

Unsupervised Probing Critical Transitions in Complex Systems Using Generative Diffusion Models

Supplementary Information

Peng Zhang^{2†}, Jun Fu^{1*†}, Rolf Findeisen³

¹State Key Laboratory of Synthetical Automation for Process
Industries, Northeastern University, Shenyang, 100819, China.

²School of Future Technology, Northeastern University, Shenyang,
100819, China.

³Control and Cyber-Physical Laboratory, Technical University of
Darmstadt, Darmstadt, 64289, Germany.

*Corresponding author(s). E-mail(s): junfu@mail.neu.edu.cn;

†These authors contributed equally to this work

Supplementary Methods

Dynamical systems and datasets

Networked dynamical systems

We considered resource–biomass[1, 2], Wilson–Cowan neuronal[3] and susceptible–infected–susceptible (SIS)[4] dynamics on undirected networks. For node i in a network of n nodes, the resource–biomass dynamics were

$$dx_i = \left[rx_i \left(1 - \frac{x_i}{k} \right) - \varepsilon(t) \frac{x_i^2}{x_i^2 + 1} + d \sum_{j=1}^n A_{ij} (x_j - x_i) \right] dt + \eta dW_i(t), \quad (\text{S1})$$

where x_i is biomass, r is the intrinsic growth rate, k is the carrying capacity, d is the coupling strength and $\varepsilon(t)$ is grazing pressure. The Wilson–Cowan neuronal dynamics

were

$$dx_i = \left[-x_i + (1 - \varepsilon(t)) \sum_{j=1}^n A_{ij} \frac{1}{1 + \exp[-\tau(x_j - \mu)]} \right] dt + \eta dW_i(t), \quad (\text{S2})$$

where x_i is neuronal activity, τ and μ set the steepness and midpoint of the activation function, and $\varepsilon(t)$ controls the reduction in interaction strength. The SIS dynamics were

$$dx_i = \left[-\delta x_i + \varepsilon(t)(1 - x_i) \sum_{j=1}^n A_{ij} x_j \right] dt + \eta dW_i(t), \quad (\text{S3})$$

where x_i is the infection probability, δ is the recovery rate and $\varepsilon(t)$ is the infection rate. In all three systems, i and j index network nodes, A_{ij} is the (i, j) entry of the adjacency matrix, t denotes simulation time, $W_i(t)$ is an independent Wiener process associated with node i , and η controls the intensity of the stochastic perturbation.

Barabási–Albert (BA)[5], Erdős–Rényi (ER)[6] and Watts–Strogatz (WS)[7] networks contained 30, 50, 70 or 100 nodes. Ten connected realizations were generated for each topology and network size. The topology parameters were sampled over the ranges in Supplementary Table 1. Each trajectory began with a burn-in of 100 simulation-time units at the initial control-parameter value. The control parameter then changed linearly over the ranges listed in Supplementary Table 1. Equations (S1)–(S3) were integrated using the Euler method with a time step of 0.1. The network-averaged state was calculated after simulation to display macroscopic dynamics and locate the realized transition.

Controlled SLBP system

The shallow-lake bream–pike (SLBP) system[8] was described by

$$dX = \left[i_b + rX \frac{N(t)}{N(t) + H_1} - c_b X^2 - p_r F_R(X) Y \right] dt + \sqrt{2D} dW_1(t), \quad (\text{S4})$$

$$dY = \left[i_p + c_e p_r F_R(X) Y \frac{V(X)}{V(X) + H_2} - m_p Y - c_p Y^2 \right] dt + \sqrt{0.2D} dW_2(t), \quad (\text{S5})$$

with

$$F_R(X) = \frac{X^2}{X^2 + H_4^2}, \quad V(X) = \frac{KH_3^2}{X^2 + H_3^2}. \quad (\text{S6})$$

Here, X and Y are the bream and pike population densities, $N(t)$ is nutrient availability, and D is the environmental noise intensity. The processes $W_1(t)$ and $W_2(t)$ are independent Wiener processes. The parameters appearing in the dynamical equations are defined and listed in Supplementary Table 2.

The nutrient level defines a low-state single-attractor regime for $N < 1.34$, a bistable regime for $1.34 < N < 3.04$ and a high-state single-attractor regime for $N > 3.04$. For time-dependent trajectories, $N(t)$ increased from 0 to 3.5 or decreased from 3.5 to 0. The slow and rapid changes had durations $T = 10^6$ and $T = 5 \times 10^4$,

respectively. The equations were integrated using an adaptive Milstein solver on a time grid with spacing $\Delta t = 0.1$, following a burn-in period of 1,000 simulation-time units. Stationary training trajectories were also generated at fixed $N = 0.5, 1, 2.5, 3$ and 5. Training-noise controls used $D = 10^{-3}, 10^{-4}$ and 10^{-5} , while the single-regime comparison used $N = 0.5, 2.5$ and 5 with $D = 10^{-3}$.

Empirical data and forecasting samples

The empirical analysis used 12 Mediterranean Sea molybdenum (Mo) time series from the MS21, MS66 and 64PE cores, obtained from the dataset of Hennekam et al.[9, 10]. The 12 core-specific datasets are listed in Supplementary Table 4. Ages expressed in thousands of years before present (ka BP) were converted to a forward-running relative time axis before sliding-window evaluation. Reference transition locations were used only for post hoc comparison. NsDiff was trained on a balanced subset of simulated trajectories from the fold, Hopf, transcritical and neutral categories in the dataset released by Bury et al.[11, 12]. Category labels were used only for balanced sampling and were not provided as model inputs or prediction targets. Transition locations and control-parameter information were not used during training or inference.

Let z_t denote the observed scalar or node-state vector at sampling time t . The observation window X_t and corresponding future sequence Y_t were defined as

$$X_t = (z_{t-L+1}, \dots, z_t), \quad Y_t = (z_{t+1}, \dots, z_{t+H}), \quad (\text{S7})$$

where L is the observation-window length and H is the prediction horizon. Only samples with complete observation and future windows were retained. For each forecasting pair (X_t, Y_t) , Y_0 denotes the future sequence Y_t at diffusion step zero, before diffusion noise is added. In this forecasting notation, X_t and Y_t denote time-series windows and are distinct from the SLBP state variables X and Y .

Model formulations and optimization objectives

The four models generated future trajectories conditioned on an observation window. We denote the observation window by X and its corresponding clean future sequence by Y_0 . At diffusion step $s \in \{1, \dots, S\}$, the diffused future sequence is denoted by Y_s , where S is the total number of diffusion steps. For DiffSTG and DiffusionTS, the observation and future sequences are concatenated as $Z_0 = [X, Y_0]$, and Z_s denotes the diffused concatenated sequence. The diffusion index s is distinguished from the sampling time t . We define $\beta_s \in (0, 1)$ as the variance schedule, $\alpha_s = 1 - \beta_s$, and $\bar{\alpha}_s = \prod_{u=1}^s \alpha_u$. Throughout this section, I is the identity matrix with the same dimensions as the diffused sequence, θ denotes trainable diffusion-model parameters, and $\|\cdot\|_F$ denotes the Frobenius norm. The model-specific formulations follow the original architectures and the implementations used in this study.

DiffSTG

DiffSTG[13] operates on the complete spatiotemporal sequence $Z_0 = [X, Y_0]$. A binary mask M retains the observed history and masks the future, giving $Z_{\text{msk}} = M \odot Z_0$.

At diffusion step s , the complete sequence is perturbed as

$$Z_s = \sqrt{\alpha_s} Z_0 + \sqrt{1 - \alpha_s} \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \quad (\text{S8})$$

UGnet takes Z_s , Z_{msk} , s and G as inputs, where G denotes the graph structure. Its output, ϵ_θ , estimates the added noise ϵ . It combines temporal downsampling and upsampling with residual gated graph convolutions. The practical training objective was

$$\mathcal{L}_{\text{DiffSTG}} = \mathbb{E}_{Z_0, s, \epsilon} \left[\|\epsilon - \epsilon_\theta(Z_s, s \mid Z_{\text{msk}}, G)\|_F^2 \right]. \quad (\text{S9})$$

Thus, the observation window and adjacency matrix condition every denoising step, while the full future sequence is generated jointly.

DiffusionTS

DiffusionTS[14] also uses the standard Gaussian endpoint but predicts the clean sequence rather than only the added noise. The training sequence was $Z_0 = [X, Y_0]$, and Z_s denotes its diffused form at diffusion step s . Its future segment corresponds to the conditional clean-future reconstruction used in the main text:

$$\begin{aligned} \hat{Z}_0(Z_s, s) &= V_{\text{trend}} + \sum_{\ell=1}^{N_{\text{dec}}} S_\ell + R, \\ \hat{Y}_0(Z_s, s) &= \Pi_Y \hat{Z}_0(Z_s, s) \\ &= \Pi_Y V_{\text{trend}} + \sum_{\ell=1}^{N_{\text{dec}}} \Pi_Y S_\ell + \Pi_Y R \\ &\equiv V_{\text{trend}}^{(Y)} + \sum_{\ell=1}^{N_{\text{dec}}} S_\ell^{(Y)} + R^{(Y)}, \end{aligned} \quad (\text{S10})$$

where Π_Y extracts the future segment, N_{dec} is the number of decoder blocks, and V_{trend} , S_ℓ and R denote the trend component, the seasonal component produced by decoder block ℓ , and the residual output of the final decoder block, respectively. The superscript (Y) denotes the corresponding future segment and is omitted in equation (4) of the main text. In the loss below, $\mathcal{F}$ denotes the discrete Fourier transform, and Re and Im denote its real and imaginary parts.

$$\begin{aligned} \mathcal{L}_{\text{DiffusionTS}} &= \mathbb{E}_{Z_0, s, \epsilon} w_s \left[\left\| Z_0 - \hat{Z}_0 \right\|_F^2 \right. \\ &\quad \left. + \lambda_F \left(\left\| \text{Re } \mathcal{F}(Z_0) - \text{Re } \mathcal{F}(\hat{Z}_0) \right\|_F^2 + \left\| \text{Im } \mathcal{F}(Z_0) - \text{Im } \mathcal{F}(\hat{Z}_0) \right\|_F^2 \right) \right], \end{aligned} \quad (\text{S11})$$

where $w_s = \sqrt{\alpha_s} \sqrt{1 - \alpha_s} / (100\beta_s)$ is the diffusion-step weight and $\lambda_F = \sqrt{L + H} / 5$ is the Fourier-domain loss weight. During conditional generation, a history mask identified the observed positions and reverse sampling filled the future positions. DiffusionTS

does not use an adjacency matrix and was applied independently to each node-level sequence.

TMDM

TMDM[15] uses a Transformer encoding of the observation window X to parameterize an approximate posterior distribution over a latent representation z :

$$q_\omega(z | X) = \mathcal{N}(z; \mu_z(X), \text{diag}\{\exp[\log \sigma_z^2(X)]\}), \quad (\text{S12})$$

where ω denotes the trainable parameters of the conditional encoder, and $\mu_z(X)$ and $\log \sigma_z^2(X)$ are its estimates of the latent mean and log-variance, respectively. The operator $\text{diag}\{\cdot\}$ constructs a diagonal covariance matrix, and the exponential is applied elementwise. A sample $z \sim q_\omega(z | X)$ is decoded into a conditional representation $\hat{Y}(X, z)$ of the future sequence. In the endpoint and diffusion equations, this sampled representation is abbreviated as $\hat{Y}(X)$, consistent with the notation used in the main text. It defines the endpoint distribution as $p(Y_S | X) = \mathcal{N}(Y_S; \hat{Y}(X), I)$. The corresponding forward marginal is

$$q(Y_s | Y_0, X) = \mathcal{N}(Y_s; \sqrt{\bar{\alpha}_s} Y_0 + (1 - \sqrt{\bar{\alpha}_s}) \hat{Y}(X), (1 - \bar{\alpha}_s) I). \quad (\text{S13})$$

The conditional model was optimized using a reconstruction term and latent regularization:

$$\mathcal{L}_{\text{cond}} = \mathbb{E}_{q_\omega(z|X)} \left[\left\| Y_0 - \hat{Y}(X, z) \right\|_F^2 \right] + k_z D_{\text{KL}} [q_\omega(z | X) \| \mathcal{N}(0, I)], \quad (\text{S14})$$

where D_{KL} denotes the Kullback–Leibler divergence and k_z controls the weight of the latent regularization term. The complete model was trained using the hybrid objective

$$\mathcal{L}_{\text{TMDM}} = \mathbb{E}_{X, Y_0, s, \epsilon} \left[\left\| \epsilon - \epsilon_\theta(Y_s, \hat{Y}(X), s) \right\|_F^2 \right] + k_{\text{cond}} \mathcal{L}_{\text{cond}}, \quad (\text{S15})$$

where $\epsilon \sim \mathcal{N}(0, I)$, ϵ_θ denotes the noise estimated by the denoising network, and k_{cond} weights the conditional-model objective. The expectation is taken over training pairs, sampled diffusion steps and Gaussian noise. We used $k_z = 0.01$ and $k_{\text{cond}} = 1$. Both the forward and reverse processes are conditioned on this representation, while the endpoint distribution retains an identity covariance.

NsDiff

NsDiff[16] parameterizes the forward-diffusion endpoint as

$$p(Y_S | X) = \mathcal{N}(Y_S; f_\phi(X), g_\psi(X)), \quad (\text{S16})$$

where $f_\phi(X)$ and $g_\psi(X)$ estimate the history-dependent endpoint mean and endpoint variance, respectively. The output of $g_\psi(X)$ parameterizes the endpoint variance of

the forward diffusion process, rather than the variance of the conditional predictive distribution $p(Y_0 | X)$. The endpoint-mean estimator was pretrained using

$$\mathcal{L}_f = \mathbb{E}_{X, Y_0} \left[\frac{1}{H|\mathcal{D}|} \|f_\phi(X) - Y_0\|_F^2 \right], \quad (\text{S17})$$

where $\mathcal{D}$ is the index set of forecast dimensions and $|\mathcal{D}|$ is the number of these dimensions. To construct the local-variance target, a sliding-window variance of width W was calculated over the concatenated sequence $[X, Y_0]$. The H values corresponding to Y_0 formed $\hat{\Sigma}_0$. The endpoint-variance estimator was pretrained using

$$\mathcal{L}_g = \frac{1}{H|\mathcal{D}|} \left\| \sqrt{g_\psi(X)} - \sqrt{\hat{\Sigma}_0} \right\|_F^2, \quad (\text{S18})$$

where the square root is applied elementwise and $\|\cdot\|_F$ denotes the Frobenius norm. The loss was averaged over the forecasting pairs in each minibatch. When optimized only with $\mathcal{L}_g$, g_ψ was used as the variance-only estimator in the controlled analysis.

The uncertainty-aware noise schedule (UANS) incorporates $g_\psi(X)$ and $\hat{\Sigma}_0$ into the forward process:

$$q(Y_s | Y_{s-1}, X) = \mathcal{N}\left(Y_s; \sqrt{\alpha_s}Y_{s-1} + (1 - \sqrt{\alpha_s})f_\phi(X), \beta_s^2 g_\psi(X) + \alpha_s \beta_s \hat{\Sigma}_0\right). \quad (\text{S19})$$

The denoising network takes Y_s , $f_\phi(X)$, $g_\psi(X)$ and s as inputs and estimates the noise and diagonal variance of the reverse process:

$$(\epsilon_\theta, \Sigma_\theta) = \xi_\theta(Y_s, f_\phi(X), g_\psi(X), s). \quad (\text{S20})$$

Following equation (13) of the original NsDiff formulation, the diffusion loss was

$$\mathcal{L}_{\text{diff}} = \mathbb{E}_{X, Y_0, s, \epsilon} \left[\|\epsilon - \epsilon_\theta\|_2^2 + \sum_i \frac{\tilde{\Sigma}_{s,i}}{\Sigma_{\theta,i}} - \sum_i \log \left(\frac{\tilde{\Sigma}_{s,i}}{\Sigma_{\theta,i}} \right) \right], \quad (\text{S21})$$

up to an additive constant. Here, $\epsilon \sim \mathcal{N}(0, I)$ is the noise added in the forward process, $\tilde{\Sigma}_s$ is the analytical posterior variance, and i indexes its diagonal elements. In the complete model used in this study, the three objectives were combined as

$$\mathcal{L}_{\text{NsDiff}} = \mathcal{L}_{\text{diff}} + \mathcal{L}_f + \mathcal{L}_g. \quad (\text{S22})$$

The pretrained f_ϕ and g_ψ modules remained trainable during this optimization. Consequently, the complete endpoint-variance estimator was optimized through both $\mathcal{L}_g$ and the generative diffusion objective, whereas the variance-only estimator was trained solely with $\mathcal{L}_g$.

Training, inference and experimental controls

Architecture-comparison training

Forecasting pairs were formed using equation (S7). DiffSTG processed all nodes jointly and received the network adjacency matrix. DiffusionTS, TMDM and NsDiff were trained on individual node-level sequences without graph input. DiffSTG perturbed a masked complete sequence, DiffusionTS reconstructed $[X, Y_0]$, and TMDM and NsDiff optimized objectives defined over the future sequence. Training data were standardized using statistics calculated from the training partition. The architecture comparison used $L = H = 100$, with 80% of the available forecasting pairs assigned to training. All models were trained for 50 epochs using Adam with a learning rate of 10^{-3} and weight decay of 10^{-5} . Each evaluated observation window generated 100 future trajectories. Model-specific diffusion steps, sampling steps and batch sizes are listed in Supplementary Table 3.

DiffSTG used a two-block UGnet with a hidden dimension of 4 and channel multipliers of (2, 2). DiffusionTS used three encoder layers, six decoder layers, four attention heads and a cosine noise schedule. TMDM used two encoder layers, one decoder layer, four attention heads and a linear noise schedule. NsDiff used four encoder layers, two decoder layers, eight attention heads and a linear noise schedule. The hidden Transformer dimension d_{model} was 64 for DiffusionTS and TMDM and 512 for NsDiff.

Model-specific inference

DiffSTG used 200 forward-diffusion steps during training and a 20-step deterministic DDIM sampler[17]. Its history mask retained the complete observed window while the future positions were generated jointly. DiffusionTS used 100 diffusion and sampling steps. At each reverse step, the observed positions were reset to the corresponding forward-perturbed history. Conditional reconstruction updates were applied to the unobserved positions, with zero updates below 5% of the diffusion trajectory, three above 90%, two between 75% and 90%, and one otherwise. TMDM and NsDiff used 20 reverse steps. TMDM initialized sampling from $\mathcal{N}(\hat{Y}(X), I)$, whereas NsDiff initialized from equation (S16). Each evaluated window produced 100 future trajectories for the sampling-based uncertainty calculation.

For NsDiff, the reverse network predicted noise and reverse variance at each step. The latter was combined with $g_{\psi}(X)$ and the diffusion schedule to recover a positive estimate of the future local variance. This estimate entered the analytical posterior mean and variance before sampling Y_{s-1} . The sliding-variance width W was 50 points for the networked and empirical analyses and 100 points for the SLBP analyses.

Controlled and empirical analyses

The controlled SLBP analyses used $L = H = 200$, 20 forward-diffusion steps and 20 reverse-sampling steps. The models were trained for 50 epochs with a batch size of 512 using Adam, with a learning rate of 10^{-3} and weight decay of 10^{-5} . The SLBP trajectories were sampled every 100 simulation-time units. Each observation window

generated 100 future trajectories, and the window advanced by 10 sampled points during evaluation.

Prediction-horizon controls used $H = 200, 500$ and $1,000$ with $L = 200$, whereas observation-window controls used $L = 200, 500$ and $1,000$ with $H = 200$. For the transition-scenario analysis, the window advanced by 10 sampled points when $T = 10^6$ and by one sampled point when $T = 5 \times 10^4$. The empirical model used $L = H = 200$, 20 forward-diffusion steps and 20 reverse-sampling steps. It was trained for 50 epochs with a batch size of 50 using the same optimizer settings and 90% of the available forecasting pairs. During empirical evaluation, the observation window advanced by one observation, and $g_\psi(X)$ was evaluated once for each window.

For the topology evaluation, one NsDiff model was trained for each dynamical system using trajectories pooled from Barabási–Albert, Erdős–Rényi and Watts–Strogatz networks. Each trained model was evaluated on held-out trajectories from the three network topologies. For the cross-dynamics evaluation, separate NsDiff models were trained on resource–biomass, Wilson–Cowan neuronal or SIS dynamics and then evaluated on all three dynamical systems. All evaluations used fixed trained models without test-time parameter updates.

The training-source control compared models trained on trajectory segments containing a single state regime with a model trained on combined segments containing both low- and high-state dynamics. These models were evaluated on complete trajectories with both increasing and decreasing nutrient levels. Stationary-data controls used the fixed nutrient levels and training-noise settings described above, while evaluation used time-dependent trajectories with $T = 10^6$ and $D = 10^{-5}$. The attractor-regime comparison used stationary training trajectories generated at $N = 0.5, 2.5$ or 5 with $D = 10^{-3}$, representing the low-state single-attractor, bistable and high-state single-attractor regimes, respectively. All models were evaluated on time-dependent trajectories that were not included in training.

Three NsDiff variants were constructed without the endpoint-variance estimator g_ψ , the endpoint-mean estimator f_ϕ , or the uncertainty-aware noise schedule. The variants were evaluated on the same held-out SLBP trajectories using sampling-based MPV and prediction error. The variance-only estimator retained g_ψ but was optimized using only $\mathcal{L}_g$, without the generative diffusion objective. For the smoothing control, standard NsDiff was applied to both raw observation windows and windows smoothed using a causal moving average of width $L/5 = 40$. Smoothing used only the current and preceding observations within each window.

Information available during training and inference

Model optimization used only paired observation windows and corresponding future sequences. At inference time, each prediction used the current observation window X_t and previously observed values. Realized future values were retained only for post hoc prediction-error calculation. Transition locations, bifurcation classes and control-parameter values were not supplied as model inputs or prediction targets.

Uncertainty and reference measures

For each observation window X_t , let $\hat{Y}_{t,h,d}^{(k)}$ denote the forecast at step h for evaluated node or variable dimension d in the k th of K independently generated future trajectories. Let $\mathcal{D}$ denote the set of evaluated nodes or variable dimensions and $|\mathcal{D}|$ its cardinality. The ensemble mean is $\bar{Y}_{t,h,d} = K^{-1} \sum_{k=1}^K \hat{Y}_{t,h,d}^{(k)}$. The mean predictive variance (MPV) was defined as

$$\text{MPV}(X_t) = \frac{1}{H|\mathcal{D}|} \sum_{h=1}^H \sum_{d \in \mathcal{D}} \left[\frac{1}{K} \sum_{k=1}^K \left(\hat{Y}_{t,h,d}^{(k)} - \bar{Y}_{t,h,d} \right)^2 \right], \quad (\text{S23})$$

where $h \in \{1, \dots, H\}$ indexes forecast steps, $d \in \mathcal{D}$ indexes evaluated nodes or variable dimensions, and $k \in \{1, \dots, K\}$ indexes generated trajectories. In this section, K denotes the number of generated trajectories and is distinct from the SLBP vegetation parameter K in equation (S6). MPV was averaged over nodes for the networked systems and evaluated for the bream variable in the SLBP system. It was calculated after model training as a compact summary of generated-trajectory uncertainty.

For the empirical analysis, the endpoint-variance module provided the direct MPV estimate

$$\widehat{\text{MPV}}_g(X_t) = \frac{1}{H} \sum_{h=1}^H g_\psi(X_t)_h. \quad (\text{S24})$$

Here, $\widehat{\text{MPV}}_g(X_t)$ denotes the MPV estimate obtained directly from $g_\psi(X_t)$. Because each empirical Mo time series is univariate, $g_\psi(X_t)_h$ is scalar at forecast step h , and no additional average over output dimensions is required. Each empirical uncertainty curve was divided by the mean of its first 40 valid values.

Several additional measures were used to interpret the predictive-uncertainty results. Prediction error was calculated separately as the mean absolute error between the ensemble-mean forecast and the realized future sequence. For the PCA analysis, each generated trajectory was flattened into a P -dimensional vector. If $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_P$ denote the eigenvalues of the covariance matrix across generated trajectories, the PCA-based effective dimension is

$$d_{0.8} = \min \left\{ d : \frac{\sum_{j=1}^d \lambda_j}{\sum_{j=1}^P \lambda_j} \geq 0.8 \right\}. \quad (\text{S25})$$

Here, $d_{0.8}$ is the minimum number of principal components required to explain at least 80% of the cumulative variance. MPV quantifies the overall dispersion of the generated trajectories, whereas $d_{0.8}$ characterizes how their variance is distributed among dominant linear modes. A decrease in MPV does not necessarily imply a decrease in $d_{0.8}$.

Lag-1 autocorrelation and variance[18] and sample entropy[19] were calculated from Gaussian-detrended empirical time series using a sliding window containing 50% of the available pre-transition series. Sample entropy was evaluated with embedding dimension $m_{\text{SE}} = 1$. The supervised reference used 20 released CNN-LSTM classifiers

for length-500 inputs[11]. Each input was detrended by Lowess smoothing[20] with a span of 0.2, predictions were evaluated every 10 observations, and probabilities were averaged across classifiers for the fold, Hopf, transcritical and neutral classes.

Transition locations in the resource–biomass and neuronal simulations were defined by the maximum absolute change over a 10-point interval in the network-averaged state. For SIS dynamics, the transition was the first 10-point sliding mean that crossed 10^{-2} in the direction of control-parameter change. The maximum adjacent change was used when no threshold crossing occurred. For the topology summary, the early pre-transition baseline B was the mean MPV between 1% and 20% of the available pre-transition MPV points. The MPV collapse depth was

$$\Delta_{\text{MPV}} = 100 \frac{B - M_{\min}}{B}, \quad (\text{S26})$$

where $M_{\min}$ is the minimum MPV within the predefined interval around the realized transition. Values were grouped by dynamical system, topology and direction of control-parameter change and summarized by the mean and sample standard deviation.

Supplementary Tables

Supplementary Table 1 Network topology and simulation settings for the networked dynamical systems.

(a) Network topology settings.

Topology	Network size, n	Construction parameters
Barabási–Albert (BA)	30, 50, 70, 100	$m \in [\text{round}(0.10n), \text{round}(0.25n)]$
Erdős–Rényi (ER)	30, 50, 70, 100	$p_{\text{edge}} \in [0.10, 0.90]$
Watts–Strogatz (WS)	30, 50, 70, 100	$n_{\text{nei}} \in [\text{round}(0.15n), \text{round}(0.35n)];$ $p_{\text{rewire}} \in [0, 0.15]$

For each topology and network size, ten connected undirected network realizations were generated. Construction parameters were sampled uniformly and independently for each realization, and disconnected realizations were regenerated. In the BA model, m is the number of edges formed by each newly added node through preferential attachment. In the ER model, p_{edge} is the independent connection probability for each node pair. In the WS model, n_{nei} is the number of nearest neighbors on each side of the initial ring and p_{rewire} is the edge-rewiring probability.

(b) Dynamical-system settings.

Dynamical system	Fixed parameters	Noise intensity	Control parameter
Resource–biomass	$r = 0.7, 0.8, 0.9; k = 10;$ $d = 0.5$	$\eta = 0.5, 0.005$	$\varepsilon : 0 \leftrightarrow 2.5; T = 10,000$
Wilson–Cowan neuronal	$\tau = 2.0, 2.2, 2.5; \mu = 1.5$	$\eta = 10^{-2}, 10^{-4}$	$\varepsilon : 0 \leftrightarrow 1; T = 10,000$
SIS epidemic	$\delta = 0.1, 0.5, 0.7$	$\eta = 10^{-4}, 10^{-6}$	$\varepsilon : 0.001 \leftrightarrow 0.11;$ $T = 1,000$

Complete parameter definitions are provided in Eqs. (S1)–(S3).

Supplementary Table 2 Parameter values for the shallow-lake bream–pike system.

Symbol	Definition	Value
i_b	Immigration rate of bream	3×10^{-4}
i_p	Immigration rate of pike	3×10^{-4}
r	Maximum growth rate of bream	7.5×10^{-3}
H_1	Half-saturation constant	0.5
H_2	Half-saturation constant	0.1
H_3	Half-saturation constant	20
H_4	Half-saturation constant	15
c_b	Intraspecific competition coefficient for bream	7.5×10^{-5}
c_p	Intraspecific competition coefficient for pike	2.75×10^{-4}
p_r	Maximum predation rate of pike	5×10^{-2}
c_e	Pike food-conversion efficiency for growth	0.1
m_p	Mortality rate of pike	2.25×10^{-3}
K	Maximum vegetation coverage	1

Supplementary Table 3 Model-specific configurations for the architecture-comparison experiments.

Model	Forward-diffusion steps	Reverse-sampling steps	Batch size
DiffSTG	200	20	200
DiffusionTS	100	100	512
TMDM	20	20	512
NsDiff	20	20	2,160

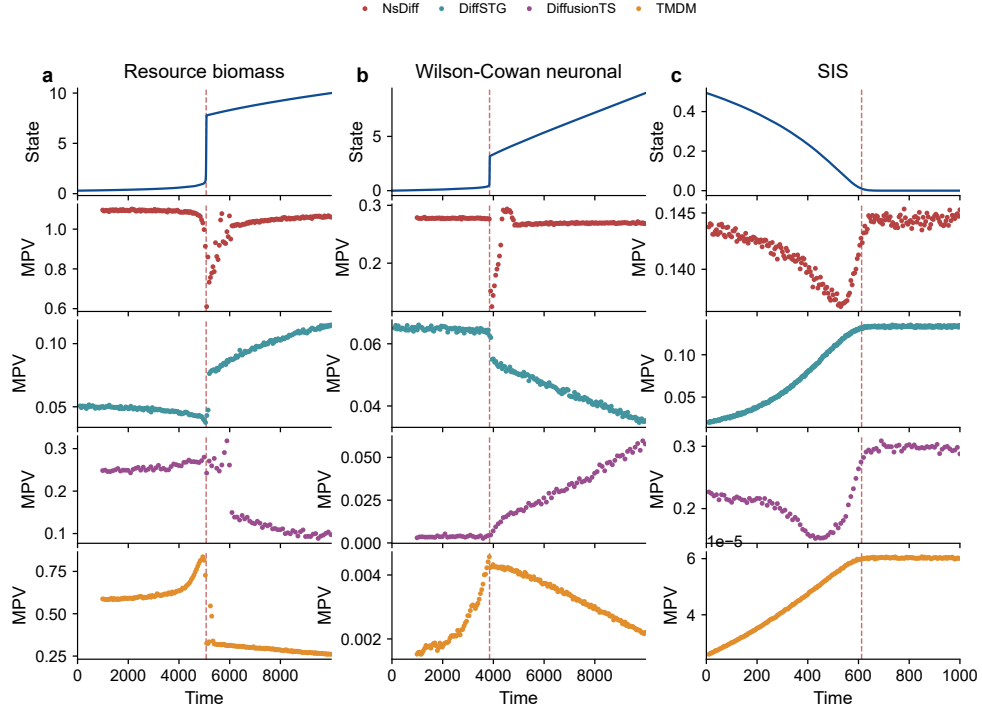
Supplementary Table 4 Mediterranean Sea datasets and empirical-analysis settings.

The main text presents S3-MS21, S4-MS66 and S5-64PE; the remaining datasets are shown in Supplementary Figure 11.

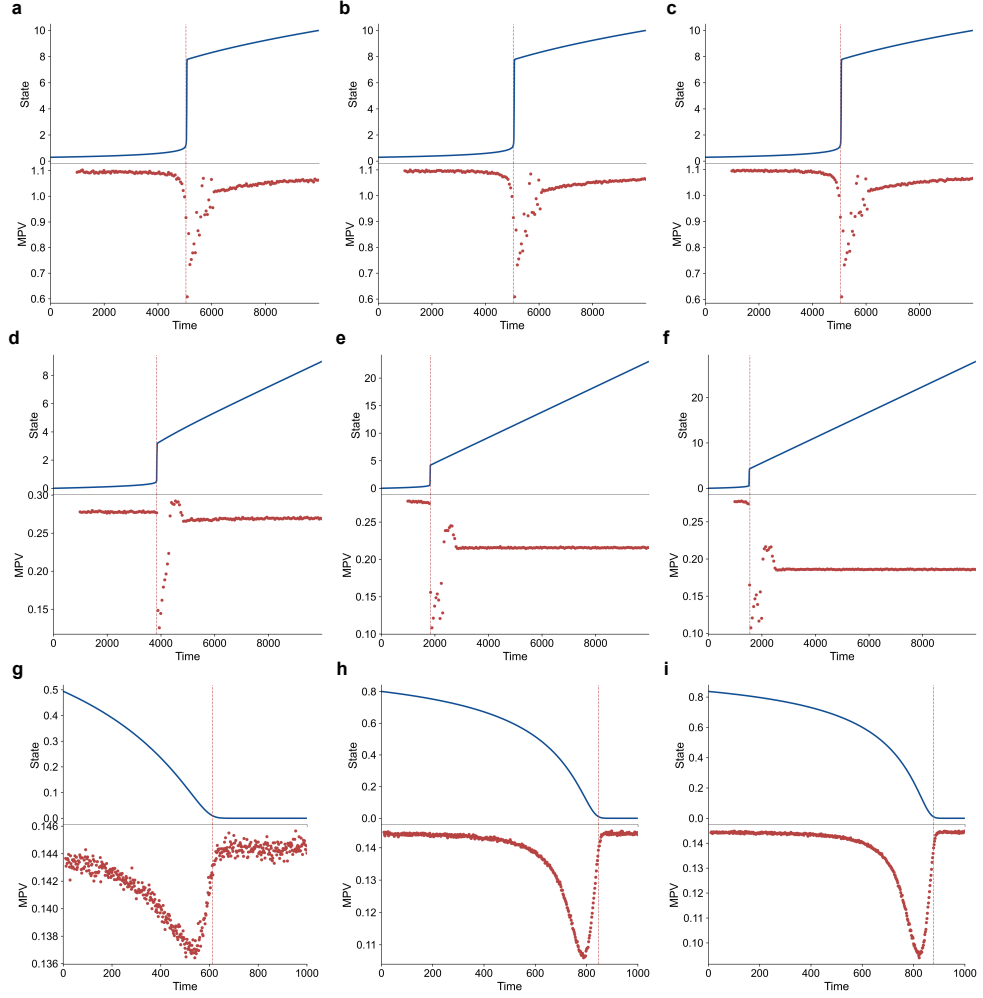
Source ID	Dataset	Core	Reference interval (ka BP)	Observations	Location
1	S1-MS21	MS21	10.5–9.7	1,328	Supplementary
2	S1-MS66	MS66	10.5–9.7	691	Supplementary
3	S3-MS21	MS21	85.8–85.1	862	Main
4	S3-MS66	MS66	85.8–85.1	388	Supplementary
5	S3-64PE	64PE	85.8–85.1	433	Supplementary
6	S4-MS66	MS66	107.8–107.2	513	Main
7	S4-64PE	64PE	107.8–107.2	515	Supplementary
8	S5-MS66	MS66	128.35–127.5	490	Supplementary
9	S5-64PE	64PE	128.35–127.5	472	Main
10	S6-64PE	64PE	177.25–176.5	258	Supplementary
11	S7-64PE	64PE	198.5–197.8	429	Supplementary
12	S8-64PE	64PE	224.9–223.6	270	Supplementary

Analysis settings. Gaussian detrending, bandwidth 0.2; $L = H = 200$; sliding step 1; direct $g_\psi(X)$ estimate normalized by its first 40 valid values; classical-reference sliding window 0.5

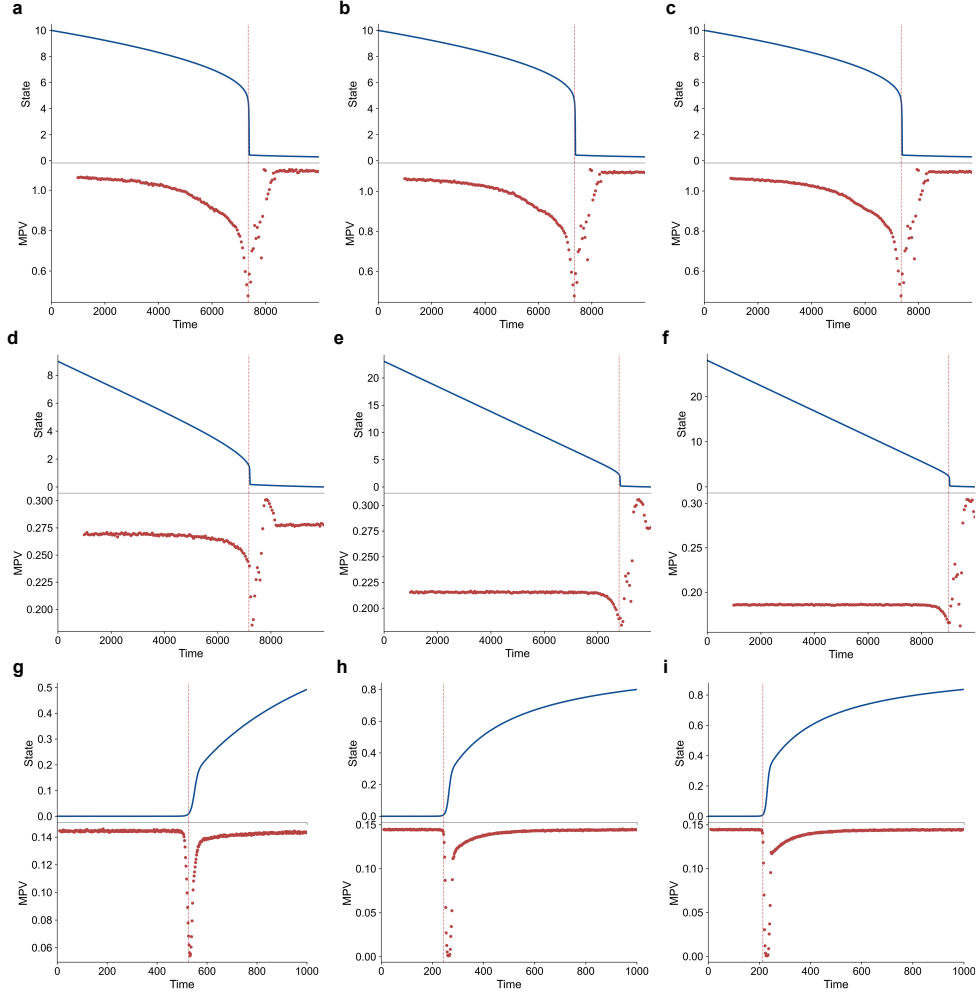
Supplementary Figures



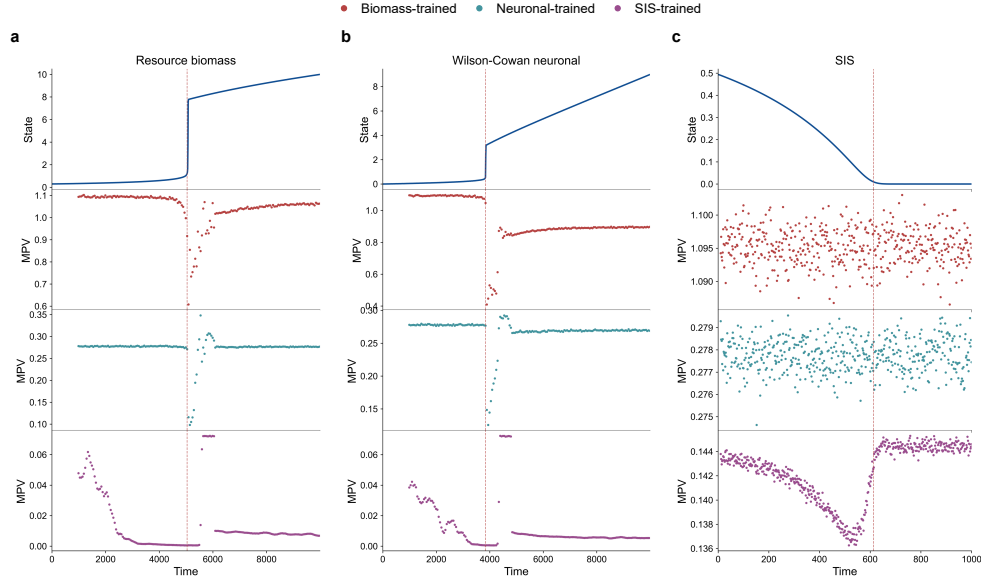
Supplementary Figure 1 Architecture-dependent predictive uncertainty with a decreasing control parameter. **a–c**, Network-averaged state and predictive uncertainty for the resource–biomass (a), Wilson–Cowan neuronal (b) and susceptible–infected–susceptible (SIS) epidemic (c) systems simulated on Barabási–Albert networks with a decreasing control parameter. In each panel, the blue line shows the network-averaged state. Red, teal, purple and orange points show mean predictive variance (MPV) produced by the non-stationary diffusion model (NsDiff), diffusion-based spatio-temporal graph model (DiffSTG), DiffusionTS and Transformer-modulated diffusion model (TMDM), respectively. MPV summarizes the dispersion among generated future trajectories. Vertical red dashed lines mark the transition times determined retrospectively from the raw network-averaged states. MPV is plotted on different scales because its magnitude differs across models and systems.



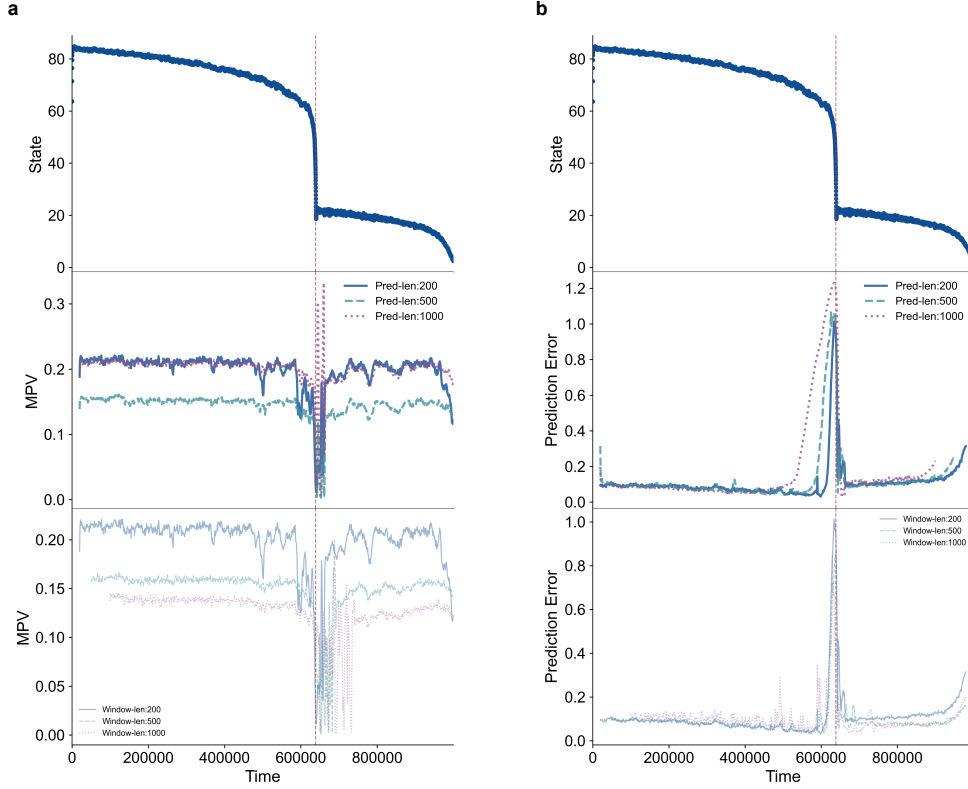
Supplementary Figure 2 Predictive uncertainty across network topologies with a decreasing control parameter. **a–c**, Resource–biomass; **d–f**, Wilson–Cowan neuronal; **g–i**, susceptible–infected–susceptible (SIS) epidemic dynamics. Panels **a,d,g**, **b,e,h** and **c,f,i** correspond to Barabási–Albert (BA), Erdős–Rényi (ER) and Watts–Strogatz (WS) networks, respectively. In each panel, the blue line shows the network-averaged state and the red points show the mean predictive variance (MPV) produced by the non-stationary diffusion model (NsDiff). Red dashed lines mark transition times.



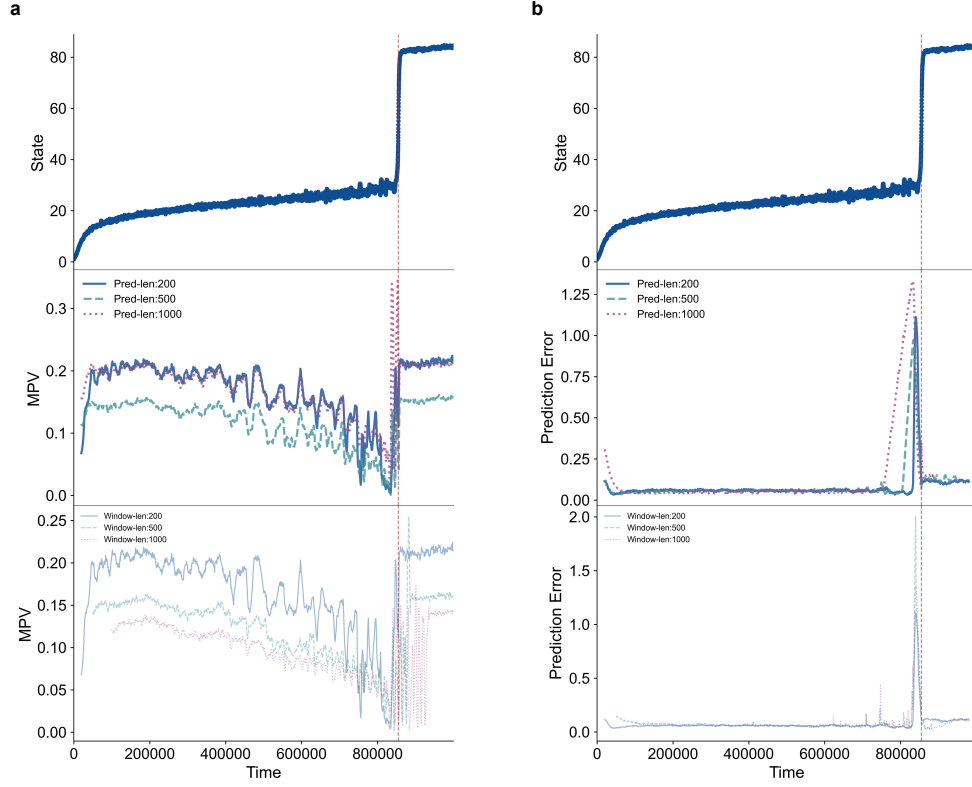
Supplementary Figure 3 Predictive uncertainty across network topologies with an increasing control parameter. **a–c**, Resource–biomass; **d–f**, Wilson–Cowan neuronal; **g–i**, susceptible–infected–susceptible (SIS) epidemic dynamics. Panels **a,d,g**, **b,e,h** and **c,f,i** correspond to Barabási–Albert (BA), Erdős–Rényi (ER) and Watts–Strogatz (WS) networks, respectively. In each panel, the blue line shows the network-averaged state and the red points show the mean predictive variance (MPV) produced by the non-stationary diffusion model (NsDiff). Red dashed lines mark transition times.



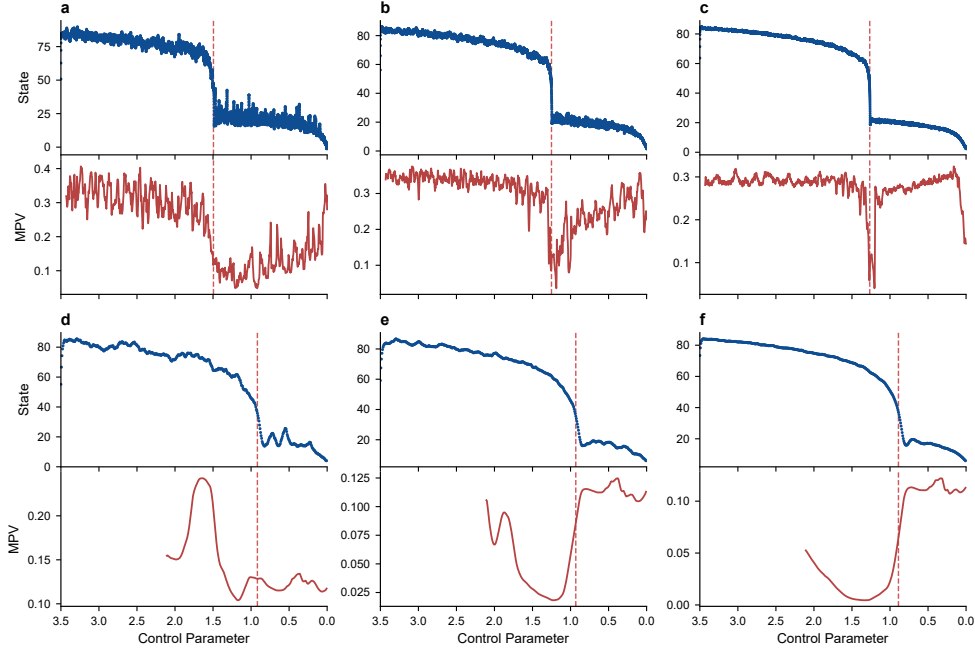
Supplementary Figure 4 Transfer of predictive uncertainty across dynamical systems. **a–c**, Resource–biomass (**a**), Wilson–Cowan neuronal (**b**) and susceptible–infected–susceptible (SIS) epidemic (**c**) test dynamics with a decreasing control parameter. In each panel, the blue line shows the network-averaged state. Red, teal and purple points show the mean predictive variance (MPV) produced by non-stationary diffusion models (NsDiff) trained on resource–biomass, Wilson–Cowan neuronal and SIS epidemic dynamics, respectively. Red dashed lines mark transition times.



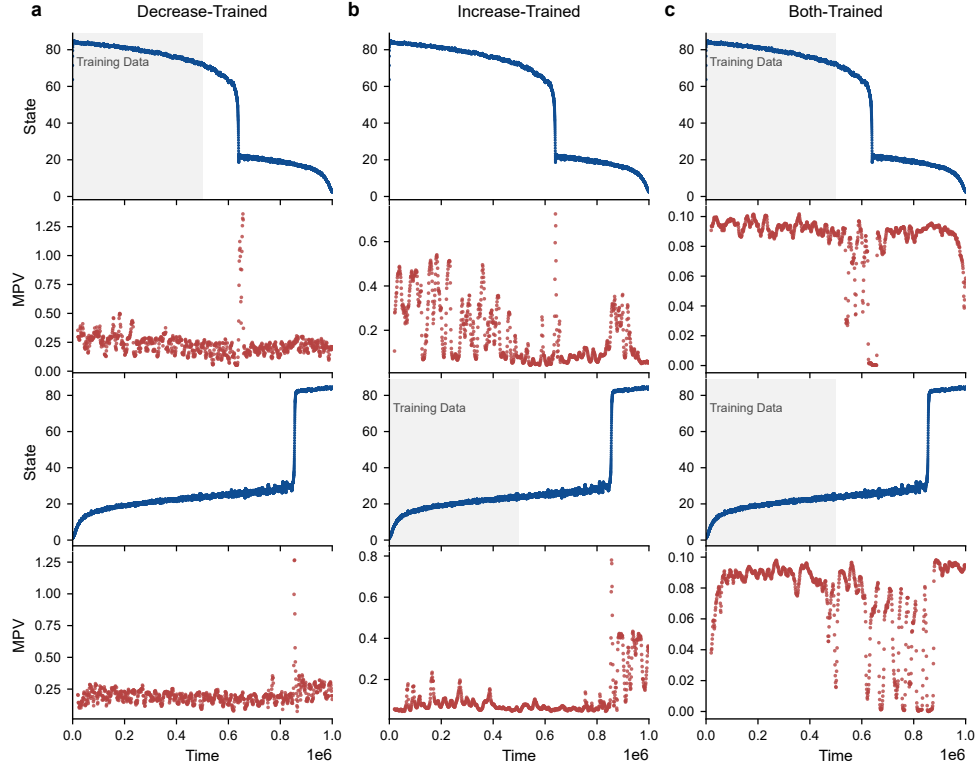
Supplementary Figure 5 Sensitivity to prediction horizon and observation-window length with decreasing nutrient level. a,b, Sensitivity analyses for the shallow-lake bream-pike (SLBP) system as the nutrient level decreases. Panel a reports mean predictive variance (MPV), and panel b reports prediction error. The three aligned plots in each panel show, in order, the system state, results obtained by varying the prediction horizon while fixing the observation-window length at 200, and results obtained by varying the observation-window length while fixing the prediction horizon at 200. Prediction horizons or observation-window lengths of 200, 500 and 1,000 sampled points are shown by blue solid, cyan dashed and purple dotted curves, respectively. Blue points show the system state. Vertical red dashed lines mark the transition time determined from the raw state trajectory.



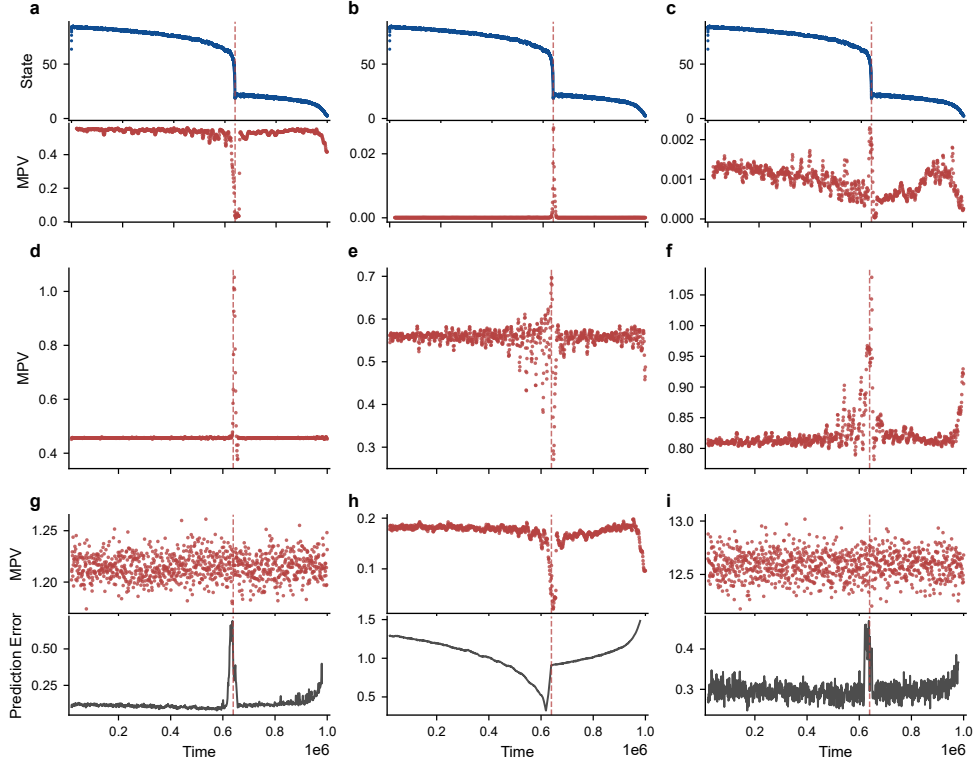
Supplementary Figure 6 Sensitivity to prediction horizon and observation-window length with increasing nutrient level. **a,b**, Sensitivity analyses for the shallow-lake bream-pike (SLBP) system as the nutrient level increases. Panel **a** reports mean predictive variance (MPV), and panel **b** reports prediction error. The three aligned plots in each panel show, in order, the system state, results obtained by varying the prediction horizon while fixing the observation-window length at 200, and results obtained by varying the observation-window length while fixing the prediction horizon at 200. Prediction horizons or observation-window lengths of 200, 500 and 1,000 sampled points are shown by blue solid, cyan dashed and purple dotted curves, respectively. Blue points show the system state. Vertical red dashed lines mark the transition time determined from the raw state trajectory.



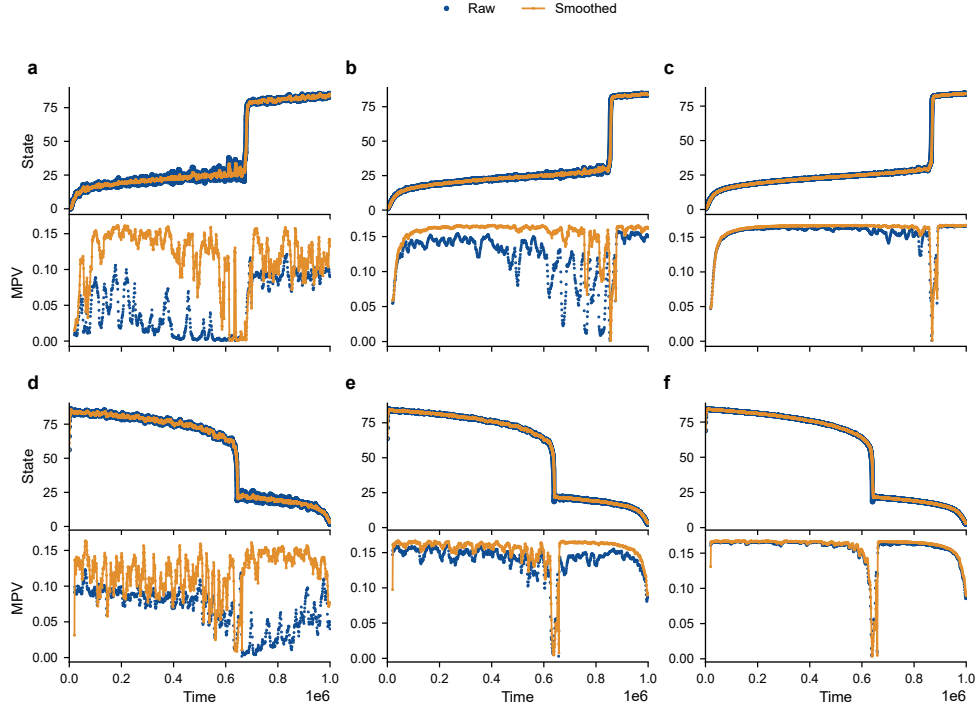
Supplementary Figure 7 Effects of control-parameter change rate and noise on transitions. **a–f**, State trajectories (blue) and corresponding mean predictive variance (MPV; red) as the nutrient control parameter N decreases linearly from 3.5 to 0 in the shallow-lake bream–pike (SLBP) system. **a–c**, Slow control-parameter change with duration $T = 10^6$ and noise intensities $D = 5 \times 10^{-4}$, 1×10^{-4} and 1×10^{-5} , respectively. These settings produce a noise-induced transition in **a** and bifurcation-induced transitions in **b,c**. **d–f**, Faster control-parameter change with $T = 5 \times 10^4$ and the same noise intensities. The state transitions occur after the static bifurcation threshold, consistent with rate-dependent tipping delay. Vertical red dashed lines indicate the control-parameter values corresponding to the transitions. Here, T denotes the duration of control-parameter change and D denotes noise intensity.



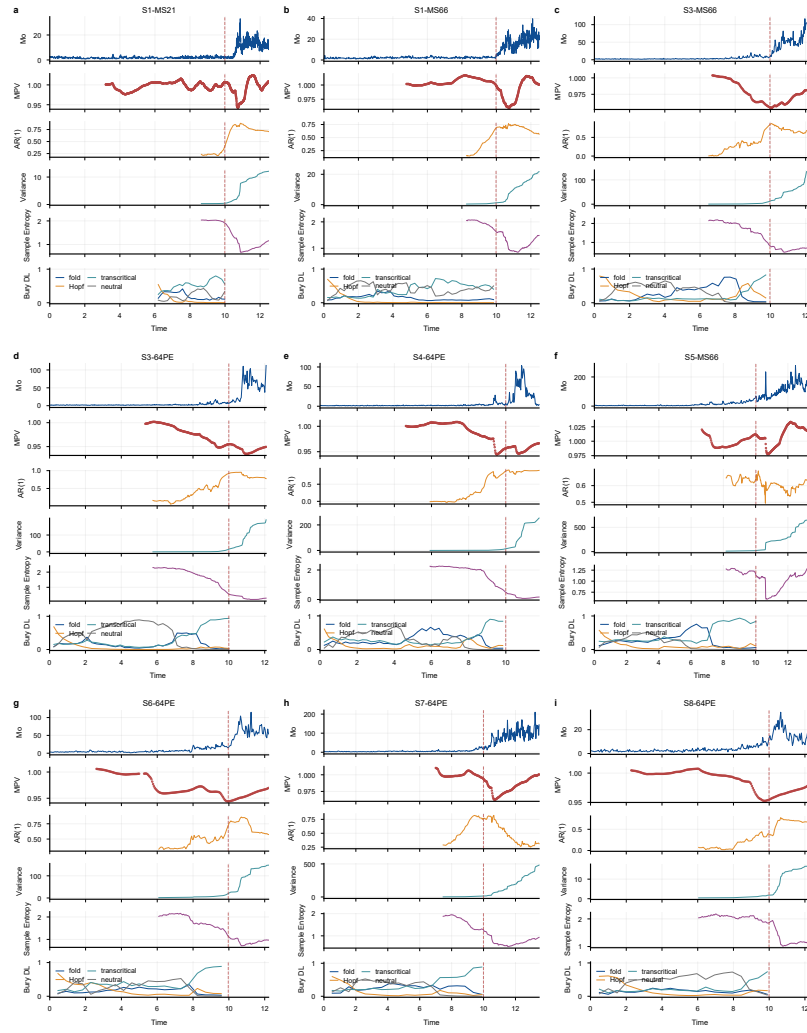
Supplementary Figure 8 Effects of training-data composition on predictive uncertainty. **a–c**, Results from shallow-lake bream-pike (SLBP) models trained using different trajectory segments. Each panel presents, in order, the state trajectory with a decreasing nutrient level and its mean predictive variance (MPV), followed by the state trajectory with an increasing nutrient level and its MPV. State trajectories are shown in blue and MPV in red. **a**, Training used only the grey-shaded high-state segment of the decreasing trajectory. **b**, Training used only the shaded low-state segment of the increasing trajectory. **c**, Training used both shaded segments. Each trained model was evaluated on both test trajectories. Grey shading marks the trajectory segments included in training.



Supplementary Figure 9 Effects of training data and model modules on uncertainty collapse. All models are evaluated on the same time-varying shallow-lake bream-pike (SLBP) test trajectory with a decreasing nutrient control parameter N . **a–c**, Test-state trajectory (blue) and mean predictive variance (MPV; red) for models trained on a pooled dataset of stationary trajectories generated at $N = 0.5, 1, 2.5, 3$ and 5 . The training noise intensity was $D = 10^{-3}, 10^{-4}$ and 10^{-5} in **a**, **b** and **c**, respectively. **d–f**, MPV for models trained separately on stationary trajectories generated at $N = 0.5$ (**d**), 2.5 (**e**) or 5 (**f**), with $D = 10^{-3}$. These values represent the low-state single-attractor, bistable and high-state single-attractor regimes, respectively. **g–i**, Sampling-based MPV (red) and prediction error (grey) for models without the endpoint-variance estimator g_ψ (**g**), endpoint-mean module f_ϕ (**h**) or uncertainty-aware noise schedule (UANS; **i**). Vertical red dashed lines mark the transition time. Here, D denotes noise intensity.



Supplementary Figure 10 Effects of causal smoothing on predictive uncertainty. **a–c,** Increasing nutrient level with noise intensities $D = 10^{-4}$, 10^{-5} and 10^{-6} , respectively. **d–f,** Decreasing nutrient level with the same sequence of noise intensities. Each panel shows, in order, the shallow-lake bream–pike (SLBP) state trajectory and the corresponding mean predictive variance (MPV). Blue points denote the raw state and the MPV calculated from the corresponding raw observation windows. Orange curves denote the causally smoothed state and the MPV calculated from the corresponding smoothed observation windows. The causal moving average used a window width of 40 sampled points and included only the current and preceding observations. Here, D denotes noise intensity.



Supplementary Figure 11 Predictive uncertainty in additional Mediterranean Sea datasets. **a–i**, Results for S1-MS21 (**a**), S1-MS66 (**b**), S3-MS66 (**c**), S3-64PE (**d**), S4-64PE (**e**), S5-MS66 (**f**), S6-64PE (**g**), S7-64PE (**h**) and S8-64PE (**i**). Each panel shows the molybdenum (Mo) time series in blue, the mean predictive variance (MPV) estimate derived from the endpoint-variance estimator $g_\psi(X)$ of the non-stationary diffusion model (NsDiff) in red, lag-1 autocorrelation in orange, variance in cyan and sample entropy in purple, where X denotes the current observation window. Class probabilities from the supervised model of Bury et al. are shown for fold (blue), Hopf (orange), transcritical (teal) and neutral (grey) classes. All derived quantities were evaluated along the time series using sliding observation windows. Vertical red dashed lines mark the reference transition locations. Axis scales differ among datasets and measures.

Supplementary References

- [1] Brauer, F. & Sánchez, D. A. Constant rate population harvesting: equilibrium and stability. *Theoretical Population Biology* **8**, 12–30 (1975).
- [2] Dakos, V., van Nes, E. H., Donangelo, R., Fort, H. & Scheffer, M. Spatial correlation as leading indicator of catastrophic shifts. *Theoretical Ecology* **3**, 163–174 (2010).
- [3] Wilson, H. R. & Cowan, J. D. A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue. *Kybernetik* **13**, 55–80 (1973).
- [4] Pastor-Satorras, R., Castellano, C., Van Mieghem, P. & Vespignani, A. Epidemic processes in complex networks. *Reviews of Modern Physics* **87**, 925–979 (2015).
- [5] Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509–512 (1999).
- [6] Erdős, P. & Rényi, A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* **5**, 17–61 (1960).
- [7] Watts, D. J. & Strogatz, S. H. Collective dynamics of ‘small-world’ networks. *Nature* **393**, 440–442 (1998).
- [8] Xu, L., Patterson, D., Levin, S. A. & Wang, J. Non-equilibrium early-warning signals for critical transitions in ecological systems. *Proceedings of the National Academy of Sciences* **120**, e2218663120 (2023).
- [9] Hennekam, R. *et al.* Early-warning signals for marine anoxic events. *Geophysical Research Letters* **47**, e2020GL089183 (2020).
- [10] Hennekam, R. *et al.* Calibrated XRF-scanning data (mm resolution) and calibration data (ICP-OES and ICP-MS) for elements Al, Ba, Mo, Ti, and U in Mediterranean cores MS21, MS66, and 64PE406E1. PANGAEA (2020). URL <https://doi.org/10.1594/PANGAEA.923197>.
- [11] Bury, T. M. *et al.* Deep learning for early warning signals of tipping points. *Proceedings of the National Academy of Sciences* **118**, e2106140118 (2021).
- [12] Bury, T. Deep learning for early warning signals of tipping points: supplementary data. Zenodo (2021). URL <https://doi.org/10.5281/zenodo.5527154>.
- [13] Wen, H. *et al.* DiffSTG: probabilistic spatio-temporal graph forecasting with denoising diffusion models. *Proc. 31st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 1–12 (2023). URL <https://doi.org/10.1145/3589132.3625614>.

- [14] Yuan, X. & Qiao, Y. *Diffusion-TS: interpretable diffusion for general time series generation*. *Proc. 12th International Conference on Learning Representations (ICLR)*, 41582–41610 (2024). URL <https://openreview.net/forum?id=4h1apFjO99>.
- [15] Li, Y. *et al.* *Transformer-modulated diffusion models for probabilistic multivariate time series forecasting*. *Proc. 12th International Conference on Learning Representations (ICLR)*, 18604–18622 (2024). URL <https://openreview.net/forum?id=qae04YACHs>.
- [16] Ye, W., Xu, Z. & Gui, N. *Non-stationary diffusion for probabilistic time series forecasting*. *Proc. 42nd International Conference on Machine Learning (ICML)*, 72112–72130 (2025). URL <https://proceedings.mlr.press/v267/ye25i.html>.
- [17] Song, J., Meng, C. & Ermon, S. *Denoising diffusion implicit models*. *Proc. 9th International Conference on Learning Representations (ICLR)* (2021). URL <https://openreview.net/forum?id=St1gjarCHLP>.
- [18] Dakos, V. *et al.* *Methods for detecting early warnings of critical transitions in time series illustrated using simulated ecological data*. *PLOS ONE* **7**, e41010 (2012).
- [19] Richman, J. S. & Moorman, J. R. *Physiological time-series analysis using approximate entropy and sample entropy*. *American Journal of Physiology-Heart and Circulatory Physiology* **278**, H2039–H2049 (2000).
- [20] Cleveland, W. S. *Robust locally weighted regression and smoothing scatterplots*. *Journal of the American Statistical Association* **74**, 829–836 (1979).