events, *Nature,* **574**(7779), 522–526.

Mousavi, H. & Eldredge, J.D., 2025. Low-order flow reconstruction and uncertainty quantification in disturbed aerodynamics using sparse pressure measurements, *J. Fluid Mech.,* **1013,** A41.

Mousavi, H., Jones, A. & Eldredge, J., 2025. Sequential estimation of disturbed aerodynamic flows from sparse measurements via a reduced latent space, preprint [physics].

Nagata, T., Nakai, K., Yamada, K., Saito, Y., Nonomura, T., Kano, M., Ito, S. & Nagao, H., 2023. Seismic wavefield reconstruction based on compressed sensing using data-driven reduced-order model, *Geophys. J. Int.,* **233**(1), 33–50.

Park, J., Yang, N. & Chandramoorthy, N., 2025. When are dynamical systems learned from time series data statistically accurate?, preprint [cs].

Pecora, L.M. & Carroll, T.L., 1990. Synchronization in chaotic systems, *Phys. Rev. Lett.,* **64**(8), 821–824.

Rathore, P., Lei, W., Frangella, Z., Lu, L. & Udell, M., 2024. Challenges in training PINNs: a loss landscape perspective, preprint [cs].

Rekoske, J.M., May, D.A. & Gabriel, A.A., 2025. Reduced-order modeling for complex 3D seismic wave propagation, *Geophys. J. Int.,* **241,** 526–548.

Rice, J.R., 1993. Spatio-temporal complexity of slip on a fault, *J. geophys. Res.: Solid Earth,* **98**(B6), 9885–9907.

Rice, J.R. & Ruina, A.L., 1983. Stability of steady frictional slipping, *J. Appl. Mech.,* **50**(2), 343–349.

Richards-Dinger, K. & Dieterich, J.H., 2012. RSQSimEarthquake Simulator, *Seismol. Res. Lett.,* **83**(6), 983–990.

Rogers, G. & Dragert, H., 2003. Episodic tremor and slip on the Cascadia subduction zone: the chatter of silent slip, *Science,* **300**(5627), 1942–1943.

Rouet-Leduc, B., Hulbert, C., Lubbers, N., Barros, K., Humphreys, C.J. & Johnson, P.A., 2017. Machine learning predicts laboratory earthquakes, *Geophys. Res. Lett.,* **44**(18), 9276–9282.

Rouet-Leduc, B., Hulbert, C. & Johnson, P.A., 2019. Continuous chatter of the Cascadia subduction zone revealed by machine learning, *Nat. Geosci.,* **12**(1), 75–79.

Rubin, A.M. & Ampuero, J.P., 2005. Earthquake nucleation on (aging) rate and state faults, *J. geophys. Res.: Solid Earth,* **110**(B11). doi: 10.1029/2005JB003686.

Rucker, C. & Erickson, B.A., 2024. Physics-informed deep learning of rate-and-state fault friction, *Comput. Methods Appl. Mech. Eng.,* **430,** 117211.

Ruina, A., 1983. Slip instability and state variable friction laws, *J. geophys. Res.: Solid Earth,* **88**(B12), 10 359–10 370.

San, O. & Maulik, R., 2018. Extreme learning machine for reduced order modeling of turbulent geophysical flows, *Phys. Rev. E,* **97**(4), 042322.

Sanz-Alonso, D. & Stuart, A.M., 2015. Long-time asymptotics of the filtering distribution for partially observed chaotic dynamical systems, *SIAM/ASA J. Uncertainty Quantification,* **3**(1), 1200–1220.

Sanz-Alonso, D., Stuart, A.M. & Taeb, A., 2023. Inverse problems and data Assimilation, preprint [stat].

Schneider, T., Stuart, A.M. & Wu, J.L., 2021. Learning stochastic closures using ensemble Kalman inversion, *Trans. Math. Its Appl.,* **5**(1), tnab003.

Shaw, B.E., Milner, K.R., Field, E.H., Richards-Dinger, K., Gilchrist, J.J., Dieterich, J.H. & Jordan, T.H., 2018. A physics-based earthquake simulator replicates seismic hazard statistics across California, *Sci. Adv.,* **4**(8), eaau0688.

Steger, S., Rohrhofer, F.M. & Geiger, B., 2022. How PINNs cheat: Predicting chaotic motion of a double pendulum: The Symbiosis of Deep Learning and Differential Equations II, in *36th Neural Information Processing Systems (NeurIPS) Conference,* New Orleans, USA.

Taira, K. *et al.*, 2017. Modal Analysis of Fluid Flows: An Overview, *AIAA J.,* **55**(12), 4013–4041.

Thomas, M.Y., Lapusta, N., Noda, H. & Avouac, J.P., 2014. Quasi-dynamic versus fully dynamic simulations of earthquakes and aseismic slip with and without enhanced coseismic weakening, *J. geophys. Res.: Solid Earth,* **119**(3), 1986–2004.

Venegas-Aravena, P. & Zaccagnino, D., 2025. Large earthquakes are more predictable than smaller ones, *Seismica,* **4**(1). doi: 10.26443/seismica.v4i1.1568.

Wang, S., Teng, Y. & Perdikaris, P., 2021. Understanding and mitigating gradient pathologies in physics-informed neural networks, *SIAM J. Sci. Comput.,* **43,** [cs]. doi: 10.1137/20M1318043.

APPENDIX A: DATA PREPARATION FOR TRAINING

Learning chaotic dynamical systems is inherently challenging and remains an active area of research. Machine-learned chaotic systems inevitably diverge from the original system over time, as small inaccuracies compound due to chaos. In our case, these challenges are heightened by learning the system in a reduced dimension, where simulations of eqs (3) and (13), despite starting from the same initial conditions, eventually diverge.

Despite this divergence, it is crucial to ensure that the machine-learned model accurately captures the system's dynamics to preserve its long-term statistical properties. The objective is for the reduced-order model (eq. 13) to replicate the statistical behaviour of the original full-scale system (eq. 3a, b), even if the exact trajectories diverge during long-term simulations. However, since g is tasked with learning a chaotic attractor that projects an infinite-dimensional system onto a finite-dimensional space $\mathbb{R}^{n_v+n_\theta}$, some degree of deviation is unavoidable. This deviation stems from the inherent limitations of approximating an infinite-dimensional attractor with a lower dimensional representation.

Since our goal is to learn an attractor that does not maintain a one-to-one relation with the original attractor, we enrich the data set by including not only points on the chaotic attractor but also points away from it. This ensures that the machine-learning model is exposed to the attractor as well as transient states, improving its ability to generalize.

Here, we explain how the machine-learning model is exposed to data off the chaotic attractor. Intuitively, we achieve this by starting with initial conditions that are statistically more spread than the attractor and using their transient evolution. To approximate the projection of the chaotic attractor onto the POD modes ($\mathcal{A}^\perp$), we use the following formulation:

$$\begin{aligned}\log_{10}(\mathcal{A}^\perp) &= (\log_{10} v, \log_{10}\theta)\\ &\approx \left\{ \left(\overline{\log_{10}(v)} + \sum_{j=1}^{n_v} \alpha_j^v \phi_j^v, \overline{\log_{10}(\theta)} + \sum_{j=1}^{n_\theta} \alpha_j^\theta \phi_j^\theta \right) \middle| \alpha^v \in \mathbb{R}^{n_v}, \right.\\ &\qquad \left. \alpha^\theta \in \mathbb{R}^{n_\theta}, \alpha^v \sim \mathcal{N}(0, \Lambda^v), \alpha^\theta \sim \mathcal{N}(0, \Lambda^\theta) \right\}, \qquad \text{(A1)}\end{aligned}$$

where $\overline{\log_{10}(v)}$ and $\overline{\log_{10}(\theta)}$ are the snapshot averages of the base-10 logarithm of the field, ϕ^v and ϕ^θ are the spatial components obtained using POD and shown in Fig. 2, and α^v and α^θ are the temporal components. Λ^v and Λ^θ are diagonal matrices derived from singular value decomposition, and contain the variance of each component. eq. (A1) provides an approximation of the projection of the attractor because it assumes a normal distribution for the temporal components.

To expose the ML model to points outside the attractor, we intentionally initialize the simulations away from the attractor (eq. A1) to capture more transient dynamics. This approach ensures that the machine-learned model is robust to inputs that do not lie on the attractor. The initial conditions are sampled using the following equations:

$$\alpha^v \sim \mathcal{N}(0, 4\Lambda^v), \tag{A2a}$$

$$\alpha^\theta \sim \mathcal{N}(0, 4\Lambda^\theta). \tag{A2b}$$

In other words, the initial conditions for all simulations are imposed to have a distribution that is more spread than the attractor itself. This generates a data set that includes points away from the attractor and makes our machine-learned model robust to inputs that are not on the attractor. We use 100 simulations based on the model described in eq. (3a, b), using the QDYN simulator (Y. Luo *et al.* 2017), each simulated for 250 years.

These considerations are not sufficient for learning a chaotic attractor that can be simulated for an arbitrarily long time. When evolving the ML model over an extended period, the trajectory might reach regions where the ML model has not encountered any data set. Since the ML model has not seen such cases, the solution may diverge. This is a common challenge when learning chaotic dynamical systems. To address this issue, we adopt one of the methods proposed in (M. Li *et al.* 2023) for learning dissipative chaotic dynamical systems.

M. Li *et al.* (2023) proposed two methods for learning dissipative dynamical systems. In the first method, they synthetically add dissipative data away from the chaotic attractor to ensure that the dynamical system learned using machine learning remains dissipative everywhere, including regions where the ML model has not seen any data. The second method ensures dissipativity by setting a threshold for the norm of the system states. When the states exceed this threshold, the ML algorithm is bypassed, and a simple linear dissipative system is used instead. In our approach, we have adopted the first method.

To generate the dissipative data set, we first sample $N_{\text{dissipation}} = 60\,000$ points $x \in \mathbb{R}^{n_v+n_\theta}$ from a normal distribution with mean r_{outer} and variance I:

$$x \sim \mathcal{N}(r_{\text{outer}}, I).$$

Next, we discard any x with a norm smaller than r_{outer}, removing approximately half of the sampled points. The remaining points are evolved using a linear dissipative dynamic defined as:

$$\dot{x} = \log\left(\frac{r_{\text{inner}}}{r_{\text{outer}}}\right) x.$$

Under this evolution, the points move closer to the center. For example, a point $x(0)$ with $\|x(0)\| = r_{\text{outer}}$ evolves to $\|x(1)\| = r_{\text{inner}}$ after one time step, with the norm of x decreasing over time. We then scale x and their one-step evolution in time using the standard deviation derived from the POD. The scaled data is included in the training set to enforce dissipation in regions away from the chaotic attractor.

One should be careful with the values of r_{outer}, r_{inner}, and the number of dissipative data points, $N_{\text{dissipation}}$. The values of r_{outer} and r_{inner} are chosen such that the dissipative data set does not interfere with the attractor. Additionally, the number of dissipative data points, $N_{\text{dissipation}}$, is kept small compared to the total data set size to maintain the focus on learning the chaotic dynamics. In fact, it should be as small as possible to minimize the effect of these points on the learning of the system dynamics. In our case, the number of additional synthetic data points added to the data set of PDE simulations constitutes only about 3 per cent of the total data set. The values $r_{\text{outer}} = 20$ and $r_{\text{inner}} = 19$ are carefully adjusted to ensure that the dynamics in eq. (13) do not diverge and remain minimally affected by the inclusion of these data.

APPENDIX B: NEURAL NETWORK STRUCTURE

As described in Section 2.3, we decompose the function g in eq. (13), into two functions g_1 and g_2. This is because the $\dot{\alpha}$ has a multiscale (slow/fast) behaviour. In this section, we provide the structure of the neural networks that are used in this paper to learn the functions g_1 and g_2.

The machine learning models used in this study are fully connected feedforward neural networks, $g_1 : \mathbb{R}^n \to \mathbb{R}^n$ and $g_2 : \mathbb{R}^{n_v+1} \to \mathbb{R}^+$, parametrized by ω_1 and ω_2, respectively, which include the weights and biases of the network. The networks consist of an input layer, four hidden layers and an output layer. The models are trained to minimize the mean squared error (MSE) loss function and are optimized using the Adam optimizer. The mathematical structure for g_1 and g_2 is described as follows.

B1 Structure of g_1

The input of the neural network g_1 is a vector $\alpha \in \mathbb{R}^n$. In the first hidden layer, the input undergoes a linear transformation followed by a nonlinear activation function:

$$h_1^1 = \tanh\left(W_1^1 \alpha + b_1^1\right), \quad W_1^1 \in \mathbb{R}^{2n \times n},$$
$$b_1^1 \in \mathbb{R}^{2n}.$$

The superscript specifies the neural network. The second hidden layer applies another linear transformation and activation function to the output of the first layer:

$$h_2^1 = \tanh(W_2^1 h_1^1 + b_2^1), \quad W_2^1 \in \mathbb{R}^{4n \times 2n}, \quad b_2^1 \in \mathbb{R}^{4n}.$$

The third hidden layer maps its input to the same dimensionality as the previous layer:

$$h_3^1 = \tanh(W_3^1 h_2 + b_3^1), \quad W_3^1 \in \mathbb{R}^{4n \times 4n}, \quad b_3^1 \in \mathbb{R}^{4n}.$$

The fourth hidden layer reduces the dimensionality of its input:

$$h_4^1 = \tanh(W_4^1 h_3^1 + b_4^1), \quad W_4^1 \in \mathbb{R}^{2n \times 4n}, \quad b_4^1 \in \mathbb{R}^{2n}.$$

Finally, the output layer applies a linear transformation to produce the output vector:

$$g_1(\alpha; \omega) = W_5^1 h_4 + b_5^1, \quad W_5^1 \in \mathbb{R}^{n \times 2n}, \quad b_5^1 \in \mathbb{R}^n.$$

The complete forward pass through the network can be expressed as:

$$g_1(\alpha;\omega_1) = W_5^1 \cdot \tanh\left(W_4^1 \cdot \tanh\left(W_3^1 \cdot \tanh\left(W_2^1 \cdot \tanh\left(W_1^1\alpha + b_1^1\right) + b_2^1\right) + b_3^1\right) + b_4^1\right) + b_5^1.$$

Here, $\omega_1 = \{W_1^1, b_1^1, W_2^1, b_2^1, \ldots, W_5^1, b_5^1\}$ represents the set of all trainable parameters of the network g_1.

B2 Structure of g_2

The neural network g_2 maps a vector $(\alpha^v, ||v||_\infty) \in \mathbb{R}^{n_v+1}$ to a positive scalar output $\Delta t \in \mathbb{R}^+$. The network is designed to learn the time step, with the output data pre-processed by taking the base-10 logarithm of the time step to address the multiscale nature of the problem. After training, the network's output is transformed back by applying the exponential function.

In the first hidden layer, the input undergoes a linear transformation followed by a nonlinear activation function:

$$h_1^{(2)} = \tanh(W_1^{(2)}\left(\alpha^v, ||v||_\infty\right) + b_1^{(2)}), \quad W_1^{(2)} \in \mathbb{R}^{2n_v\times(n_v+1)}, \quad b_1^{(2)} \in \mathbb{R}^{2n_v}.$$

The second hidden layer applies another linear transformation and activation function to the output of the first layer:

$$h_2^{(2)} = \tanh(W_2^{(2)}h_1^{(2)} + b_2^{(2)}), \quad W_2^{(2)} \in \mathbb{R}^{4n_v\times 2n_v}, \quad b_2^{(2)} \in \mathbb{R}^{4n_v}.$$

Similarly, the third hidden layer maps its input to the same dimensionality as the previous layer:

$$h_3^{(2)} = \tanh(W_3^{(2)}h_2^{(2)} + b_3^{(2)}), \quad W_3^{(2)} \in \mathbb{R}^{4n_v\times 4n_v}, \quad b_3^{(2)} \in \mathbb{R}^{4n_v}.$$

The fourth hidden layer reduces the dimensionality of its input:

$$h_4^{(2)} = \tanh(W_4^{(2)}h_3^{(2)} + b_4^{(2)}), \quad W_4^{(2)} \in \mathbb{R}^{2n_v\times 4n_v}, \quad b_4^{(2)} \in \mathbb{R}^{2n_v}.$$

Finally, the output layer applies a linear transformation to produce the output vector:

$$\log_{10}(g_2(\alpha, ||v||_\infty; \omega_2)) = W_5^{(2)}h_4^{(2)} + b_5^{(2)}, \quad W_5^{(2)} \in \mathbb{R}^{1\times 2n_v}, \quad b_5^{(2)} \in \mathbb{R}.$$

The complete forward pass through the network can be expressed as:

$$\log_{10}(g_2(\alpha, ||v||_\infty; \omega_2)) = W_5^{(2)} \cdot \tanh\left(W_4^{(2)} \cdot \tanh\left(W_3^{(2)} \cdot \tanh\left(W_2^{(2)} \cdot \tanh\left(W_1^{(2)}(\alpha, ||v||_\infty\right) + b_1^{(2)}\right) + b_2^{(2)}\right) + b_3^{(2)}\right) + b_4^{(2)}\right) + b_5^{(2)}.$$

Here, $\omega_2 = \{W_1^{(2)}, b_1^{(2)}, W_2^{(2)}, b_2^{(2)}, \ldots, W_5^{(2)}, b_5^{(2)}\}$ represents the set of all trainable parameters of the network g_2.