

Supplementary materials: Recovering complex ecological dynamics from time series using state-space universal dynamic equations

Supplementary methods 1: Three-species food chain simulation example

Simulation model parameters and procedure and data processing: We created simulated datasets from the stochastic three-species food chain model by approximating solutions to main text Eqs. 14-16 using the SRIW1 algorithm implemented in DifferentialEquations.jl. We then log-transformed and standardized the time series of each variable sampled at intervals of length Δt and added a normally-distributed noise term to each time and state variable combination. This process yielded time series of observations \hat{x}_t , \hat{y}_t and \hat{z}_t corresponding to each population series

$$S1.1) \quad \hat{x}_t = \frac{\log(x(t)) - \frac{\Delta t}{T} \sum_{\tau \in \{0:\Delta t:T\}} \log(x(\tau))}{\sqrt{\frac{\Delta t}{T} \sum_{\tau \in \{0:\Delta t:T\}} \left(\log(x(\tau)) - \frac{\Delta t}{T} \sum_{i \in \{0:\Delta t:T\}} \log(x(i)) \right)^2}} + \epsilon_{x,t}$$

$$S1.2) \quad \hat{y}_t = \frac{\log(y(t)) - \frac{\Delta t}{T} \sum_{\tau \in \{0:\Delta t:T\}} \log(y(\tau))}{\sqrt{\frac{\Delta t}{T} \sum_{\tau \in \{0:\Delta t:T\}} \left(\log(y(\tau)) - \frac{\Delta t}{T} \sum_{i \in \{0:\Delta t:T\}} \log(y(i)) \right)^2}} + \epsilon_{y,t}$$

$$S1.3) \quad \hat{z}_t = \frac{\log(z(t)) - \frac{\Delta t}{T} \sum_{\tau \in \{0:\Delta t:T\}} \log(z(\tau))}{\sqrt{\frac{\Delta t}{T} \sum_{\tau \in \{0:\Delta t:T\}} \left(\log(z(\tau)) - \frac{\Delta t}{T} \sum_{i \in \{0:\Delta t:T\}} \log(z(i)) \right)^2}} + \epsilon_{z,t}$$

$$S1.4) \quad \epsilon_{j,t} \sim N(0, \sigma_\epsilon^2).$$

We chose the model parameters that would produce chaotic dynamics in the absence of stochasticity based on the findings of Hastings and Powell (1991). These parameter values are listed in Table S1.1. We experimented with different values of process and observation errors as indicated in the table.

Supplemantray Table 1: Food chain model parameters

Parameter	Interpretation	Value
a_1	Attack rate of species y on species x	5.0
a_2	Attack rate of species z on species y	0.1
b_1	Functional response curvature x - y	2.75
b_2	Functional response curvature y - z	2.0
d_1	Mortality rate species y	0.4
d_2	Mortality rate species z	0.01
σ_x	Standard deviation of white noise term added to species x	(0.025, 0.1)
σ_ϵ	Observation errors	(0.05, 0.25)
T	Time series length	333.3
dt	Sampling interval	3.333

Selecting hyperparameters: Observation errors for the marginal-likelihood training procedures were set to match the observation errors used to simulate the datasets. The values of observation and process errors for the joint-likelihood were chosen by training the models on a small number of testing sets and checking that the state estimates produced a reasonable approximation of the trends in the data using visual diagnostics. We applied a similar approach to select the order of the smoothing penalty used in the gradient matching algorithm. To limit the computational expense of the simulation tests, we set the regularization weight to 0 for all models. The hyperparameter values for each training routine are given in Table S1.2.

Supplementary table 2: The hyperparameters used for each NODE model of the simulated food chain datasets.

<i>Model</i>	<i>Training method</i>	<i>Obs. error</i>	<i>Proc. error</i>	<i>Reg .</i>	<i>d*</i>
<i>Discrete time</i>	Marginal-likelihood	0.005, 0.0625**	NA	0	NA
<i>Discrete time</i>	Joint-likelihood	0.005, 0.065	0.025	0	NA
<i>Continuous time</i>	Marginal-likelihood	0.005, 0.0625	NA	0	NA
<i>Continuous time</i>	Joint-likelihood	0.05	0.00825	0	NA
<i>Continuous time</i>	Gradient matching	NA	NA	0	12
<i>Continuous time</i>	Shooting	NA	NA	0	NA

*order of the penalty term in the smoothing algorithm used in the gradient matching algorithm.

** value selected to match simulated dataset

*Alternative models for three-species food chain data*_We used Gaussian process empirical dynamic models implemented in the gpEDM R package ¹. The gpEDM package supports univariate time series, but can capture more complex dynamics by incorporating time lags into the model through a process called time delay embedding. We constructed a forecast for the multivariate dataset by fitting gpEDMs to the time series of each species and then averaging the forecasting ability of the three models. We set the embedding dimension of the models $E = 3$ to match the known dimensions of the system and set the length of time lags $\tau = 0$.

We compared the UDE models to linear state-space models implemented using the MARSS R package ². The equations for the process and observation models are given below, where x_t, y_t and z_t are the state variables, \hat{x}_t, \hat{y}_t and \hat{z}_t are the observations, and the remaining parameters are estimated from the data.

Process model:

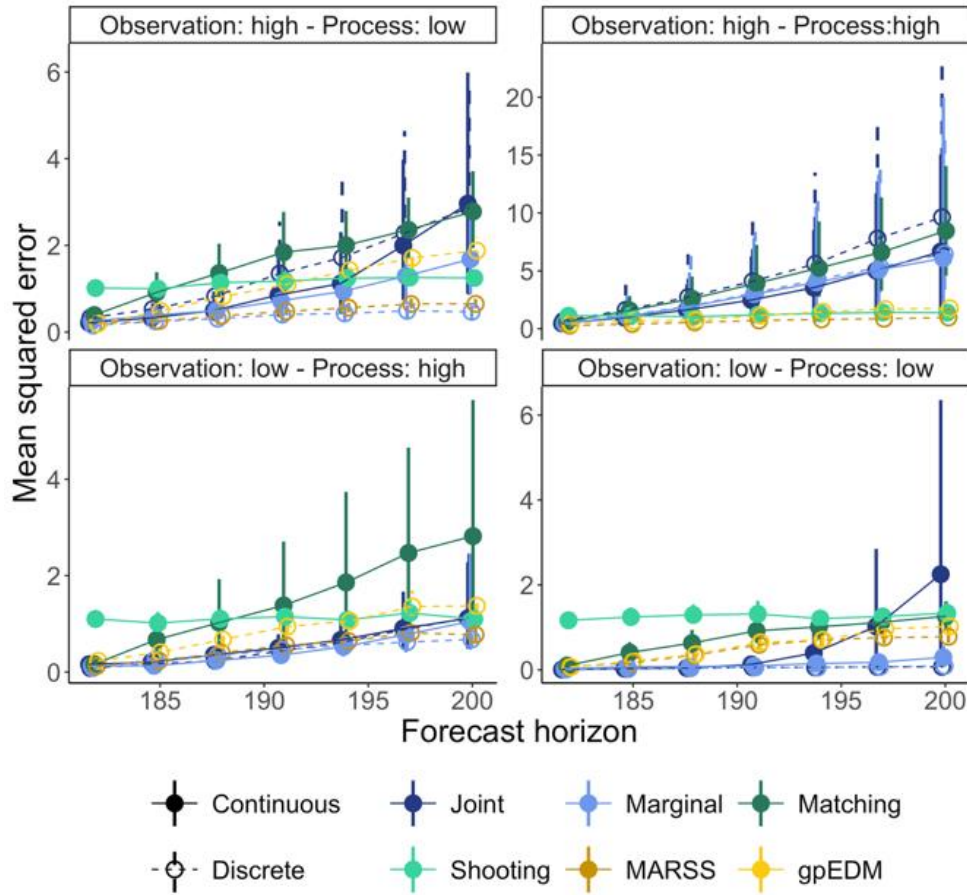
$$S1.5) \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \end{bmatrix} = \begin{bmatrix} \alpha_{x,x} & \alpha_{x,y} & \alpha_{x,z} \\ \alpha_{y,x} & \alpha_{y,y} & \alpha_{y,z} \\ \alpha_{z,x} & \alpha_{z,y} & \alpha_{z,z} \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} + \begin{bmatrix} v_{t,x} \\ v_{t,y} \\ v_{t,z} \end{bmatrix}$$

$$S1.6) \begin{bmatrix} v_{t,x} \\ v_{t,y} \\ v_{t,z} \end{bmatrix} \sim MvNormal \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} q_{x,x} & q_{x,y} & q_{x,z} \\ q_{x,y} & q_{y,y} & q_{y,z} \\ q_{x,z} & q_{y,z} & q_{z,z} \end{bmatrix} \right)$$

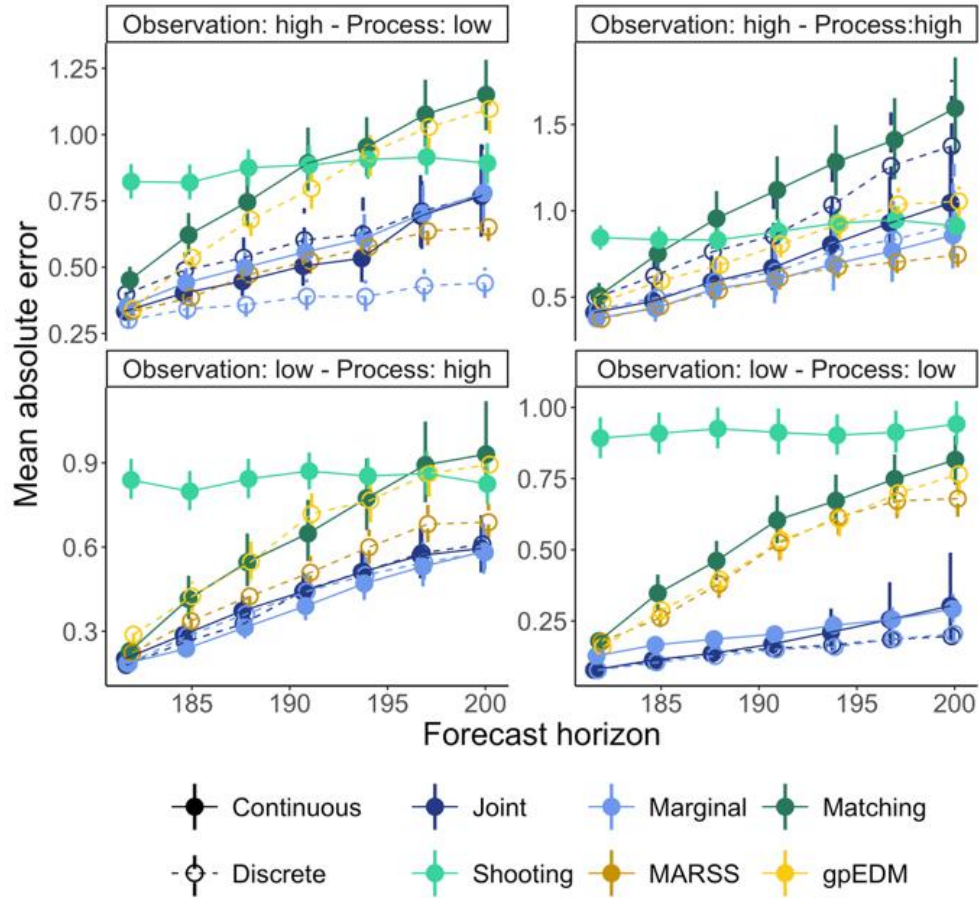
Observation model:

$$S1.7) \begin{bmatrix} \hat{x}_t \\ \hat{y}_t \\ \hat{z}_t \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} + \begin{bmatrix} \epsilon_{t,x} \\ \epsilon_{t,y} \\ \epsilon_{t,z} \end{bmatrix}$$

$$S1.8) \begin{bmatrix} \epsilon_{t,x} \\ \epsilon_{t,y} \\ \epsilon_{t,z} \end{bmatrix} \sim MvNormal \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \right)$$



Supplementary figure 1: In the main text we presented the forecasting skill of the models in terms of the mean absolute error. This weights errors evenly regardless of how large they are. Another common metric for forecasting skill is the mean squared error which gives disproportionate weight to larger errors. When observation errors are large the state-space UDE models have higher mean squared forecasting error than nearly all the alternative methods, suggesting that they are more likely to have large very large forecasting errors under these conditions.



Supplementary figure 2: In the main text we only presented results from UDE models with a continuous-time formulation. However, we also tested discrete-time UDE models. The mean absolute forecasting errors are shown for three models using dashed lines. The performance of the discrete-time UDEs models is consistent with the forecasting skill of their continuous-time counterparts.

Supplementary method 2: Kelp forest model simulation example

Simulation model parameters: We chose parameters for the kelp forest model that produced time series that flickered between periods of high and low kelp abundance. We found these parameters experimentally by simulating time series until we found parameter combinations (Table S2.1) that resulted in kelp- and urchin-dominated states roughly 50% of the time. Each time series used to train the UDE models was generated by simulating the model for a burn-in period of 150 timesteps to reduce the influence of initial conditions on the datasets, then simulated for an additional 55 years to produce the training and testing sets. We then log-transformed the resulting time series and added normally-distributed observation errors at each time point with mean 0 and variance 0.25 before using them to train the models.

Supplementary table 3: Parameters of the simulation model for the kelp abundance time series

PARAMETER	INTERPRETATION	VALUE
-----------	----------------	-------

r	Kelp per capita growth rate when rare	0.5
K	Kelp carrying capacity	1.0
α	Urchin grazing intensity	13
β	Rate of decline in urchin grazing with increasing kelp abundance	8
σ_x^2	Variance of observed environmental covariates	0.055
σ_v^2	Variance of unobserved environmental conditions	0.05
ρ	Autocorrelation of environmental conditions	0.9
σ_ϵ^2	Variance of observation errors	0.0025

Selecting hyperparameters: We used the same approach to select hyperparameters for the UDE models that we applied to the food chain model in Supplementary Appendix 1.2. The value chosen are given in Table S2.2.

Supplementary table 4: The hyperparameters used for each UDE models of the simulated kelp forest datasets.

<i>Model</i>	<i>Training method</i>	<i>Obs. error</i>	<i>Proc. error</i>	<i>Reg.</i>	<i>d*</i>
<i>UDE</i>	Marginal-likelihood	0.0025	NA	0	NA
<i>UDE</i>	Joint-likelihood	0.025	0.025	0	NA
<i>UDE fixed r</i>	Marginal-likelihood	0.0025	NA		
<i>UDE fixed r</i>	Joint-likelihood	0.025	0.025		
<i>NODE</i>	Marginal-likelihood	0.0025	NA	0	NA
<i>NODE</i>	Joint-likelihood	0.01	0.025	0	NA
<i>NODE</i>	Gradient matching	NA	NA	0	12
<i>NODE</i>	Shooting	NA	NA	0	NA

*order of the penalty term in the smoothing algorithm used in the gradient matching algorithm.

Alternative kelp urchin models: We also compared the kelp forest UDE model that included known functional forms to a fully nonparametric UDE model (NODE) that used a neural network to learn the process model:

S2.1) $F(\mathbf{u}, X) = NN(\mathbf{u}, X)$.

We compared the UDE models to linear state-space models implemented using the MARSS R package. We model changes in log kelp abundance u_t between observations as linear function of the current abundance and the observed component of environmental variation X_t , and estimated the process error τ^2 and observation error σ^2 variance terms.

Process model:

S2.2) $[u_{t+1}] = [a][u_t] + [b_0] + [b][X_t] + [v_t]$

S2.3) $[v_{t+1}] \sim MvNormal([0], [\tau])$

Observation model:

$$S2.4) [y_t] = [u_t] + [\epsilon_t]$$

$$S2.5) [\epsilon] \sim MvNormal([0], [\sigma])$$

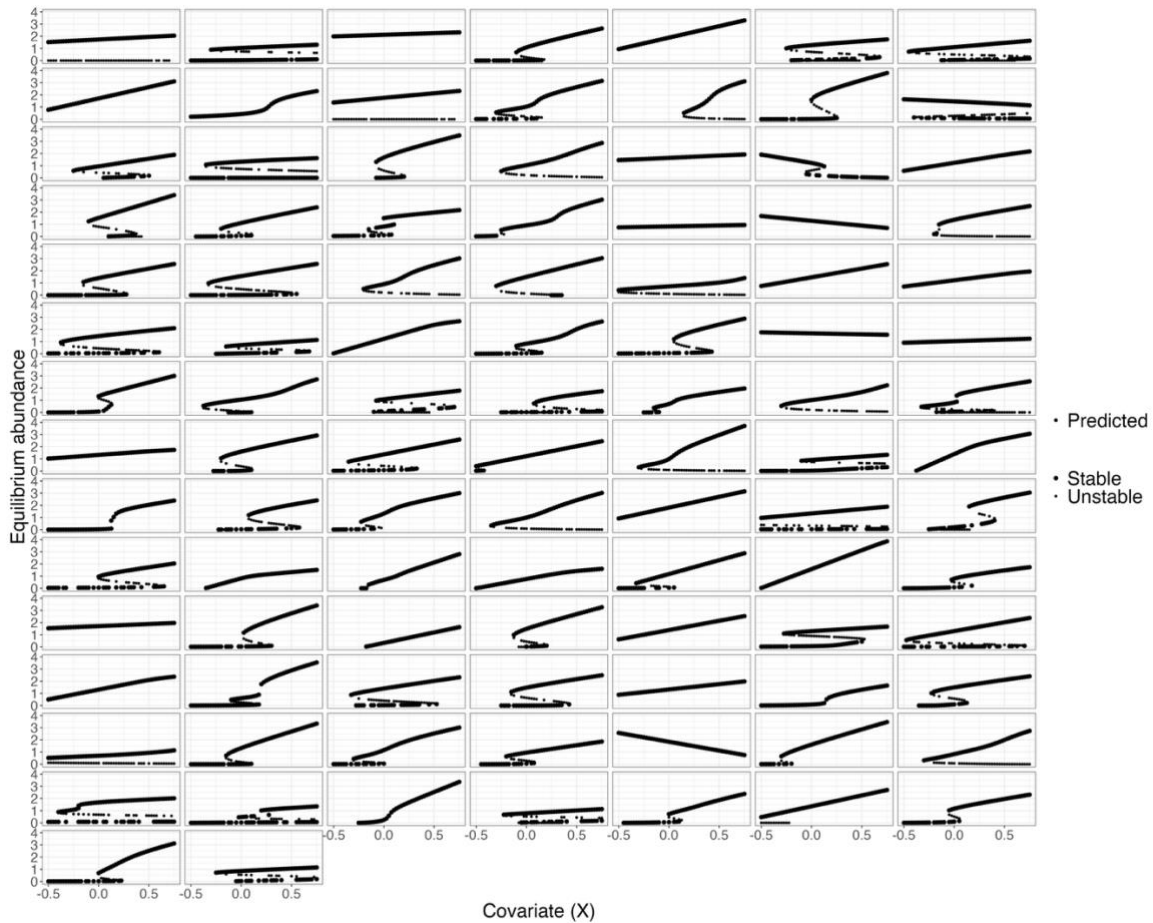
We used Gaussian process empirical dynamic models implemented in the gpEDM R package. We fit gpEDMs to the kelp time series that predict the log observed abundance y_t with the first two lagged observations $\{y_{t-1}, y_{t-2}\}$ and the observed component X_t .

Calculating bifurcation diagrams: We tested the ability of the UDE to recover ecological tipping points in the simulated time series by fitting the models to 100 simulated datasets with 50 observations. We calculated equilibrium points as a function of the covariate by solving for the zeros of the trained UDE

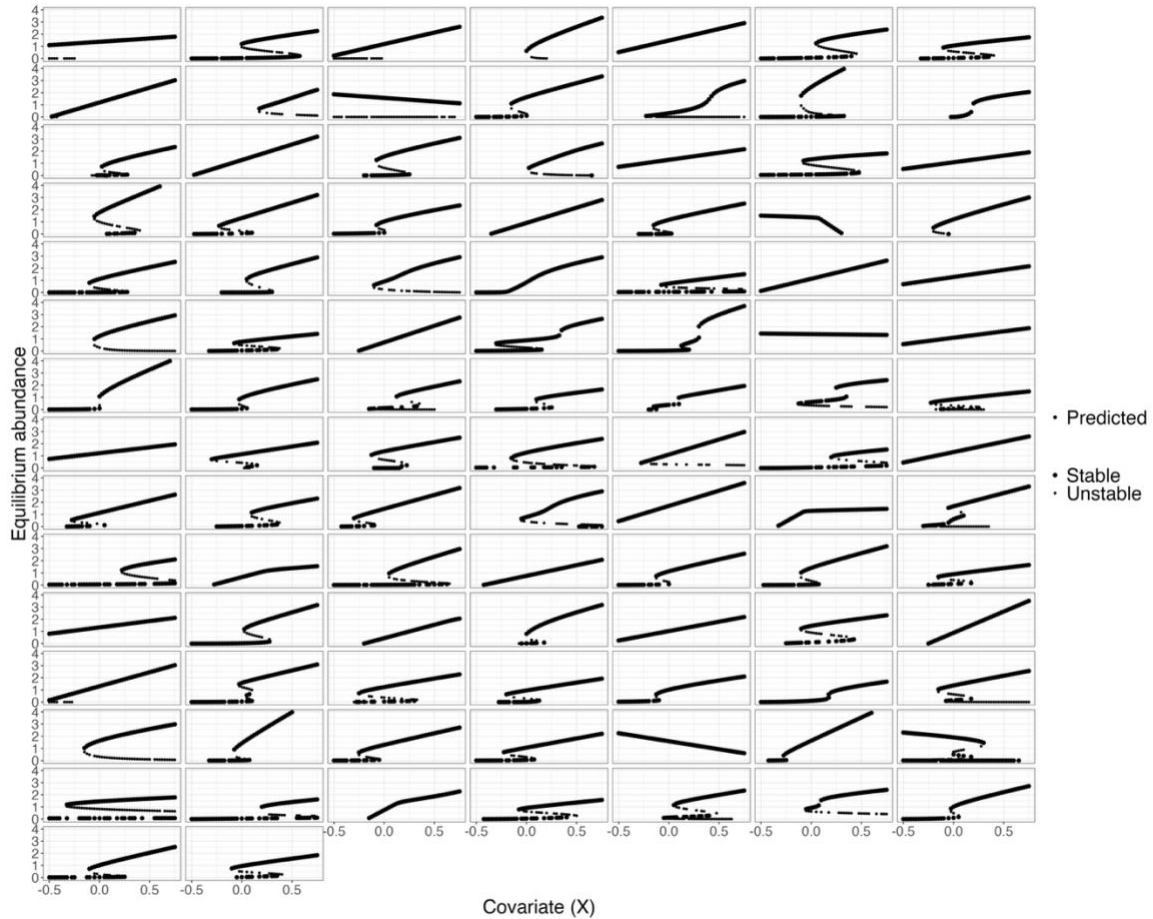
$$S2.6) 0 = F(u, X, \theta) - u.$$

We calculated the stability of the equilibrium points by evaluating the partial derivative of the model with respect to log kelp abundance around the equilibrium point $u^*(X)$. When the absolute value of the partial derivative is less than one the equilibrium is stable

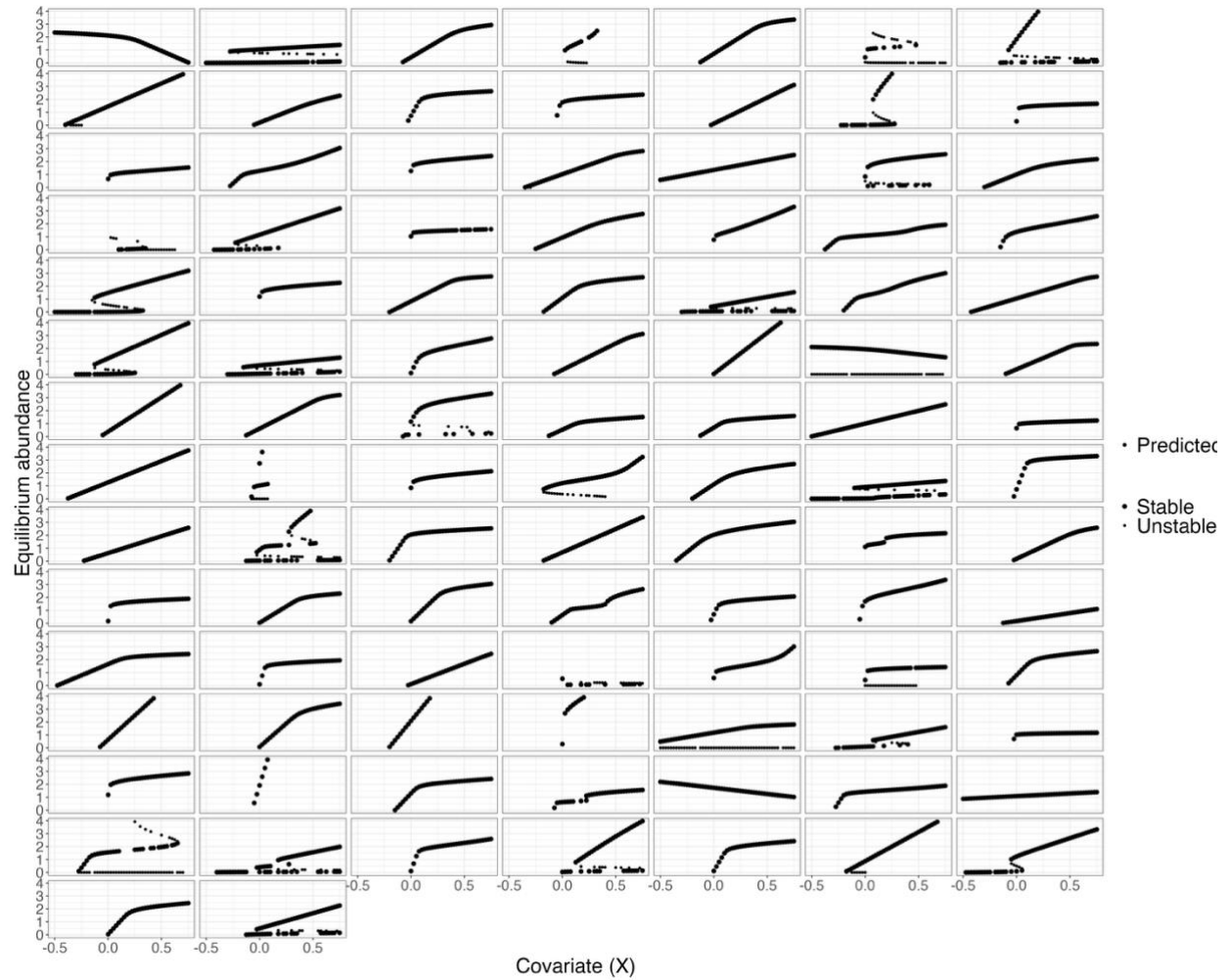
$$S2.7) \left| \frac{d}{du} F(u^*(X), X, \theta) \right| < 1.$$



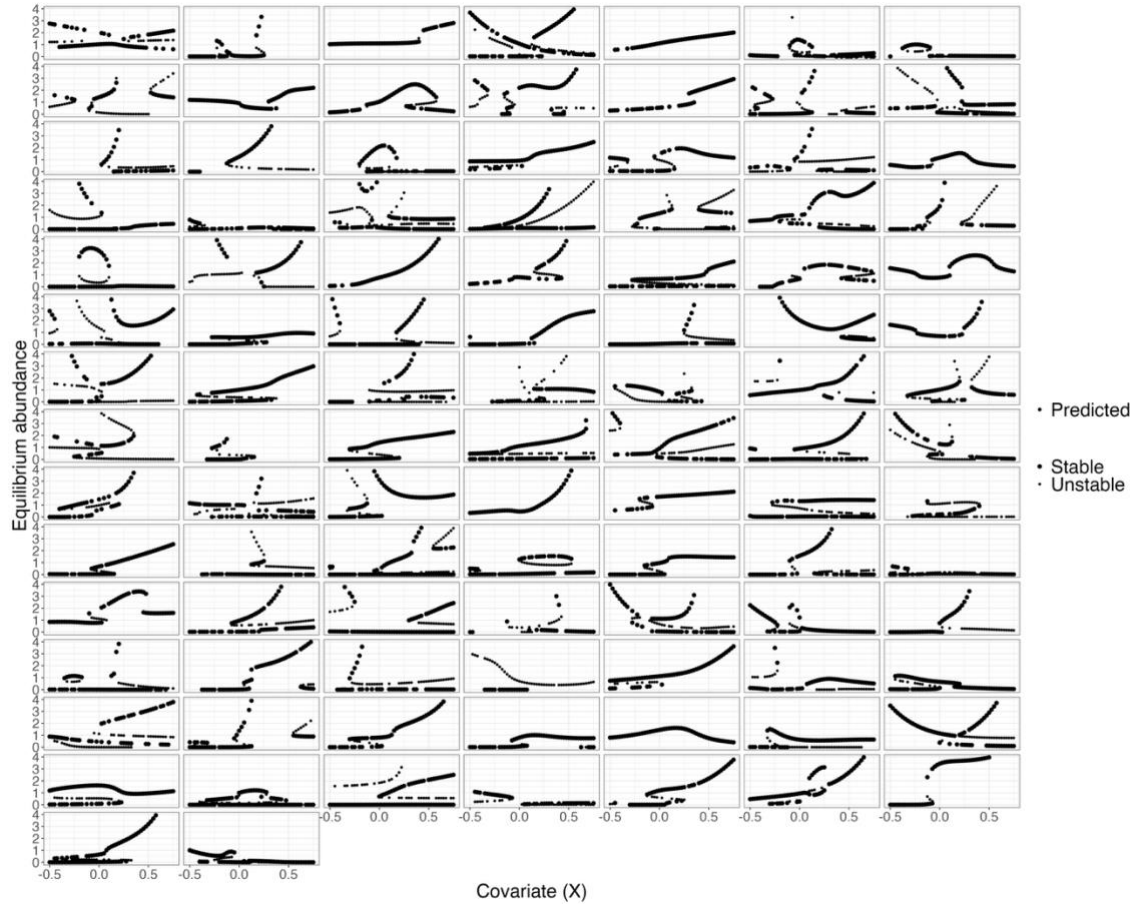
Supplementary figure 3: In main text Figure 3 we showed nine bifurcation diagrams predicted by the UDE model trained with a fixed growth rate parameter using the marginal-likelihood method. We argued based on these predictions that the model could infer the qualitative features of the kelp-urchin system's bifurcation diagram. However, nine simulations are too few to be confident in this finding. Here we present bifurcation diagrams from 100 simulations, 61 of which show a fold bifurcation as a function of the covariate. Several of the examples that do not predict a fold bifurcation pick up other qualitative features including alternative stable states at high and low kelp abundances (71 of 100), or a highly nonlinear decline in kelp abundance as a function of the covariate (65 of 100, including the simulations with fold bifurcations). These more extensive simulation tests suggest that results from the nine simulations in the main text hold when the experiment is repeated many more times.



Supplementary figure 4: In the main text and Figure S2.1 we show that models trained with the marginal-likelihood method can recover the qualitative features of the kelp-urchin model's bifurcation diagram. This figure shows the results repeating the analysis with the joint-likelihood training method. Using the joint-likelihood method, the model recovers the fold bifurcation in 71 of 100 simulations.



Supplementary figure 5: In Figures 3, S2.1, and S2.2 we show the bifurcation diagrams predicted by UDE models of the kelp-urchin system where the kelp growth rate was fixed at the true value during the training process. In this figure, we performed the analysis using the same model and training routine used in Figure S2.1, but we also estimated the kelp growth rate. When the growth rate parameter is estimated for the data along with the parameters of the neural network, the model predicted the correct qualitative features of the bifurcation diagram for a much smaller fraction of the time. The model predicted a fold bifurcation in about 24 of 100 of the simulations shown in this figure.



Supplementary figure 6: We tested the ability of the fully nonparametric NODE models to recover the qualitative features of the kelp-urchin model's bifurcation diagram, training using the marginal-likelihood method. The nonparametric model tended to predict much more complex relationships between the covariate and equilibrium kelp abundance than the true data generating process, suggesting that incorporating mechanistic information can improve UDE models' ability to infer qualitative features of a system's dynamics like bifurcation diagrams from time series data.

Supplementary methods 3: Fisheries data example

Data processing: We obtained time series of historical catch and abundance for chilipepper rockfish (*Sebastes goodei*) and cowcod (*Sebastes levis*) from the RAM Legacy Stock Assessment Database (<https://www.ramlegacy.org/>). We rescaled both datasets to span from zero to three by dividing each time series by the maximum value and multiplying by three. Transforming the data in this way ensures that each time series is weighted equally in the loss function. We calculated a coefficient $q = \max(B_t) / \max(H_t)$ to ensure that the units of harvest and biomass matched for the interaction terms included in the models.

Bioeconomic model derivation: Our dataset included observations of population biomass B and harvest H . However, it makes more mechanistic sense to describe changes in population biomass

B and fishing mortality U . To this end, we defined an ODE model in terms of changes in biomass and fishing mortality and then used the relationship between fishing mortality harvest and biomass to derive a model for harvest. We assume harvest is equal to the product of the fishing mortality rate and biomass

$$\text{S3.1)} H = UB/q.$$

Therefore, the rate of change in harvest over time can be expressed using the product rule

$$\text{S3.2)} \frac{dH}{dt} = \frac{dU}{dt} \frac{B}{q} + \frac{dB}{dt} \frac{U}{q}.$$

Substituting main text Eqs. 22 and 23 into Eq. S3.2 and simplifying yields the final model

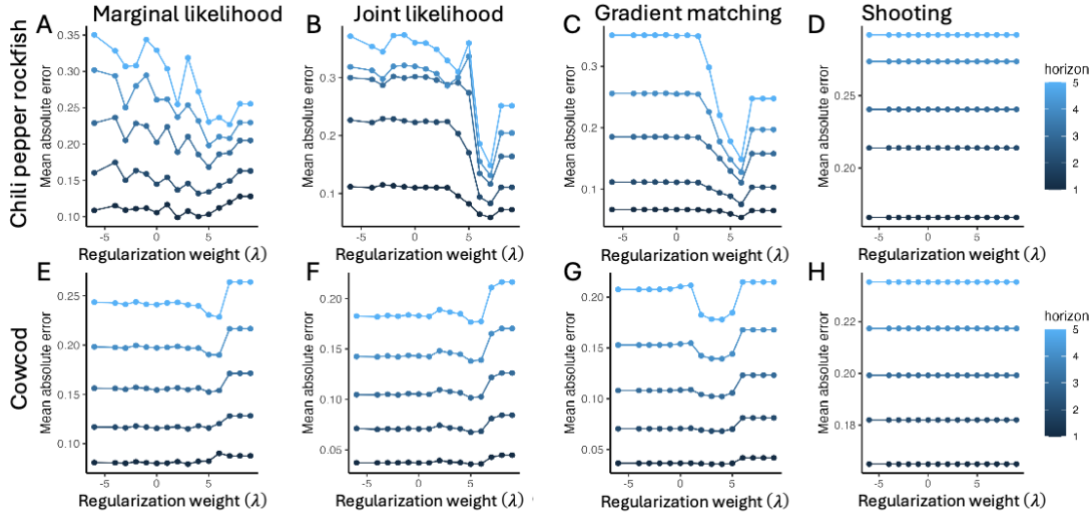
$$\begin{aligned} \text{S3.4)} \frac{dH}{dt} &= \left(U \times NN(B, I_{t>1992}) \right) \frac{B}{q} + \left(rB \left(1 - \frac{B}{K} \right) - qH \right) \frac{U}{q}, \\ &= \left(\frac{qH}{B} \times NN(B, I_{t>1992}) \right) \frac{B}{q} + \left(rB \left(1 - \frac{B}{K} \right) - qH \right) \frac{H}{B}, \\ &= NN(B, I_{t>1992})H + \left(rH \left(1 - \frac{B}{K} \right) - \frac{qH^2}{B} \right). \end{aligned}$$

Selecting hyperparameters: No prior information was available for setting the observation and process error terms for the state-space UDE models, but we found the models performed well using our initial guesses for these values. We chose regularization parameters for each model based on leave-future-out cross-validation tests (Figs. S3.1, 3.2), and the selected values are given in Table S3.1.

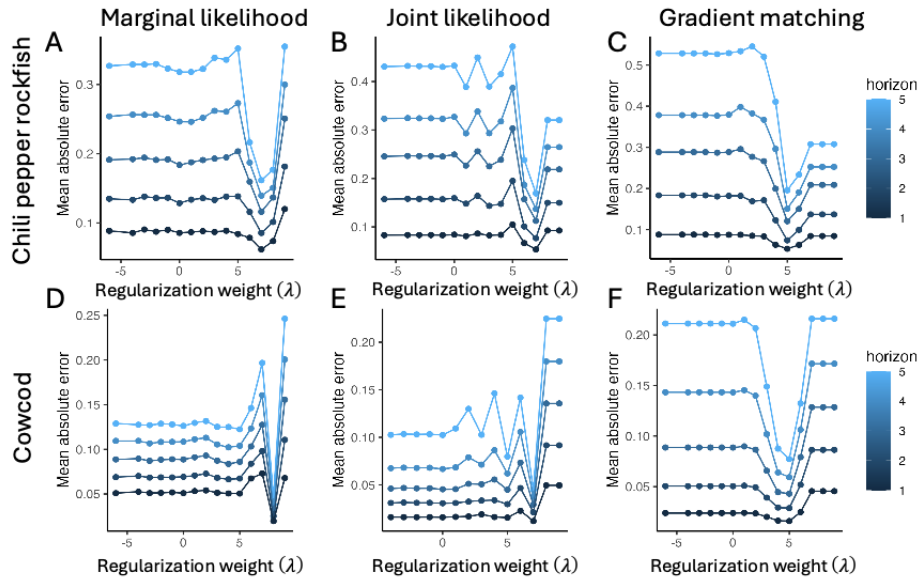
Supplementary table 5: Hyperparameter values for each of the training routines and UDE models used in the fisheries data example.

<i>Model</i>	<i>Training method</i>	<i>Obs. error</i>	<i>Proc. error</i>	<i>Reg</i>	<i>d*</i>
<i>UDE Chilipepper</i>	Marginal-likelihood	0.01	NA	1e5	NA
<i>UDE Chilipepper</i>	Joint-likelihood	0.025	0.025	1e7	NA
<i>UDE Chilipepper</i>	Gradient matching	NA	NA	1e6	12
<i>NODE Chilipepper</i>	Marginal-likelihood	0.01	NA	1e7	NA
<i>NODE Chilipepper</i>	Joint-likelihood	0.25	0.025	1e7	NA
<i>NODE Chilipepper</i>	Gradient matching	NA	NA	1e7	12
<i>NODE Chilipepper</i>	Shooting	NA	NA	1e0	NA
<i>UDE Cowcod</i>	Marginal-likelihood	0.01	NA	1e6	NA
<i>UDE Cowcod</i>	Joint-likelihood	0.025	0.025	1e7	NA
<i>UDE Cowcod</i>	Gradient matching	NA	NA	1e5	12
<i>NODE Cowcod</i>	Marginal-likelihood	0.01	NA	1e8	NA
<i>NODE Cowcod</i>	Joint-likelihood	0.025	0.025	1e6	NA
<i>NODE Cowcod</i>	Gradient matching	NA	NA	1e4	12

NODE Cowcod | Shooting NA NA 1e0 NA
 *order of the penalty term in the smoothing algorithm used in the gradient matching algorithm.



Supplementary figure 7: Selecting the regularization parameter for the fisheries NODE models. The panels show the estimated forecasting skill of NODE models trained with each of the four training procedures (columns) on each of the two datasets (rows) at one through five year forecasting horizons (colors). The x-axis is the log of the regularization weight used in the training process. We selected the regularization weight that produced the lowest forecasting errors across the five time horizons for further analysis.



Supplementary figure 8: Selecting the regularization parameter for the fisheries UDE models. The panels show the estimated forecasting skill of UDE models trained with each of

the three training procedures (the shooting method failed for this model) on each of the two datasets (rows) at one through five-year forecasting horizons (colors). The x-axis is the log of the regularization weight used in the training process. We selected the regularization weight that produced the lowest forecasting errors across the five time-horizons for further analysis. We did not use the shooting training procedure in this case because the ODE solvers were not sufficiently stable to simulate the dynamics of the model across the full-time horizon.

Alternative model details: We used the same procedure to construct Gaussian process EDM models for the fisheries time series as we used for the three-species food chain example, except we set the embedding dimension $E = 5$, because the true embedding dimension is unknown and gpEDMs use automatic relevance determination to prevent overfitting if too many time lags are included.

We compared the UDE models to linear state-space models implemented using the MARSS R package. The equations for the process and observation models are given below, where B_t and H_t are the state variables, \hat{B}_t and \hat{H}_t are the observations, $I_{t>1992}$ is zero before 1992 and one after, and the remaining parameters are estimated from the data.

Process model:

$$\text{S3.5) } \begin{bmatrix} B_{t+1} \\ H_{t+1} \end{bmatrix} = \begin{bmatrix} \alpha_{B,B} & \alpha_{B,H} \\ \alpha_{H,B} & \alpha_{H,H} \end{bmatrix} \begin{bmatrix} B_t \\ H_t \end{bmatrix} + \begin{bmatrix} b_B \\ b_H \end{bmatrix} + \begin{bmatrix} b_B \\ b_H \end{bmatrix} [I_{t>1992}] + \begin{bmatrix} v_{t,B} \\ v_{t,H} \end{bmatrix}$$

$$\text{S3.6) } \begin{bmatrix} v_{t,B} \\ v_{t,H} \end{bmatrix} \sim \text{MvNormal} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} q_{B,B} & q_{B,H} \\ q_{B,H} & q_{H,H} \end{bmatrix} \right)$$

Observation model:

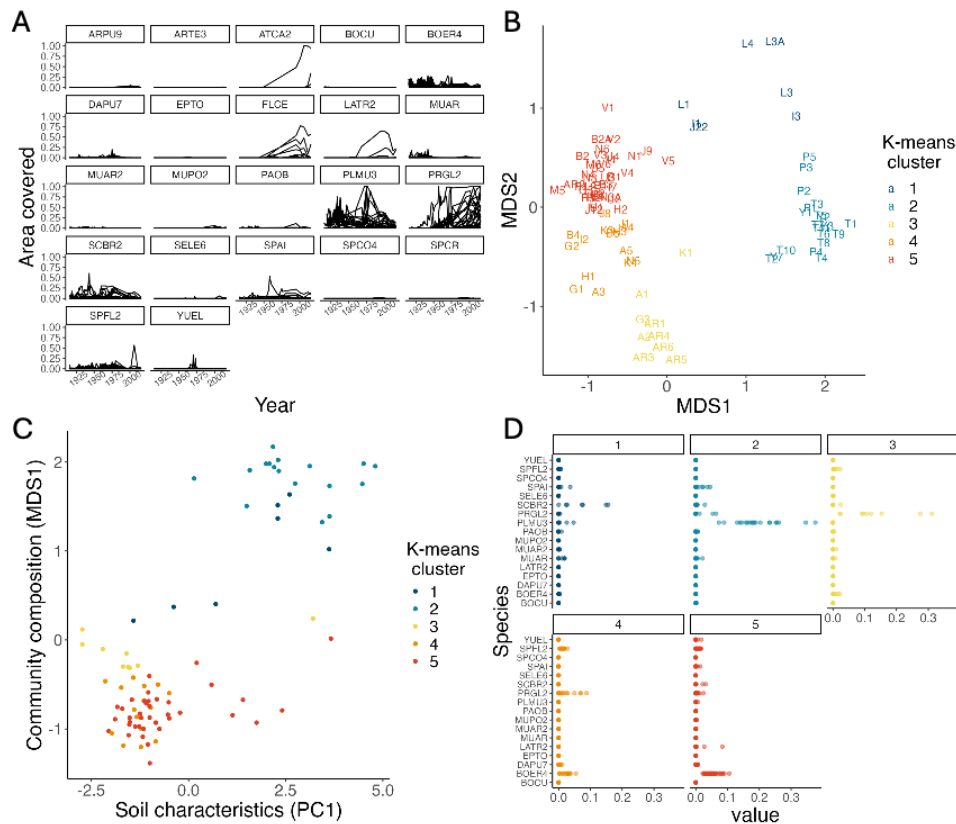
$$\text{S3.7) } \begin{bmatrix} \hat{B}_t \\ \hat{H}_t \end{bmatrix} = \begin{bmatrix} B_t \\ H_t \end{bmatrix} + \begin{bmatrix} \epsilon_{t,B} \\ \epsilon_{t,H} \end{bmatrix}$$

$$\text{S3.8) } \begin{bmatrix} \epsilon_{t,B} \\ \epsilon_{t,H} \end{bmatrix} \sim \text{MvNormal} \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \right)$$

Supplementary methods 4: Jornada Experimental Range example

Data selection: The Jornada Experimental Range long-term quadrat dataset spans several distinct plant communities, which were largely determined by soil characteristics at the quadrat site (Fig. S4.1). We categorized the quadrats by first projecting the community composition data into a lower-dimensional space using non-metric multidimensional scaling (NMDS) with the Bray-Curtis dissimilarity metric. The resulting NMDS fit was a good representation of data with a stress of 0.138. We grouped the quadrats based on their NMDS scores using the k -means clustering algorithm (Fig. S4.1B). Both NMDS scores and the k -means clusters were predicted by soil texture characteristics (Fig. S4.1C), with the most important predictors being the percents of sand, silt, clay and very fine sand (Table S4.1). Community composition in clusters 4 and 5 (more negative in NMDS dimension 1) was associated with soils that had coarser textures (higher percent sand, lower percent clay, silt, and very fine sand). These quadrats tended to have

high abundances of Black grama (*Bouteloua eriopoda*) at the beginning of the time series and were dominated by honey mesquite (*Prosopis glandulosa*) and other grass species after the regime shift in the 1950s (Fig. S4.1A). We chose to focus our analysis on this group of quadrats, which corresponded to *k*-means clusters 3, 4, and 5. These clusters were primarily differentiated by the abundances of honey mesquite and mesa dropseed (*Sporobolus flexuosus*) (Fig. S4.2, PRGLS2 and SPFL2).

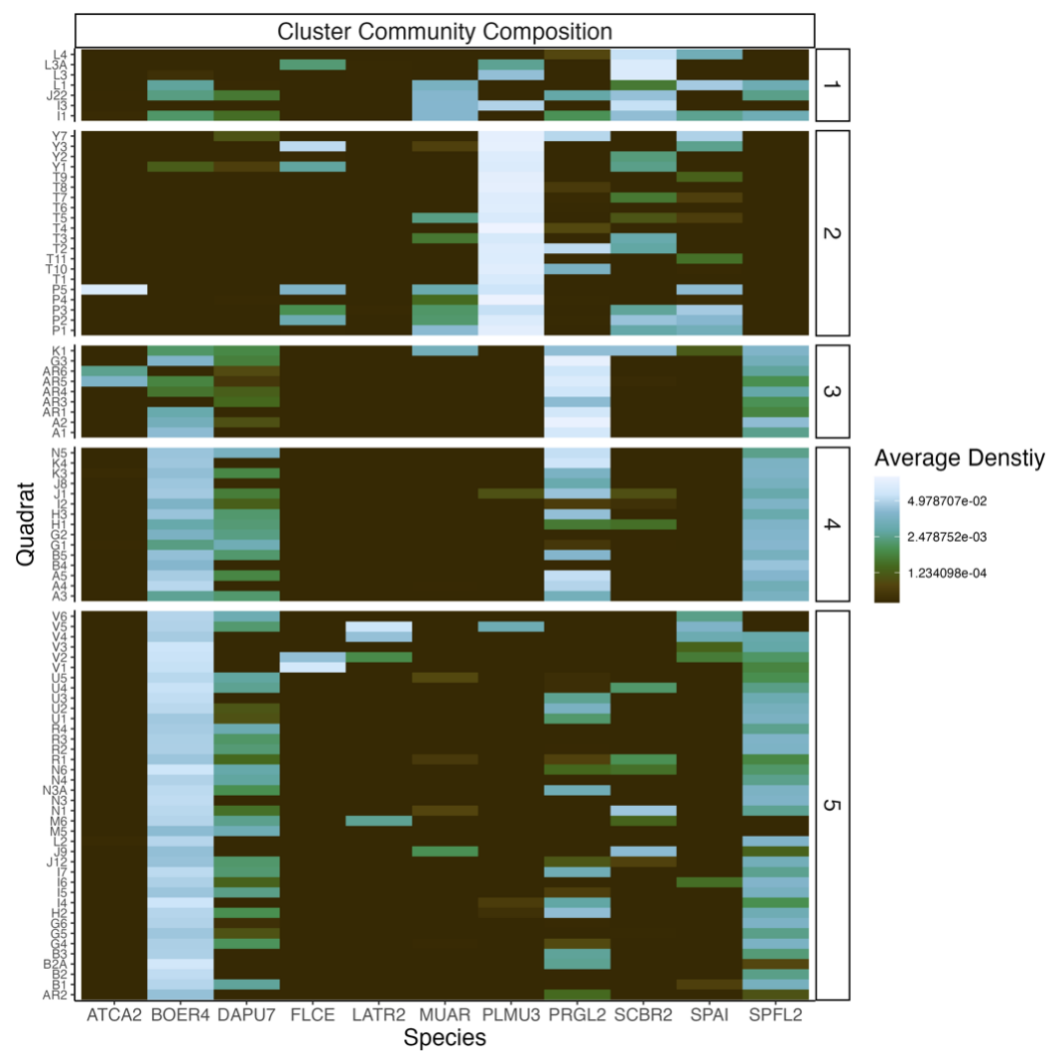


Supplemental figure 9: Jornada Experimental Range plant community composition quadrat data. A) Raw time series data of the density of each species (facets) in each quadrat (lines). B) An NMDS plot of the average community composition in each quadrat from 1915 to 2016. C) Correlation between community composition (NMDS dimension 1) and soil characteristics (principal component 1). D) Average species composition in each quadrat, grouped by *k*-means algorithm into five clusters. See Table S4.2 for definitions of the species abbreviations.

Supplementary table 6: Species abbreviations, growth form, scientific and common names.

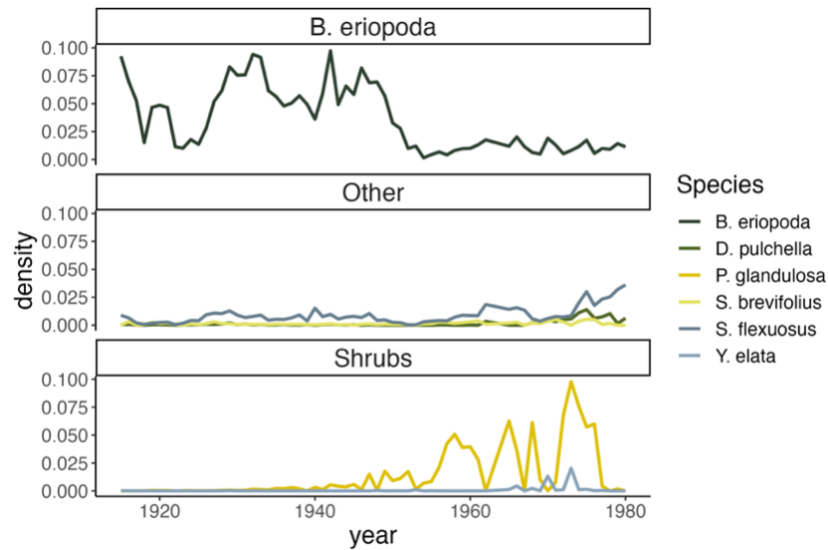
Abbreviation	Genus	Species	Form	Common Name
SCBR2	<i>Scleropogon</i>	<i>brevifolius</i>	GRASS	burrograss
PLMU3	<i>Pleuraphis</i>	<i>mutica</i>	GRASS	tobosa grass
BOER4	<i>Bouteloua</i>	<i>eriopoda</i>	GRASS	black grama

SPFL2	<i>Sporobolus</i>	<i>flexuosus</i>	GRASS	mesa dropseed
MUAR	<i>Muhlenbergia</i>	<i>arenacea</i>	GRASS	ear muhly
SPAI	<i>Sporobolus</i>	<i>airoides</i>	GRASS	alkali sacaton
DAPU7	<i>Dasyochloa</i>	<i>pulchella</i>	GRASS	low woollygrass
PRGL2	<i>Prosopis</i>	<i>glandulosa</i>	SHRUB	honey mesquite
YUEL	<i>Yucca</i>	<i>elata</i>	SHRUB	soaptree yucca
LATR2	<i>Larrea</i>	<i>tridentata</i>	SHRUB	creosote bush



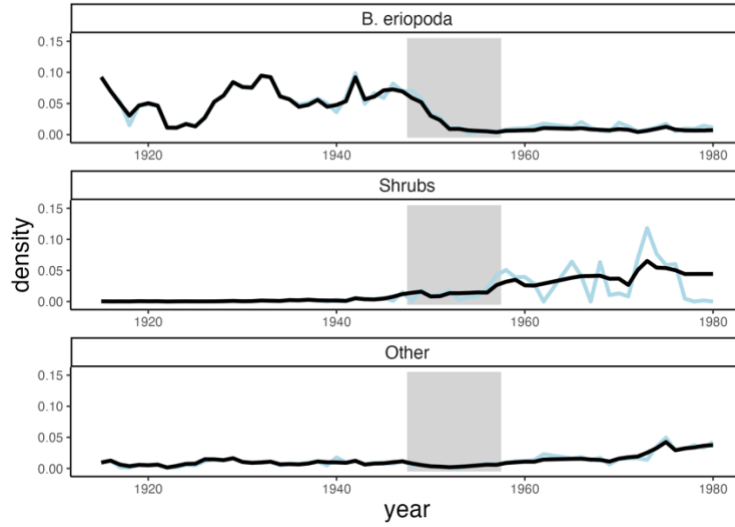
Supplementary figure 10: Average densities of the 11 most common species in each quadrat over the study period. The quadrats are grouped into clusters identified by applying the k-means algorithm to NMDS based on community composition data. We trained the UDE models on data from quadrats grouped into clusters 4 and 5.

The shrub group was primarily composed of *Prosopis glandulosa*, but *Yucca elate* also appeared in low abundance (Fig. S4.4). The other species group was primarily comprised of three grasses: *Sporobolus flexuosus*, *Dasyochloa pulchella*, and *Scleropogon brevifolius* (Fig. S4.4). Christensen et al. (2023) categorized these three species as transient grasses, which are often present in the Jornada rangeland, but rarely comprise large fractions of the plant community in a given area.



Supplementary figure 11 Annual densities of each species within the species groups used to train the UDE models.

We trained the UDE models on the average abundances of species within three species groups across all quadrats sorted into clusters 3, 4, and 5 of our community composition analysis. Not every quadrat was observed every year. This caused the annual average species abundances to vary over time, because quadrats with high abundances of certain species were not included in the average. This effect was particularly consequential for the shrub time series. Shrub-dominated communities are inherently patchy, and as a consequence, some quadrats were almost entirely covered by shrubs while others were almost entirely bare ground. The combination of this spatial pattern and inconsistent sampling appears to cause a high degree of variability in the shrub time series. To reduce the variability caused by the sampling procedure, we calculated the average abundance of each species group in each quadrat and used a linear interpolation to impute missing values in years when a quadrat was not sampled. We found this procedure had little or no effect on the *B. eriopoda* and other species time series but smoothed the shrub time series considerably (Fig. S4.3).



Supplementary figure 12: Comparison of annual plant densities averaged over monitored quadrats (blue line) to annual plant densities averaged over all quadrats using a linear interpolation to impute missing quadrat and year combinations (black line). The gray shading is the period of severe drought.

Derivation of UDE2 the plant – soil feedback model: The model of plant-soil feedbacks for the Jornada Experimental Range dataset included two state variables that represented the quantity and distribution of soil water content: a homogeneously-distributed stock M and a heterogeneously-distributed stock H . The dynamics of each stock are governed by a differential equation that depends on the composition of the plant community \mathbf{X} , precipitation $P(t)$, and temperature $T(t)$

$$\text{S4.1) } \frac{dM}{dt} = f(\mathbf{X})P(t) - M(a_M + b_M T(t))$$

$$\text{S4.2) } \frac{dH}{dt} = (1 - f(\mathbf{X}))P(t) - H(a_H + b_H T(t)).$$

These variables are not observed directly in our dataset, and the state-space UDE framework does not currently support unobserved dynamic variables. Because of this limitation, we chose to approximate the value of these variables by assuming the dynamics of the soil water pools are fast compared to changes in temperature, precipitation, and community composition. Although we cannot confirm the validity of this assumption without data on soil moisture levels, it allows us to capture the qualitative features of plant-soil feedback without adding new state variables. Using this assumption, we set $M^*(t)$ and $H^*(t)$ at their equilibrium values, which we found by setting the right-hand side to zero and solving for M and H , respectively. This yields approximate values of M and P that are functions of the current plant community composition, precipitation, and temperature:

$$\text{S4.3) } M^*(t) = \frac{f(\mathbf{X}(t))P(t)}{(a_M + b_M T(t))}$$

$$S4.4) P^*(t) = \frac{(1-f(X(t)))P(t)}{(a_H + b_H T(t))}.$$

Selecting hyperparameters: The time series we used to train the UDE models to the Jornada rangeland data were calculated by taking means of the abundance $y_{j,i}$ each species group i in each quadrat j . We estimated the observation error by calculating the standard error of the mean abundance $\hat{y}_{i,t}$ given the quadrat level observation $y_{j,i,t}$

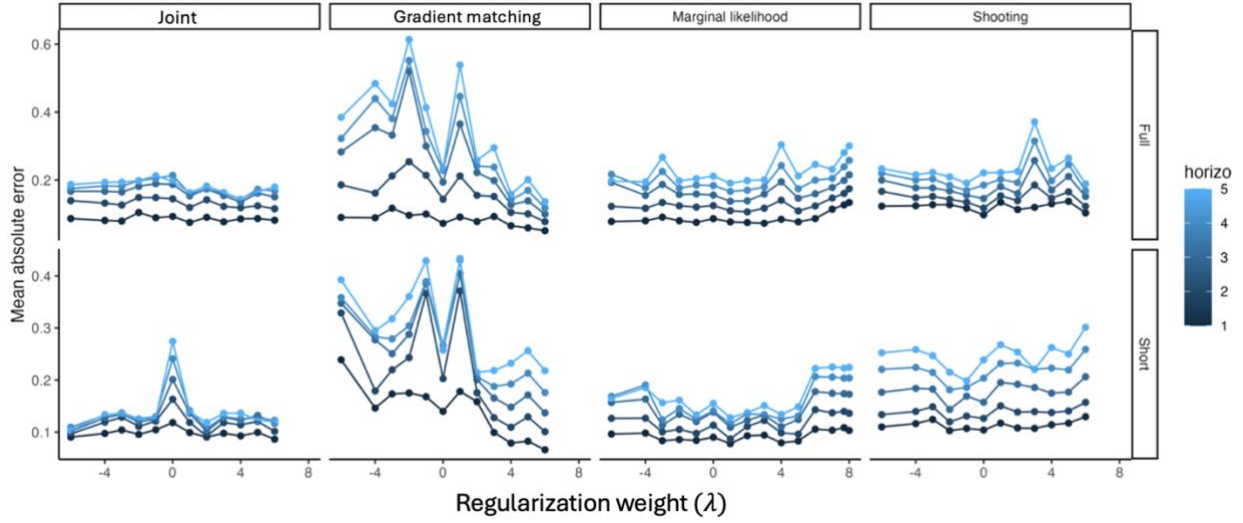
$$S4.5) \sigma_{\epsilon,i,t} = \frac{1}{\sqrt{n}} \sqrt{\sum_{i=1}^n \frac{(y_{j,i,t} - \hat{y}_{i,t})^2}{n-1}}.$$

We averaged these estimates over time to get the final point estimates of the observation errors used in our models. Following the simulated examples, we selected the process error weight for the joint-likelihood by visually inspecting the estimated states $\hat{\mathbf{u}}_t$. The regularization terms for all models were selected by leave-future-out cross-validation (Figs. S4.5, S4.6).

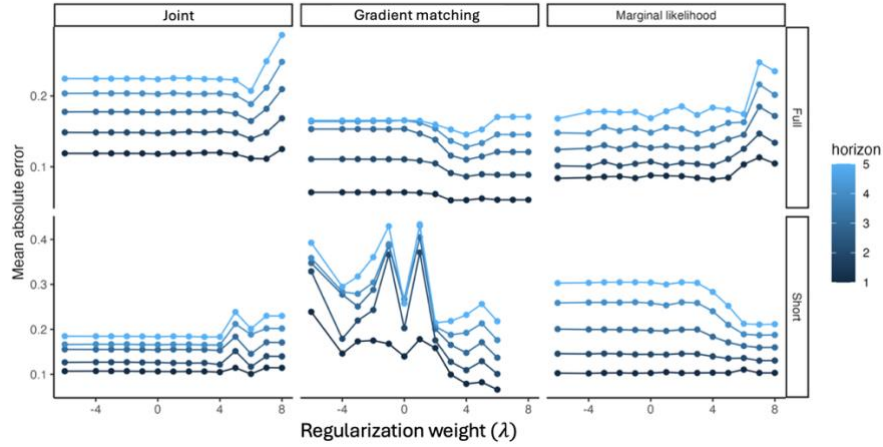
Supplementary table 7: Hyperparameters used in each of the training routines for the UDE models in the Jornada rangeland example. The column for regularization weight contains two values, one for the models trained shortened dataset and one value for models trained on the full dataset. The first value (short dataset) was used to create the bifurcation diagrams in Figure 5F and the second (full dataset) the bifurcation diagrams in Figure 5E. The shooting training routine did not converge for UDE1.

<i>Model</i>	<i>Training method</i>	<i>Obs. error</i>	<i>Proc. error</i>	<i>Reg. λ</i>	<i>d*</i>
<i>UDE1</i>	Marginal-likelihood	{0.0058,0.0309,0.0026 }	NA	1e4, 1e3	NA
<i>UDE1</i>	Joint-likelihood	{0.0058,0.0309,0.0026 }	$0.5\sigma_{\epsilon}$	1e2, 1e4	NA
<i>UDE1</i>	Gradient matching	NA	NA	1e6, 1e4	12
<i>UDE2</i>	Marginal-likelihood	{0.0058,0.0309,0.0026 }	NA	1e8, 1e4	NA
<i>UDE2</i>	Joint-likelihood	{0.0058,0.0309,0.0026 }	$0.5\sigma_{\epsilon}$	1e4, 1e6	NA
<i>UDE2</i>	Gradient matching	NA	NA	1e6, 1e4	12
<i>UDE2</i>	Shooting	NA	NA	1e-2, 1e-2	NA

*order of the penalty term in the smoothing algorithm used in the gradient matching algorithm.



Supplementary figure 13: Selecting the regularization parameter for the Jornada rangeland competition model (UDE1). The panels show the estimated forecasting skill of the UDE models trained with each of the four training procedures (columns) on each of the two datasets (rows) at one through five year forecasting horizons (colors). The x-axis is the log of the regularization weight used in the training process. We selected the regularization weight that produced the lowest forecasting errors across the five time-horizons for further analysis.



Supplementary figure 14: Selecting the regularization parameter for the Jornada rangeland plant-soil feedback model (UDE2). The panels show the estimated forecasting skill of the UDE models trained with each of the three training procedures (columns) on each of the two datasets (rows) at one through five year forecasting horizons (colors). The x-axis is the log of the regularization weight used in the training process. We selected the regularization weight that produced the lowest forecasting errors across the five-time horizons for further analysis. The shooting method failed to train this model so we did not include it in our results.

Alternative model details: We used the same procedure from testing the gpEDMs described for the previous examples. The embedding dimension was unknown in this case, so it was set to $E = 5$ to allow the model to pick up complex relationships if they occurred in the dataset. Note that

we log-transformed the data to fit the models, produced forecasts in log-space and then back-transformed the data and the forecasts to ensure we calculated the forecasting error on the same scale used for the UDE models.

We trained MARSS models on the log-transformed density data following the same back transformation procedure to estimate the forecasting skill used for the gpEDM models. The equations for the process and observation models are given below, where x_t , y_t , and z_t correspond to the log density of *B. eriopoda*, shrubs, and other species respectively. The variables \hat{x}_t , \hat{y}_t , and \hat{z}_t are the observations, P_t is precipitation, and T_t is temperature.

Process model:

$$\text{S4.6)} \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ z_{t+1} \end{bmatrix} = \begin{bmatrix} \alpha_{x,x} & \alpha_{x,y} & \alpha_{x,z} \\ \alpha_{y,x} & \alpha_{y,y} & \alpha_{y,z} \\ \alpha_{z,x} & \alpha_{z,y} & \alpha_{z,z} \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} + \begin{bmatrix} b_{x,P} & b_{x,T} \\ b_{y,P} & b_{y,T} \\ b_{z,P} & b_{z,T} \end{bmatrix} \begin{bmatrix} P_t \\ T_t \end{bmatrix} + \begin{bmatrix} v_{t,x} \\ v_{t,y} \\ v_{t,z} \end{bmatrix}$$

$$\text{S4.7)} \begin{bmatrix} v_{t,x} \\ v_{t,y} \\ v_{t,z} \end{bmatrix} \sim \text{MvNormal} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} q_{x,x} & q_{x,y} & q_{x,z} \\ q_{x,y} & q_{y,y} & q_{y,z} \\ q_{x,z} & q_{y,z} & q_{z,z} \end{bmatrix} \right)$$

Observation model:

$$\text{S4.8)} \begin{bmatrix} \hat{x}_t \\ \hat{y}_t \\ \hat{z}_t \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix} + \begin{bmatrix} \epsilon_{t,x} \\ \epsilon_{t,y} \\ \epsilon_{t,z} \end{bmatrix}$$

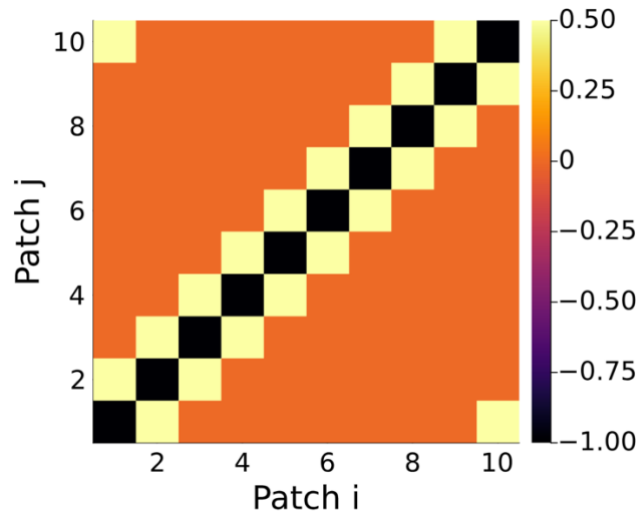
$$\text{S4.9)} \begin{bmatrix} \epsilon_{t,x} \\ \epsilon_{t,y} \\ \epsilon_{t,z} \end{bmatrix} \sim \text{MvNormal} \left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} r & 0 & 0 \\ 0 & r & 0 \\ 0 & 0 & r \end{bmatrix} \right)$$

Supplementary methods 5: Metacommunity predator-prey simulation example

Simulation model parameters: The parameter values used to simulate the multi-patch predator-prey models are given in Table S5.1. Figure S5.1 shows the connectivity matrix describing the movement of predators between patches in the model for a ten-patch model. The same matrix structure with dispersal between adjacent patches and a periodic boundary is used when simulating larger or smaller number of patches.

Supplementary table 8: model parameters for the simulated multi-patch predator prey datasets

PARAMETER	DESCRIPTION	VALUE
τ	Variance of Brownian motion	0.15
r	Logistic growth rate of prey	5.0
α	Attack rate	8.0
θ	Predator conversion efficiency	0.75
m	Predator mortality rate	0.85
h	Prey handling time	1.0



Supplementary figure 15: Matrix describing the rate of movement of individuals between patches within the spatial predator-prey model. We assume predators move between adjacent patches with a periodic boundary condition. Predators disperse to the two adjacent patches at a rate of 0.5 individuals per unit time, the entries along the diagonal are equal to negative one to capture the effect of individuals leaving.

Supplementary references

1. Munch, S. & Rogers, T. GPEDM: Gaussian Process regression for Empirical Dynamic Modeling. (2024).
2. Holmes, E., E., Ward, E., J. & Wills, K. MARSS: Multivariate Autoregressive State-space Models for Analyzing Time-series Data. *The R Journal* **4**, 11 (2012).