Supplementary Information for

Data Assimilation in Operator Algebras

David Freeman, Dimitrios Giannakis, Brian Mintz, Abbas Ourmazd, and Joanna Slawinska

Dimitrios Giannakis.
E-mail: dimitrios.giannakis@dartmouth.edu

This PDF file includes:

Supplementary text
Figs. S1 to S2
Tables S1 to S3
SI References

Supporting Information Text

This SI Appendix addresses the following topics:

1. Technical assumptions on the dynamical system, observation map, and forecast observable (section 1).
2. Finite-dimensional approximation methods underlying QMDA and associated pseudocode (section 2).
3. Quantum circuit implementation in Qiskit (sec 3).
4. Forecast skill metrics (section 4).
5. Properties of the Lorenz 96 (L96) multiscale and Community Climate System Model version 4 (CCSM4) datasets (section 5).

Equation numbers in the SI Appendix are prefixed by 'S'. Equation and figure numbers without S prefixes refer to equations and figures in the main text, respectively. Tables S1 and S2 provide a summary of the definitions of the main symbols used in the main text and SI Appendix. Table S3 summarizes the data attributes and QMDA parameters for the L96 multiscale and El Niño Southern Oscillation (ENSO) experiments. Figures S1 and S2 show forecast skill scores for the L96 multiscale and ENSO experiments, respectively, for various values of the QMDA Hilbert space dimension L.

Table S1. Symbols used in the main text and SI Appendix (continues to Table S2).

Symbol	Meaning
$\boldsymbol{A}_L$	$L \times L$ real matrix representing $\pi_L f$ in the $\{\phi_l\}$ basis of H_L
$\boldsymbol{A}_{L,N}$	$L \times L$ real matrix representing $\pi_{L,N} \hat f_N$ in the $\{\phi_{l,N}\}$ basis of $H_{L,N}$
$\mathfrak{A} = L^\infty(X,\mu)$	Abelian algebra of observables
$\mathfrak{A}_* = L^1(X,\mu)$	Predual of $\mathfrak{A}$
$\hat{\mathfrak{A}}_N = L^\infty(X,\mu_N)$	Abelian algebra of observables with respect to sampling measure
$\lvert \boldsymbol{b} \rangle$	Quantum computational basis vectors of $\mathbb{B}_{\mathfrak{n}}$
$\mathcal{B}(\mathbb{R})$	Borel σ-algebra on $\mathbb{R}$
$\mathfrak{B} = B(H)$	Algebra of bounded operators on H, equipped with operator norm, $\lVert A \rVert_{\mathfrak{B}}$
$\mathfrak{B}_* = B_1(H)$	Predual of $\mathfrak{B}$ (space of trace-class operators on H), equipped with trace norm, $\lVert A \rVert_{\mathfrak{B}_*}$
$\mathfrak{B}_L = B(H_L)$	Algebra of linear maps on H_L, identified with a subalgebra of $\mathfrak{B}$
$\hat{\mathfrak{B}}_N = B(\hat H_N)$	Algebra of linear maps on $\hat H_N$
$\hat{\mathfrak{B}}_{L,N} = B(H_{L,N})$	Algebra of linear maps on $H_{L,N}$, identified with a subalgebra of $\hat{\mathfrak{B}}_N$
$\mathbb{B}_{\mathfrak{n}}$	$2^{\mathfrak{n}}$-dimensional Hilbert space associated with $\mathfrak{n}$ qubits
$C(X)$	Space of continuous functions on X, equipped with uniform norm $\lVert f \rVert_{C(X)}$
$\mathfrak{C}$	Algebra of bounded operators on $C(X)$, equipped with operator norm $\lVert A \rVert_{\mathfrak{C}}$
$E_f : \mathcal{B}(\mathbb{R}) \to \mathfrak{A}$	Projection-valued measure (PVM) of f
$E_{\pi f} : \mathcal{B}(\mathbb{R}) \to \mathfrak{B}$	PVM of πf; shorthand notation: $E \equiv E_{\pi f}$
$E_{\pi_L f} : \mathcal{B}(\mathbb{R}) \to \mathfrak{B}_L$	PVM of $\pi_L f$; shorthand notation: $E_L \equiv E_{\pi_L f}$
$E_{\pi_{L,N} f} : \mathcal{B}(\mathbb{R}) \to \mathfrak{B}_{L,N}$	PVM of $\pi_{L,N} f$; shorthand notation: $E_{L,N} \equiv E_{\pi_{L,N} \hat f_N}$
$\boldsymbol{E}_L : \mathcal{B}(\mathbb{R}) \to \mathbb{M}_L$	Matrix representation of E_L in the $\{\phi_l\}$ basis of H_L
$\boldsymbol{E}_{L,N} : \mathcal{B}(\mathbb{R}) \to \mathbb{M}_L$	Matrix representation of $E_{L,N}$ in the $\{\phi_{l,N}\}$ basis of H_L
$\mathcal{E}(\mathfrak{A}) \subset \mathfrak{A}$	Effect space of $\mathfrak{A}$
$\mathcal{E}(\hat{\mathfrak{A}}_N) \subset \hat{\mathfrak{A}}_N$	Effect space of $\hat{\mathfrak{A}}_N$
$\mathcal{E}(\mathfrak{B}) \subset \mathfrak{B}$	Effect space of $\mathfrak{B}$
$\mathcal{E}(\mathfrak{B}_L) \subset \mathfrak{B}_L$	Effect space of $\mathfrak{B}_L$
$\mathcal{E}(\mathfrak{B}_{L,N}) \subset \mathfrak{B}_{L,N}$	Effect space of $\mathfrak{B}_{L,N}$
$F : Y \to \mathcal{E}(\mathfrak{A})$	Effect-valued feature map
$\mathcal{F} : Y \to \mathcal{E}(\mathfrak{B})$	Effect-valued feature map (operator-valued), $\mathcal{F} = \pi \circ F$
$\mathcal{F}_L : Y \to \mathcal{E}(\mathfrak{B}_L)$	Projected feature map, $\mathcal{F}_L = \mathbf{\Pi}_L \circ \mathcal{F}$
$\boldsymbol{F}_L : Y \to \mathbb{M}_L$	Matrix representation of $\mathcal{F}_L$ in the $\{\phi_l\}$ basis of H_L
$\hat F_N : Y \to \mathcal{E}(\hat{\mathfrak{A}}_N)$	Effect-valued feature map (data-dependent)
$\hat{\mathcal{F}}_N : Y \to \mathcal{E}(\hat{\mathfrak{B}}_N)$	Effect-valued feature map (operator-valued), $\hat{\mathcal{F}}_N = \hat\pi_N \circ \hat F_N$
$\mathcal{F}_{L,N} : Y \to \mathcal{E}(\mathfrak{B}_{L,N})$	Projected feature map, $\mathcal{F}_{L,N} = \mathbf{\Pi}_{L,N} \circ \hat{\mathcal{F}}_N$
$\boldsymbol{F}_{L,N} : Y \to \mathbb{M}_L$	Matrix representation of $\mathcal{F}_{L,N}$ in the $\{\phi_{l,N}\}$ basis of $H_{L,N}$
$f : X \to \mathbb{R}$	Forecast observable, identified with a self-adjoint element of $\mathfrak{A}$
$\hat f_N : X_N \to \mathbb{R}$	Restriction of f to training trajectory X_N, identified with self-adjoint element of $\hat{\mathfrak{A}}_N$; training values $f_n = f(x_n)$
$H = L^2(X,\mu)$	Hilbert space associated with the invariant measure
$H_L \subset H$	L-dimensional subspace spanned by leading L eigenfunctions ϕ_l of K
$\hat H_N = L^2(X,\mu_N)$	Hilbert space associated with the sampling measure
$H_{L,N} \subset \hat H_N$	L-dimensional subspace spanned by leading L eigenfunctions $\phi_{l,N}$ of K_N
$h : X \to Y$	Observation map
$K : H \to H$	Kernel integral operator associated with k
$k : X \times X \to \mathbb{R}$	Pullback kernel from training data space, $k(x,x') = \kappa(z(x), z(x'))$
$K_N : H_N \to H_N$	Kernel integral operator associated with k_N
$\boldsymbol{K}_N$	$N \times N$ real matrix representing K_N
$k_N : X \times X \to \mathbb{R}$	Data-dependent pullback kernel from training data space, $k_N(x,x') = \kappa_N(z(x), z(x'))$
L	Hilbert space dimension
$\mathbb{M}_L$	Algebra of $L \times L$ complex matrices
$\mathfrak{M}_{\mathfrak{n}}$	Algebra of bounded operators on $\mathbb{B}_{\mathfrak{n}}$, equipped with operator norm $\lVert A \rVert_{\mathfrak{M}_{\mathfrak{n}}}$
$\mathfrak{n}$	Number of qubits
N	Number of training samples
$P^t : \mathfrak{A}^* \to \mathfrak{A}^*$	Transfer operator on the dual of $\mathfrak{A}$, $P^t \alpha = \alpha \circ U^t$; restricts to $P^t : S(\mathfrak{A}) \to S(\mathfrak{A})$ and $P^t : S_*(\mathfrak{A}) \to S_*(\mathfrak{A})$
$\mathcal{P}^t : \mathfrak{B}^* \to \mathfrak{B}^*$	Transfer operator on the dual of $\mathfrak{B}$, $\mathcal{P}^t \gamma = \gamma \circ \mathcal{U}^t$; restricts to $\mathcal{P}^t : S(\mathfrak{B}) \to S(\mathfrak{B})$ and $\mathcal{P}^t : S_*(\mathfrak{B}) \to S_*(\mathfrak{B})$
$\mathcal{P}_L^{(t)} : \mathfrak{B}^* \to \mathfrak{B}^*$	Projected transfer operator, $\mathcal{P}_L^{(t)} \gamma = \gamma \circ \mathcal{U}_L^{(t)}$; identified with $\mathcal{P}_L^{(t)} : \mathfrak{B}_L^* \to \mathfrak{B}_L^*$, restricts to $\mathcal{P}_L^{(t)} : S(\mathfrak{B}_L) \to S(\mathfrak{B}_L)$
$\mathcal{P}_{L,N}^{(q)} : \hat{\mathfrak{B}}_N^* \to \hat{\mathfrak{B}}_N^*$	Transfer operator, $\mathcal{P}_{L,N}^{(q)} \gamma = \gamma \circ \mathcal{U}_{L,N}^{(q)}$; identified with $\mathcal{P}_{L,N}^{(q)} : \mathfrak{B}_{L,N}^* \to \mathfrak{B}_{L,N}^*$, restricts to $\mathcal{P}_{L,N}^{(q)} : S(\hat{\mathfrak{B}}_N) \to S(\hat{\mathfrak{B}}_N)$
$\mathbb{P}_{E,p} : \mathcal{B}(\mathbb{R}) \to [0,1]$	Probability distribution induced by PVM $E : \mathcal{B}(\mathbb{R}) \to \mathfrak{A}$ given $\omega_p \in S_*(\mathfrak{A})$; shorthand notation $\mathbb{P}_{f,p} \equiv P_{E_f,p}$
$\mathbb{P}_{E,\rho} : \mathcal{B}(\mathbb{R}) \to [0,1]$	Probability distribution induced by PVM $E : \mathcal{B}(\mathbb{R}) \to \mathfrak{B}$ given $\omega_\rho \in S_*(\mathfrak{B})$; shorthand notation $\mathbb{P}_{\pi f,\rho} \equiv P_{E_{\pi f},\rho}$
$S_0, \dots, S_{M-1} \subseteq \mathbb{R}$	Spectral bins (intervals) for evaluation of forecast distribution
$S(\mathfrak{A}) \subset \mathfrak{A}^*$	State space of $\mathfrak{A}$
$S_*(\mathfrak{A}) \subset S(\mathfrak{A})$	Space of normal states of $\mathfrak{A}$
$S(\mathfrak{B}) \subset \mathfrak{B}^*$	State space of $\mathfrak{B}$
$S_*(\mathfrak{B}) \subset S(\mathfrak{B})$	Space of normal states of $\mathfrak{B}$
$S_C(\mathfrak{B}) \subset S_*(\mathfrak{B})$	Normal states given by linear combinations of pure states with uniformly bounded continuous state vectors
$S(\mathfrak{B}_L)$	State space of $\mathfrak{B}_L$
$S(\mathfrak{B}_{L,N})$	State space of $\mathfrak{B}_{L,N}$

Table S2. Symbols used in the main text and SI Appendix (continues from Table S1).

Symbol	Meaning
$U^t : H \to H$	Koopman operator, $U^t f = f \circ \Phi^t$; restricts to $U^t : \mathfrak{A} \to \mathfrak{A}$
$\tilde{U}^t : H \to H$	Koopman operator on continuous functions, $U^t f = f \circ \Phi^t$
$\mathcal{U}^t : \mathfrak{B} \to \mathfrak{B}$	Induced Koopman operator, $\mathcal{U}^t A = U^t A U^{t*}$
$U_L^{(t)} : H \to H$	Projected Koopman operator, $U_L^{(t)} = \mathbf{\Pi}_L U^t$; identified with operator $U_L^{(t)} : H_L \to H_L$
$\mathcal{U}_L^{(t)} : \mathfrak{B} \to \mathfrak{B}$	Projected Koopman operator, $\mathcal{U}_L^{(t)} A = U_L^{(t)} A U_L^{(t)*}$; identified with operator $\mathcal{U}_L^{(t)} : \mathfrak{B}_L \to \mathfrak{B}_L$
$\boldsymbol{U}_L^{(t)}$	$L \times L$ real matrix representing $U_L^{(t)}$ in the $\{\phi_l\}$ basis of H_L
$\hat{U}_N^q : H_N \to H_N$	Shift operator
$U_{L,N}^{(q)} : H_{L,N} \to H_{L,N}$	Projected shift operator, $U_{L,N}^{(q)} = \mathbf{\Pi}_{L,N} \hat{U}_N^q$; identified with operator $U_{L,N}^{(q)} : H_{L,N} \to H_{L,N}$
$\mathcal{U}_{L,N}^{(q)} : \mathfrak{B} \to \mathfrak{B}$	Projected shift operator, $\mathcal{U}_{L,N}^{(q)} A = U_L^{(t)} A U_L^{(t)*}$; identified with operator $\mathcal{U}_{L,N}^{(q)} : \mathfrak{B}_{L,N} \to \mathfrak{B}_{L,N}$
$\boldsymbol{U}_{L,N}^{(q)}$	$L \times L$ real matrix representing $U_{L,N}^{(q)}$ in the $\{\phi_{l,N}\}$ basis of $H_{L,N}$
$W_{L,N} : H_{L,N} \to \mathbb{B}_\mathfrak{n}$	Unitary mapping into quantum computational Hilbert space
$\mathcal{W}_{L,N} : \mathfrak{B}_{L,N} \to \mathfrak{M}_\mathfrak{n}$	Induced unitary, $\mathcal{W}_{L,N} A = W_{L,N} A W_{L,N}^*$
X	Dynamical state space
$X_N \subset X$	Training trajectory $X_N = \{x_0, \dots, x_{N-1}\}$; training dynamical states $x_n = \Phi^{n\,\Delta t}(x_0)$
Y	Observations space; training observations $y_n = h(x_n)$
Z	Training data space; training data $z_n = z(x_n)$
$z : X \to Z$	Map into training data space
$\Gamma : S_*(\mathfrak{A}) \to S_*(\mathfrak{B})$	Embedding of normal states of $\mathfrak{A}$ into normal states of $\mathfrak{B}$
Δt	Sampling interval
δ_x	Dirac δ-measure supported at a point x
ϵ	Kernel bandwidth parameter
$\zeta \in \mathbb{B}_\mathfrak{n}$	Quantum computational state vector
$\eta_{\text{bump}} : \mathbb{R} \to \mathbb{R}$	Bump shape function
$\eta_{\text{gauss}} : \mathbb{R} \to \mathbb{R}$	Gaussian shape function
$\iota : C(X) \to H$	Map from continuous functions to $L^p(X, \mu)$ equivalence class
$\iota_N : C(X) \to H_N$	Map from continuous functions to $L^p(X, \mu_N)$ equivalence class
$\kappa : Z \times Z \to \mathbb{R}_+$	Kernel on training data space
λ_l	Kernel eigenvalue corresponding to ϕ_l
$\lambda_{l,N}$	Kernel eigenvalue corresponding to $\phi_{l,N}$
μ	Invariant measure
μ_N	Sampling measure supported on the trajectory X_N
ν_N	Pushforward of the sampling measure into data space, $\nu_N = z_*(\mu_N)$
$\xi \in H$	Unit vector associated with vector state $\omega_\rho \in S_*(\mathfrak{B})$, $\rho = \langle \xi, \cdot \rangle \xi$
$\xi_L \in H_L$	Unit vector associated with vector state $\omega_{\rho_L} \in S_*(\mathfrak{B})$, $\rho_L = \langle \xi_L, \cdot \rangle \xi_L$
$\xi_{L,N} \in H_{L,N}$	Unit vector associated with vector state $\omega_{\rho_{L,N}} \in S_*(\mathfrak{B})$, $\rho_{L,N} = \langle \xi_{L,N}, \cdot \rangle_N \xi_{L,N}$
$\boldsymbol{\xi}_{L,N} \in \mathbb{C}^L$	Column vector representation of $\xi_{L,N}$ in the $\{\phi_{l,N}\}$ basis of $H_{L,N}$
$\Pi_L : H \to H$	Projection onto H_L; identified with map $\Pi_L : H \to H_L$
$\mathbf{\Pi}_L : \mathfrak{B} \to \mathfrak{B}$	Projection onto $\mathfrak{B}_L$, $\mathbf{\Pi}_L A = \Pi_L A \Pi_L$; identified with map $\mathbf{\Pi}_L : \mathfrak{B} \to \mathfrak{B}_L$
$\Pi_{L,N} : \hat{H}_N \to \hat{H}_N$	Projection onto $H_{L,N}$; identified with map $\Pi_{L,N} : \hat{H}_N \to H_{L,N}$
$\mathbf{\Pi}_{L,N} : \hat{\mathfrak{B}}_N \to \hat{\mathfrak{B}}_N$	Projection onto $\mathfrak{B}_{L,N}$, $\mathbf{\Pi}_{L,N} A = \Pi_{L,N} A \Pi_{L,N}$; identified with map $\mathbf{\Pi}_{L,N} : \hat{\mathfrak{B}}_N \to \mathfrak{B}_{L,N}$
$\pi : \mathfrak{A} \to \mathfrak{B}$	Regular representation of $\mathfrak{A}$
$\tilde{\pi} : C(X) \to \mathfrak{C}$	Regular representation of $C(X)$
$\pi_L : \mathfrak{A} \to \mathfrak{B}_L$	Projected regular representation, $\pi_L = \mathbf{\Pi}_L \circ \pi$
$\hat{\pi}_N : \hat{\mathfrak{A}}_N \to \hat{\mathfrak{B}}_N$	Regular representation of $\hat{\mathfrak{A}}_N$
$\pi_{L,N} : \mathfrak{A}_N \to \mathfrak{B}_{L,N}$	Projected regular representation, $\pi_{L,N} = \mathbf{\Pi}_{L,N} \circ \hat{\pi}_N$
τ	Forecast lead time
$\Phi^t : X \to X$	Dynamical flow
$\phi_l \in H$	Eigenvectors of K (basis vectors for H and H_L)
$\varphi_l \in C(X)$	Continuous representative of ϕ_l
$\phi_{l,N} \in \hat{H}_N$	Eigenvectors of K_N (basis vectors for $\hat{H}_N$ and $H_{L,N}$)
$\varphi_{l,N} \in C(X)$	Continuous representative of $\phi_{l,N}$
$\boldsymbol{\phi}_{l,N} \in \mathbb{R}^N$	Column vector representation of $\phi_{l,N}$
χ_S	Characteristic function of a set S
$\psi : Y \times Y \to [0,1]$	Kernel on observation space
$\psi_N : Y \times Y \to [0,1]$	Data-dependent kernel on observation space
$\omega_p \in S_*(\mathfrak{A})$	Normal state induced by probability density $p \in \mathfrak{A}_*$
$\omega_\rho \in S_*(\mathfrak{B})$	Normal state induced by density operator $\rho \in \mathfrak{B}_*$
$\omega_{\rho_L} \in S_*(\mathfrak{B}_L)$	Normal state induced by density operator $\rho_L \in \mathfrak{B}_{L,*}$; extends to normal state $\omega_{\rho_L} \in S_*(\mathfrak{B}_L)$
$\omega_{\rho_{L,N}} \in S_*(\mathfrak{B}_{L,N})$	Normal state induced by density operator $\rho_{L,N} \in \mathfrak{B}_{L,N}$
†	Hermitian conjugate (complex conjugate transpose)
$\cdot$	Normalized Euclidean inner product on $\mathbb{C}^N$, $\boldsymbol{f} \cdot \boldsymbol{g} = \boldsymbol{f}^\dagger \boldsymbol{g} / N$
$\odot$	Elementwise array multiplication
$\langle \cdot, \cdot \rangle$	Inner product of H
$\langle \cdot, \cdot \rangle_N$	Inner product of $\hat{H}_N$
$\langle \cdot, \cdot \rangle_\mathfrak{n}$	Inner product of $\mathfrak{M}_\mathfrak{n}$

Table S3. Dataset attributes and QMDA parameters for the L96 multiscale and CCSM4 ENSO experiments.

	L96 Multiscale	**ENSO**
Dataset attributes		
Number of training samples N	40,000	15,600
Number of test samples $\hat{N}$	13,200	2400
Delay parameter Q	0	5
Observation space (Y) dimension d	9	44,414
Training data space (Z) dimension d_Z	9	488,554
Kernel κ for basis computation		
Kernel type	Eq. (S4)	Eq. (S5)
Neighborhood parameter k_{nn} for bandwidth function	400	400
Bandwidth range parameters (J_1, J_2)	$(-40, 40)$	$(-40, 40)$
Bandwidth exponent parameter	$(J_2 - J_1)/200$	$(J_2 - J_1)/200$
Bandwidth scaling parameter s_κ	1	2
Kernel ψ for data assimilation		
Kernel type	Eq. (S25)	Eq. (S25)
Neighborhood parameter k_{nn} for bandwidth function	400	400
Bandwidth range parameters (J_1, J_2)	$(-40, 40)$	$(-40, 40)$
Bandwidth exponent parameter	$(J_2 - J_1)/500$	$(J_2 - J_1)/500$
Bandwidth scaling parameter s_ψ	0.6	1
Data assimilation		
Number of basis functions L (Hilbert space dimension)	2000	1000
Number of timesteps J_{o} per observation	1	1
Number of forecast timesteps J_{f}	150	12
Quantum circuit simulation		
Number of qubits $\mathfrak{n}$	10	N/A
Number of shots M	10^6	N/A

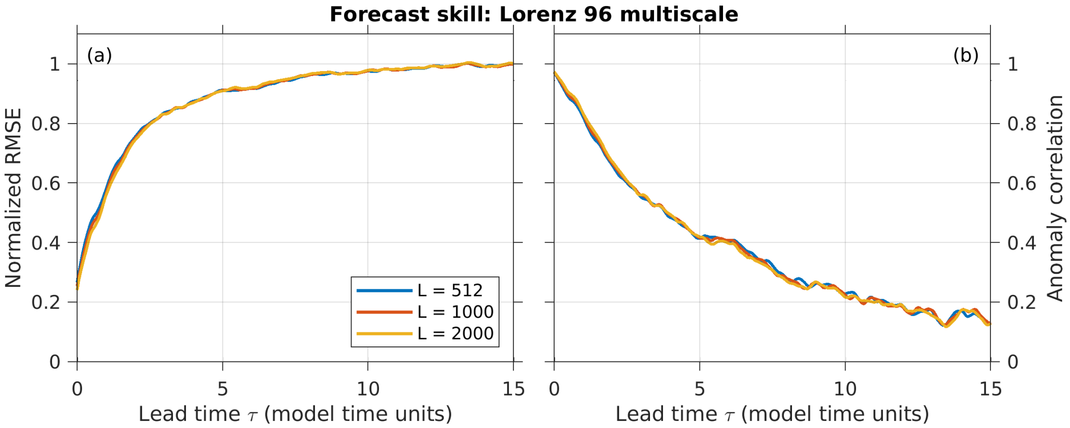

Fig. S1. NRMSE (a) and AC (b) skill scores for forecasts of the x_1 variable of the L96 multiscale system, obtained for representative values of the Hilbert space dimension parameter L in the range 512–2000. The remaining QMDA parameters are listed in Table S3. The case $L = 2000$ corresponds to Figs. 2 and 3 in the main text.

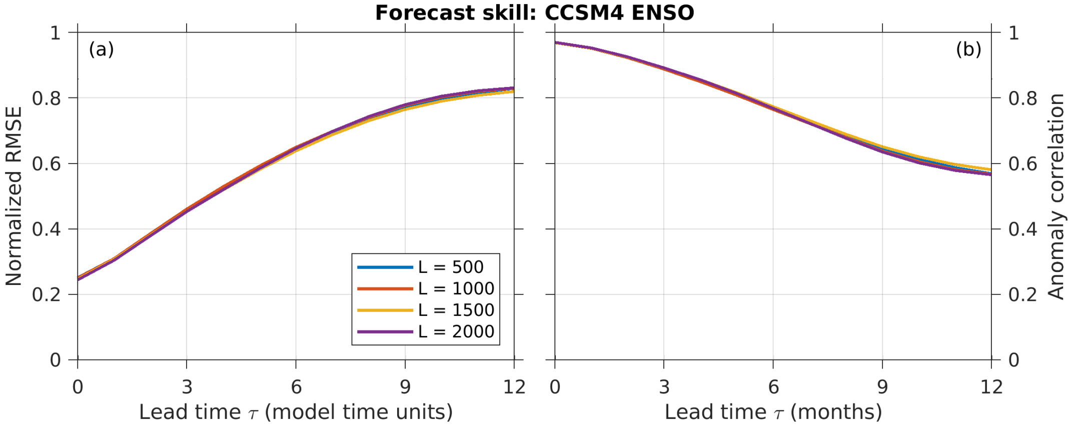

Fig. S2. NRMSE (a) and AC (b) skill scores for forecasts of the Niño 3.4 index in CCSM4, obtained for representative values of the Hilbert space dimension parameter in the range 500–2000. The remaining QMDA parameters are listed in Table S3. The case $L = 1000$ corresponds to Figs. 4 and 5 in the main text.

1. Assumptions

We make the following standing assumption on the dynamical system and forecast observable.

Assumption 1.

(a) $\Phi^t : X \to X$, $t \in \mathbb{R}$, is a continuous-time, continuous flow, on a compact metrizable space X.
(b) μ is an invariant, ergodic, Borel probability measure under Φ^t.
(c) The forecast observable $f : X \to \mathbb{R}$ is a real-valued function lying in $\mathfrak{A} = L^\infty(X, \mu)$.

Note that the support of μ is a closed subset of the compact space X, and thus is compact. Moreover, the compactness assumption on X can be replaced by the weaker assumption that Φ^t has a forward-invariant compact set X_+ that contains the support of μ (which is again necessarily compact). The analysis performed below can be carried over to this setting by replacing the space of continuous functions $C(X)$ (which is a Banach space equipped with the uniform norm when X is compact) with $C(X_+)$.

For the purpose of data-driven approximation, we additionally require:

Assumption 2.

(a) For the sampling interval $\Delta t > 0$, the discrete-time system induced by the map $\Phi^{\Delta t} : X \to X$ is ergodic with respect to μ.
(b) The forecast observable $f : X \to \mathbb{R}$ is continuous.
(c) The observation map $h : X \to Y$ is continuous.

Assumption 2(a) implies that for μ-a.e. initial condition $x_0 \in X$, the sampling measures $\mu_N = \sum_{n=0}^{N-1} \delta_{x_n}/N$ with $x_n = \Phi^{n\,\Delta t}(x_0)$ weak-* converge to the invariant measure μ; that is,

$$\lim_{N\to\infty} \int_X f\, d\mu_N = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} f(x_n) = \int_X f\, d\mu, \quad \forall f \in C(X). \qquad \text{[S1]}$$

Henceforth, we will assume for convenience that the states $x_0, x_1, \dots$ are all distinct—aside from the trivial case that the support of μ is a singleton set consisting of a fixed point, this assumption holds for μ-a.e. initial condition x_0, and ensures that the Hilbert space $\hat{H}_N = L^2(X, \mu_N)$ has dimension N.

In what follows,

$$\langle f, g\rangle = \int_X f^* g\, d\mu, \quad \langle f, g\rangle_N = \int_X f^* g\, d\mu_N = \frac{1}{N}\sum_{n=0}^{N-1} f^*(x_n) g(x_n)$$

will denote the inner products of H and $\hat{H}_N$, respectively. The Hilbert space $\hat{H}_N$ is isomorphic to $\mathbb{C}^N$ equipped with the normalized dot product $\boldsymbol{f} \cdot \boldsymbol{g} \equiv \boldsymbol{f}^\dagger \boldsymbol{g}/N$, where $\boldsymbol{f}^\dagger$ is the Hermitian conjugate (complex conjugate transpose) of the column vector $\boldsymbol{f} \in \mathbb{C}^N$. Under this isomorphism, two elements $f, g \in \hat{H}_N$ are represented by column vectors $\boldsymbol{f} = (f(x_0), \dots, f(x_{N-1}))^\top$ and $\boldsymbol{g} = (g(x_0), \dots, g(x_{N-1}))^\top$, and we have $\langle f, g\rangle_N = \boldsymbol{f} \cdot \boldsymbol{g}$.

2. Finite-dimensional approximation

This section provides an overview and pseudocode listings of the data-driven approximation techniques underpinning QMDA. We begin with Algorithm S1, which gives a high-level description of the QMDA pipeline employed in the L96 multiscale and ENSO experiments presented in the main text. The algorithm is divided up into two parts:

1. A training phase, which uses the training data $y_0, \dots, y_{N-1} \in Y$ and $f_0, \dots, f_{N-1} \in \mathbb{R}$ for the observation map h and forecast observable f, respectively, to build an orthonormal basis $\{\phi_{0,N}, \dots, \phi_{L-1,N}\}$ of the Hilbert space $H_{L,N}$. The basis is used to approximate the Koopman operator U^t of the dynamical system, the multiplication operator πf representing the forecast observable, and the effect-valued map $\mathcal{F}$ employed in the analysis step.
2. A prediction phase, which iteratively executes the sequential forecast–analysis steps of QMDA given a test dataset of observations $\hat{y}_0, \dots, \hat{y}_{\hat{N}-1} \in Y$. The state of the data assimilation system at time t_n is a vector state of the operator algebra $\mathfrak{B}_{L,N}$, induced by a unit vector $\xi_n \in H_{L,N}$. This vector is represented in the $\{\phi_{l,N}\}$ basis of $H_{L,N}$ by a column vector $\boldsymbol{\xi}_n \in \mathbb{C}^L$.

Algorithm S1 depends on a number of lower-level procedures, which we describe in the following subsections.

A. Kernel eigenfunctions. Following refs. (1–5), we use eigenvectors of kernel integral operators to construct both the L-dimensional Hilbert spaces H_L and their data-driven counterparts $H_{L,N}$. We make the following assumptions on the kernels used to define these operators.

Assumption 3.

(a) $k : X \times X \to \mathbb{R}$ is a continuous, symmetric kernel.
(b) $k_0, k_1, \dots : X \times X \to \mathbb{R}$ is a family of continuous, symmetric kernels such that, as $N \to \infty$, k_N converges uniformly to k.

Algorithm S1 QMDA pipeline employed in the L96 and ENSO experiments described in the main text.

Inputs

1. Delay embedding parameter $Q \in \mathbb{N}$.
2. Hilbert space dimension $L \in \mathbb{N}$.
3. Number of spectral bins $M \in \mathbb{N}$.
4. Kernel neighborhood parameter k_{nn} in $\mathbb{N}$.
5. Bandwidth exponent parameter $a > 0$ and range parameters $J_1, J_2 \in \mathbb{N}$.
6. Number of forecast timesteps $J_{\text{f}} \in \mathbb{N}$.
7. Training data from observation map, $y_{-Q}, \dots, y_{N-1+Q} \in Y \equiv \mathbb{R}^d$ with $y_n = h(x_n)$.
8. Training data from forecast observable, $f_0, \dots, f_{N-1+Q} \in \mathbb{R}$ with $f_n = f(x_n)$.
9. Observed data $\hat{y}_0, \dots, \hat{y}_{\hat{N}-1} \in Y$ in test period.

Require: All training data are induced by the same sequence of (unknown) time-ordered states $x_{-Q}, \dots, x_{N-1+Q} \in X$ with $x_n = \Phi^{n\,\Delta t}(x_0)$, taken at a fixed sampling interval $\Delta t > 0$.

Outputs

1. Mean forecast $\{\bar{f}_{nj}\}$ for $n \in \{0, \dots, \hat{N}-1\}$ and $j \in \{0, \dots, J_{\text{f}}\}$, where $\bar{f}_{nj}$ has initialization time $t_n = n\,\Delta t$ in the test period and lead time $\tau_j = j\,\Delta t$.
2. Forecast uncertainty $\{\sigma_{nj}\}$ for $n \in \{0, \dots, \hat{N}-1\}$ and $j \in \{0, \dots, J_{\text{f}}\}$, where σ_{nj} has initialization time t_n and lead time τ_j.
3. Spectral bins (intervals) $S_0, \dots, S_{M-1} \subseteq \mathbb{R}$.
4. Forecast probability vectors $\{\boldsymbol{p}_{nj}\}$ for $n \in \{0, \dots, \hat{N}-1\}$ and $j \in \{0, \dots, J_{\text{f}}\}$, where $\boldsymbol{p}_{nj} = (p_{0nj}, \dots, p_{M-1,nj})$ is the probability that, for initialization time t_n and lead time τ_j, the forecast observable f lies in S_m.

Training phase

1. Apply Eq. (S6) to the training data y_n to build the delay-embedded dataset $z_0, \dots, z_{N-1} \in Z \equiv \mathbb{R}^{(2Q+1)d}$.
2. Set d_Z to the Euclidean distance on Z. Execute Algorithm S2 with inputs $\{z_n\}_{n=0}^{N-1}$, d_Z, and k_{nn} to obtain a kernel bandwidth function $b_Z : Z \to \mathbb{R}_+$.
3. Execute Algorithm S3 with inputs $\{z_n\}_{n=0}^{N-1}$, d_Z, and k_{nn} to obtain basis vectors $\boldsymbol{\phi}_0, \dots, \boldsymbol{\phi}_{L-1} \in \mathbb{R}^N$ for $H_{L,N}$.
4. Execute Algorithm S8 with inputs $\{f_n\}_{n=0}^{N-1}$ to obtain spectral bins $S_0, \dots, S_{M-1} \subset \mathbb{R}$.
5. Execute Algorithm S7 with inputs $\{f_n\}_{n=0}^{N-1}$, $\{\boldsymbol{\phi}_l\}_{l=0}^{L-1}$, and $\{S_m\}_{m=0}^{M-1}$ to obtain the projected multiplication operator $\boldsymbol{A} \in \mathbb{M}_L$ representing f and spectral projectors $\boldsymbol{E}_0, \dots, \boldsymbol{E}_{M-1} \in \mathbb{M}_L$.
6. For each $j \in \{1, \dots, J_{\text{f}}\}$, execute Algorithm S9 with inputs j and $\{\boldsymbol{\phi}_l\}_{l=0}^{L-1}$ to obtain Koopman operator matrices $\boldsymbol{U}^{(1)}, \dots, \boldsymbol{U}^{(J_{\text{f}})} \in \mathbb{M}_L$.
7. Set d_Y to the Euclidean distance on Y. Execute Algorithm S2 with inputs $\{y_n\}_{n=0}^{N-1}$, d_Y, and k_{nn} to obtain a kernel bandwidth function $b_Y : Y \to \mathbb{R}_+$.
8. Define the scaled distance function $\tilde{d}_Y : Y \times Y \to \mathbb{R}_+$ with $\tilde{d}_Y(y, y') = d_Y(y, y')/\sqrt{b_Y(y) b_Y(y')}$. Execute Algorithm S2 with inputs $\{y_n\}_{n=0}^{N-1}$, a, J_1, J_2, $\tilde{d}_Y$, and η_{bump} (where η_{bump} is the bump function from Eq. (S26)) to obtain an optimal bandwidth parameter ϵ_*.
9. Define the kernel function $\psi : Y \times Y \to [0, 1]$ with $\psi(y, y') = \eta_{\text{bump}}(\tilde{d}_Y(y, y')/\epsilon_*)$. Execute Algorithm S12 with inputs ψ, $\{y_n\}_{n=0}^{N-1}$ and $\{\boldsymbol{\phi}_l\}_{l=0}^{L-1}$ to obtain the effect-valued feature map $\boldsymbol{F} : Y \to \mathbb{M}_L$.

Prediction phase

1. Set the initial state vector $\boldsymbol{\xi}_0 = (1, 0, \dots, 0)^\top \in \mathbb{C}^L$.
2. For each $n \in \{1, \dots, \hat{N}-1\}$ execute the forecast–analysis cycle in Algorithm S11 with inputs J_{f}, $J_{\text{o}} = 1$, $\{\boldsymbol{U}^{(j)}\}_{j=1}^{J_{\text{f}}}$, $\boldsymbol{A}$, $\{\boldsymbol{E}_m\}_{m=0}^{M-1}$, $\boldsymbol{F}$, $\boldsymbol{\xi}_{n-1}$, and $\hat{y}_n$.

   Return:

   - The mean forecasts $\bar{f}_{n-1,0}, \dots, \bar{f}_{n-1,J_{\text{f}}}$.
   - The forecast uncertainties $\sigma_{n-1,0}, \dots, \sigma_{n-1,J_{\text{f}}}$.
   - The forecast probability vectors $\boldsymbol{p}_{n-1,0}, \dots, \boldsymbol{p}_{n-1,J_{\text{f}}}$.
   - The posterior state vector $\boldsymbol{\xi}_n$ given the observation y_n.

As we describe below, the kernels k_N are typically data-dependent kernels obtained by normalization of a fixed (data-independent) kernel on X.

Defining $K : H \to H$ as the integral operator

$$Kf = \int_X k(\cdot, x) f(x)\, d\mu(x),$$

we have that K is a real, self-adjoint, Hilbert-Schmidt operator, and thus there exists a real, orthonormal basis $\{\phi_0, \phi_1, \dots\}$ of $H = L^2(X, \mu)$ consisting of eigenvectors of K,

$$K\phi_l = \lambda_l \phi_l, \quad \lambda_l \in \mathbb{R}, \tag{S2}$$

where the eigenvalues $\lambda_0, \lambda_1, \dots$ are ordered in order of decreasing absolute value and converge to 0 as $l \to \infty$. In the data-driven setting, we replace H by the N-dimensional Hilbert space $\hat{H}_N$, and define the integral operator $K_N : \hat{H}_N \to \hat{H}_N$ as

$$K_N f = \int_X k_N(\cdot, x) f(x)\, d\mu_N(x) = \frac{1}{N} \sum_{n=0}^{N-1} k_N(\cdot, x_n) f(x_n).$$

The operator K_N has an associated real, orthonormal eigenbasis $\{\phi_{0,N}, \dots, \phi_{N-1,N}\}$ of $\hat{H}_N$, where

$$K_N \phi_{l,N} = \lambda_{l,N} \phi_{l,N}, \quad \lambda_{l,N} \in \mathbb{R}, \tag{S3}$$

and the eigenvalues $\lambda_{l,N}$ are ordered again in order of decreasing modulus.

An important property of the eigenvectors ϕ_l and $\phi_{l,N}$ corresponding to nonzero eigenvalues is that they have continuous representatives. Specifically, assuming that λ_l and $\lambda_{l,N}$ are nonzero, we define $\varphi_l, \varphi_{l,N} \in C(X)$ such that

$$\varphi_l(x) = \frac{1}{\lambda_l} \int_X k(x, x') \phi_l(x')\, d\mu(x'), \quad \varphi_{l,N}(x) = \frac{1}{\lambda_{l,N}} \int_X k_N(x, x') \phi_{l,N}(x')\, d\mu_N(x').$$

It then follows from Eq. (S2) and Eq. (S3), respectively, that $\varphi_l = \phi_l$ μ-a.e. and $\varphi_{l,N} = \phi_{l,N}$ μ_N-a.e. Note that the latter relation simply means that $\varphi_{l,N}(x_n) = \phi_{l,N}(x_n)$ for every $n \in \{0, \dots, N-1\}$.

The following theorem summarizes the spectral convergence of the operators K_N to K and the convergence of the associated basis functions. The results are based on techniques developed in ref. (6). Additional details and proofs for the setting of ergodic dynamical systems and data-dependent kernels employed in this work can be found, e.g., in refs. (5, 7).

Theorem 1. *Under Assumptions 1–3, the following hold as $N \to \infty$ for μ-a.e. initial condition $x_0 \in X$.*

(a) For each nonzero eigenvalue λ_l of K, the sequence of eigenvalues $\lambda_{l,N}$ of K_N converges to λ_l, including multiplicities.

(b) If $\phi_l \in H$ is an eigenvector of K corresponding to λ_l with continuous representative $\varphi_l \in C(X)$, there exists a sequence of eigenvectors $\phi_{l,N}$ of K_N corresponding to eigenvalue $\lambda_{l,N}$, whose continuous representatives $\varphi_{l,N} \in C(X)$ converge uniformly to φ_l.

In numerical applications, we use the $\hat{H}_N \simeq \mathbb{C}^N$ isomorphism to represent the eigenvectors $\phi_{l,N}$ by N-dimensional column vectors $\boldsymbol{\phi}_{l,N} \in \mathbb{C}^N$ (which are real since the $\phi_{l,N}$ are real) with $\boldsymbol{\phi}_{l,N} = (\phi_{l,N}(x_0), \dots, \phi_{l,N}(x_{N-1}))^\top$. The vectors $\boldsymbol{\phi}_{l,N}$ are solutions of the eigenvalue problem

$$\boldsymbol{K}_N \boldsymbol{\phi}_{l,N} = \lambda_{l,N} \boldsymbol{\phi}_{l,N}$$

for the $N \times N$ kernel matrix $\boldsymbol{K}_N = [K_{ij,N}]_{i,j=0}^{N-1}$ with $K_{ij,N} = k_N(x_i, x_j)/N$. We impose the orthonormality condition $\boldsymbol{\phi}_{l,N} \cdot \boldsymbol{\phi}_{m,N} = \delta_{l,m}$, which is equivalent to $\langle \phi_{l,N}, \phi_{m,N} \rangle_N = \delta_{lm}$ on $\hat{H}_N$.

Henceforth, we will assume that for a given choice of basis vectors ϕ_l of H and so long as λ_l is nonzero, the data-driven basis vectors $\phi_{l,N}$ of $\hat{H}_N$ are chosen such that they converge to ϕ_l as per Theorem 1. This assumption leads to no loss of generality since every real, orthonormal eigenbasis $\phi_{l,N}$ can be orthogonally rotated to a basis that converges to ϕ_l without affecting the results of the computations presented below.

B. Choice of kernel. Since our training data z_n are in the space Z, we employ kernels which are pullbacks of kernels on that space; specifically, we set $k(x, x') = \kappa(z(x), z(x'))$ and $k_N(x, x') = \kappa_N(z(x), z(x'))$, where $\kappa : Z \times Z \to \mathbb{R}$ and $\kappa_N : Z \times Z \to \mathbb{R}$ are continuous, symmetric kernels. With this approach, all kernel computations can be executed using the data $z_n \in Z$ without knowledge of the underlying dynamical states $x_n \in X$.

Following ref. (5), we construct the kernels κ_N by applying the bistochastic normalization procedure introduced in ref. (8) to the family of variable-bandwidth diffusion kernels developed in ref. (9). Using the training data z_n, we construct a variable-bandwidth radial basis function kernel $\tilde{\kappa}_N : Z \times Z \to \mathbb{R}$ of the form

$$\tilde{\kappa}_N(z, z') = \eta_{\text{gauss}}\left(\frac{d(z, z')}{\epsilon_N \sqrt{b_N(z) b_N(z')}} \right), \tag{S4}$$

where $\eta_{\text{gauss}} : \mathbb{R} \to \mathbb{R}$ is the Gaussian shape function, $\eta_{\text{gauss}}(u) = e^{-u^2}$, $d : Z \to Z \to \mathbb{R}_+$ is a distance function (which we nominally set to the Euclidean when $Z = \mathbb{R}^d$), $\epsilon_N > 0$ is a bandwidth parameter, and $b_N : Z \to \mathbb{R}_+$ is a (data-dependent) bandwidth function. The construction of the bandwidth function, which resembles a kernel density estimation procedure, is summarized in Algorithm S2. The bandwidth parameter ϵ_N is tuned automatically using Algorithm S6, which is described in section 2.C below. In order to have flexibility to adjust the bandwidth parameter around the value computed by the tuner, we introduce a scaling parameter $s_\kappa > 0$ and set $\epsilon_N = s_\kappa \epsilon_*$, where ϵ_* is the output of Algorithm S6. In general, we find that the automatic tuning procedure performs well, so $s_\kappa = 1$ is typically a good choice, but in some cases involving datasets of large intrinsic dimension and sparse sampling, using a somewhat more conservative bandwidth parameter (e.g., $s_\kappa = 2$) can lead to higher-quality basis functions. Further details on Algorithms S2 and S6 can be found in refs. (1, 9).

Algorithm S2 Kernel bandwidth function.

Inputs

1. Dataset $x_0, x_1, \dots, x_{N-1} \in \mathcal{X}$; $\mathcal{X}$ is an arbitrary set.
2. Distance function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$.
3. Neighborhood parameter $k_{\text{nn}} \in \mathbb{N}$.
4. Bandwidth exponent parameter $a > 0$ and range parameters $J_1, J_2 \in \mathbb{N}$.

Outputs

1. Bandwidth function $b : \mathcal{X} \to \mathbb{R}_+$.

Steps

1. Construct the function $r : \mathcal{X} \to \mathbb{R}_+$, where $r^2(x) = \sum_{j=1}^{k_{\text{nn}}} d^2(x, I(x,j))/k_{\text{nn}}$ and $I(x,j) \in \{0, \dots, N-1\}$ is the index of the j-th nearest neighbor of x in the $\{x_i\}_{i=0}^{N-1}$ dataset with respect to the distance d.
2. Construct the distance-like function $\tilde{d} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with $\tilde{d}(x,x') = d(x,x')/\sqrt{r(x)r(x')}$.
3. Execute Algorithm S6 with inputs $\{x_n\}_{n=0}^{N-1}$, $\tilde{d}$, η_{gauss}, k_{nn}, a, J_1, and J_2 to obtain an optimal bandwidth ϵ_* and dimension estimate m_*.
4. Construct the kernel $\tilde{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, where $\tilde{k}(x,x') = \eta_{\text{gauss}}(\tilde{d}(x,x')/\epsilon_*)$.
5. **Return:** The function $b : \mathcal{X} \to \mathbb{R}_+$ such that $b(x) = \sum_{j=0}^{N-1} \tilde{k}(x,x_j)/(N(\pi\epsilon_* r^2(x))^{m_*/2})$.

Using the kernel $\tilde{\kappa}_N$, we perform the sequence of normalization steps described in ref. (8) to obtain a symmetric, positive, positive-definite kernel κ_N which is Markovian with respect to the pushforward $\nu_N := z_*(\mu_N)$ of the sampling measure on Z,

$$\int_Z \kappa_N(z,z')\, d\nu_N(z') = \frac{1}{N}\sum_{n=0}^{N-1} \kappa_N(z,z_n) = 1, \quad \forall z \in Z.$$

Algorithm S3 describes the computation of the eigenvectors $\boldsymbol{\phi}_{l,N}$ associated with this kernel. We note that due to the particular form of the normalization leading to κ_N, the eigenvectors $\boldsymbol{\phi}_{l,N}$ can be computed without explicit formation of the $N \times N$ kernel matrix $\boldsymbol{K}_N$. Instead, we compute the $\boldsymbol{\phi}_{l,N}$ through the singular value decomposition (SVD) of an $N \times N$ kernel matrix $\hat{\boldsymbol{K}}_N = [\hat{\kappa}_N(z_i,z_j)]_{i,j=0}^{N-1}$ associated with a non-symmetric kernel function $\hat{\kappa}_N : Z \times Z \to \mathbb{R}$ that factorizes $\boldsymbol{K}_N$ as $\boldsymbol{K}_N = N\hat{\boldsymbol{K}}_N\hat{\boldsymbol{K}}_N^\top$. As is common practice in kernel methods, when the kernel matrix $\hat{\boldsymbol{K}}_N$ is too large for our available computational resources, we fix a parameter $\hat{k}_{\text{nn}} \in \mathbb{N}$ and approximate $\hat{\boldsymbol{K}}_N$ by a sparse matrix obtained by setting the $N - \hat{k}_{\text{nn}}$ smallest values in each row of $\hat{\boldsymbol{K}}_N$ to 0. The steps leading to $\tilde{\kappa}_N$ are listed in Algorithm S4. See Appendix B in ref. (5) for further details.

The data-driven basis from Algorithm S3 was used in our L96 multiscale experiments. As noted in the main text, in the ENSO experiments we used a modified version of the algorithm that replaces $\tilde{\kappa}_N$ from Eq. (S4) with a product kernel that captures covariability between data in Z and data in a response space $\hat{Y}$ (in the case of ENSO, a sequence space of Niño 3.4 indices). Specifically, given a function $\hat{h} : Z \to \hat{Y}$, we define $\tilde{\kappa}_{\hat{h},N} : Z \times Z \to \mathbb{R}$ as

$$\tilde{\kappa}_{\hat{h},N}(z,z') = \eta_{\text{gauss}}\left(\frac{d(z,z')}{\epsilon_N\sqrt{b_N(z)b_N(z')}}\right)\eta_{\text{gauss}}\left(\frac{d_{\hat{Y}}(\hat{h}(z),\hat{h}(z'))}{\epsilon_{\hat{Y},N}\sqrt{b_{\hat{Y},N}(\hat{h}(z))b_{\hat{Y},N}(\hat{h}(z'))}}\right), \qquad \text{[S5]}$$

where $d_{\hat{Y}} : \hat{Y} \times \hat{Y} \to \mathbb{R}_+$ is a distance function on $\hat{Y}$, $\hat{\epsilon}_{\hat{Y},N}$ is a bandwidth parameter, and $b_{\hat{Y},N} : \hat{Y} \to \mathbb{R}_+$ is a bandwidth function. The bandwidth function $b_{\hat{Y}}$ is computed via Algorithm S2 using $\{\hat{h}(z_0), \dots, \hat{h}(z_N)\}$ as input data, and $\epsilon_{\hat{Y},N}$ is tuned via Algorithm S6. Once the kernel function $\tilde{\kappa}_{\hat{h},N}$ is formed, the computation of the associated bistochastic kernel and eigenfunctions proceeds analogously to Algorithm S3. We summarize the entire procedure in Algorithm S5 for completeness.

In addition to the bistochastic kernel from Algorithms S3 and S5, QMDA can be implemented with a variety of kernels, including non-symmetric kernels satisfying a detailed-balance condition (e.g., the family of normalized kernels from the diffusion maps algorithm (10)). Two basic guidelines on kernel choice are that the data-dependent kernels k_N are regular-enough such that the integral operators K_N converge spectrally to K (in the sense of Theorem 1), and the limit kernel k is "rich-enough" such that all eigenvalues λ_l are strictly positive (i.e., k is an $L^2(\mu)$ integrally strictly-positive kernel (11)). In that case, as N and L increase, the eigenvectors $\phi_{l,N}$ provide a consistent approximation of an orthonormal basis for the entire Hilbert space

Algorithm S3 Orthonormal basis vectors of $\hat{H}_N$ from variable-bandwidth bistochastic kernel. We suppress N subscripts from our notation for $\boldsymbol{\phi}_{l,N}$.

```
Inputs
  1. Dataset $z_0, z_1, \dots, z_{N-1} \in Z$.
  2. Distance function $d : Z \times Z \to \mathbb{R}_+$.
  3. Neighborhood parameters $k_{\text{nn}}, \hat{k}_{\text{nn}} \in \mathbb{N}$.
  4. Bandwidth exponent parameter $a > 0$ and range parameters $J_1, J_2 \in \mathbb{N}$.
  5. Bandwidth scaling parameter $s_\kappa > 0$.
  6. Number of basis vectors $L \leq N$.
Outputs
  1. Column vectors $\boldsymbol{\phi}_0, \dots, \boldsymbol{\phi}_{L-1} \in \mathbb{R}^N$.
Steps
  1. Execute Algorithm S2 with inputs $\{z_n\}_{n=0}^{N-1}$, $d$, $a$, $J_1$, and $J_2$ to obtain a bandwidth function $b : Z \to \mathbb{R}_+$.
  2. Construct the distance-like function $\tilde{d} : Z \times Z \to \mathbb{R}_+$ with $\tilde{d}(z, z') = d(z, z')/\sqrt{b(z)b(z')}$.
  3. Execute Algorithm S6 with inputs $\{z_n\}_{n=0}^{N-1}$, $\tilde{d}$, $\eta_{\text{gauss}}$, $a$, $J_1$, and $J_2$ to obtain an optimal bandwidth $\epsilon_*$.
  4. Construct the kernel $\tilde{\kappa} : Z \times Z \to \mathbb{R}_+$ with $\tilde{\kappa}(z, z') = \eta_{\text{gauss}}(\tilde{d}(z, z')/(s_\kappa \epsilon_*))$.
  5. Execute Algorithm S4 with inputs $\{z_n\}_{n=0}^{N-1}$ and $\tilde{\kappa}$ to obtain a non-symmetric kernel function $\hat{\kappa} : Z \times Z \to \mathbb{R}_+$.
  6. Form the $N \times N$ kernel matrix $\hat{\boldsymbol{K}} = [\hat{K}_{ij}]_{i,j=0}^{N-1}$ with $\hat{K}_{ij} = \hat{\kappa}(z_i, z_j)$.
  7. If $\hat{k}_{\text{nn}} < N$, set the $N - k_{\text{nn}}$ smallest elements in each row of $\hat{\boldsymbol{K}}$ to 0 and use a sparse array to store $\hat{\boldsymbol{K}}$.
  8. Return: The leading $L$ left singular vectors $\boldsymbol{\phi}_0, \dots, \boldsymbol{\phi}_{L-1}$ of $\hat{\boldsymbol{K}}$, arranged in order of decreasing corresponding singular value, and normalized such that $\|\boldsymbol{\phi}_l\|_2 = \sqrt{N}$.
```

Algorithm S4 Factorization of bistochastic kernel function.

```
Inputs
  1. Dataset $x_0, x_1, \dots, x_{N-1} \in \mathcal{X}$; $\mathcal{X}$ is an arbitrary set.
  2. Kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$.
Outputs
  1. Non-symmetric kernel function $\hat{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$.
Steps
  1. Construct the degree function $d : \mathcal{X} \to \mathbb{R}_+$, where $d(x) = \sum_{j=0}^{N-1} k(x, x_j)$.
  2. Construct the function $q : \mathcal{X} \to \mathbb{R}_+$, where $q(x) = \sum_{j=0}^{N-1} k(x, x_j)/d(x_j)$.
  3. Return: The kernel function $\hat{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$, where $\hat{k}(x, x') = k(x, x')/(d(x)q^{1/2}(x'))$.
```

H. The bistochastic kernels k_N from Algorithm S3 have this property if the map $z : X \to Z$ is injective. In the case of the delay coordinate map $z : X \to Z = Y^{2Q+1}$,

$$z(x) = (h(\Phi^{-Q\,\Delta t}(x)), h(\Phi^{(-Q+1)\,\Delta t}(x)), \dots, \Phi^{Q\,\Delta t}(x)) \qquad \text{[S6]}$$

injectivity holds for sufficiently large delay parameter Q under appropriate assumptions on delay-coordinate maps (12).

For certain classes of kernels k constructed from shape functions with rapid decay (e.g., the Gaussian shape function η_{gauss}), the asymptotic behavior of the eigenfunctions in the limit of vanishing bandwidth parameter ϵ_N may be studied using the theory of heat kernels (13). Under appropriate conditions (e.g., the support of the invariant measure μ is a differentiable manifold or a metric measure space), the eigenfunctions are extremizers of a Dirichlet energy induced by the kernel, which defines a notion of regularity of functions akin to a Sobolev norm. In such cases, for any given $L \in \mathbb{N}$, the set of orthonormal vectors $\{\phi_0, \dots, \phi_{L-1}\}$ (which we will use in section 2.D to define the subspaces $H_L \subset H$ used in QMDA) is optimal in the sense of having maximal regularity with respect to the kernel-induced Dirichlet energy.

C. Bandwidth tuning. Algorithm S6 is a tuning procedure for bandwidth-dependent kernels $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ of the form $k(x, x') = \kappa(d(x, x')/\epsilon)$, where $\mathcal{X}$ is an arbitrary set, $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ is a distance-like function, $\eta : \mathbb{R}_+ \to \mathbb{R}_+$ a positive kernel shape function, and $\epsilon > 0$ a kernel bandwidth parameter. The tuning approach in Algorithm S6 was proposed in ref. (14) using scaling arguments for heat-like kernels on manifolds, and was also used in refs. (1, 4, 9). It takes as input a dataset in $\mathcal{X}$ and a logarithmic grid of candidate bandwidth values ϵ_j, and returns an "optimal" bandwidth ϵ_* from this candidate set that maximizes a kernel-induced dimension function $m(\epsilon_j)$ for the dataset. If k is a heat-like kernel on a Riemannian manifold, $m(\epsilon_*)$ is an estimator of the manifold's dimension, but $m(\epsilon_*)$ also provides a notion of dimension for non-smooth sets.

Algorithm S5 Orthonormal basis vectors of $\hat{H}_N$ from variable-bandwidth bistochastic kernel based on product kernel. We suppress N subscripts from our notation for $\boldsymbol{\phi}_{l,N}$.

```
Inputs
  1. Dataset $z_0, z_1, \dots, z_{N-1} \in Z$.
  2. Distance function $d : Z \times Z \to \mathbb{R}_+$.
  3. Response function $\hat{h} : Z \to \hat{Y}$.
  4. Distance function $d_{\hat{Y}} : \hat{Y} \times \hat{Y} \to \mathbb{R}_+$.
  5. Neighborhood parameters $k_{\text{nn}}, \hat{k}_{\text{nn}} \in \mathbb{N}$.
  6. Bandwidth exponent parameter $a > 0$ and range parameters $J_1, J_2 \in \mathbb{N}$.
  7. Bandwidth scaling parameter $s_\kappa > 0$.
  8. Number of basis vectors $L \le N$.

Outputs
  1. Column vectors $\boldsymbol{\phi}_0, \dots, \boldsymbol{\phi}_{L-1} \in \mathbb{R}^N$.

Steps
  1. Execute Algorithm S2 with inputs $\{z_n\}_{n=0}^{N-1}$, $d$, $a$, $J_1$, and $J_2$ to obtain a bandwidth function $b : Z \to \mathbb{R}_+$.
  2. Construct the distance-like function $\tilde{d} : Z \times Z \to \mathbb{R}_+$ with $\tilde{d}(z, z') = d(z, z')/\sqrt{b(z)b(z')}$.
  3. Execute Algorithm S6 with inputs $\{z_n\}_{n=0}^{N-1}$, $\tilde{d}$, $\eta_{\text{gauss}}$, $a$, $J_1$, and $J_2$ to obtain an optimal bandwidth $\epsilon_*$.
  4. Execute Algorithm S2 with inputs $\{\hat{h}(z_n)\}_{n=0}^{N-1}$, $d_{\hat{Y}}$, $a$, $J_1$, and $J_2$ to obtain a bandwidth function $b_{\hat{Y}} : \hat{Y} \to \mathbb{R}_+$.
  5. Construct the distance-like function $\tilde{d}_{\hat{Y}} : \hat{Y} \times \hat{Y} \to \mathbb{R}_+$ with $\tilde{d}_{\hat{Y}}(\hat{y}, \hat{y}') = d_{\hat{Y}}(\hat{y}, \hat{y}')/\sqrt{b_{\hat{Y}}(\hat{y})b_{\hat{Y}}(\hat{y}')}$.
  6. Execute Algorithm S6 with inputs $\{\hat{h}(z_n)\}_{n=0}^{N-1}$, $\tilde{d}_{\hat{Y}}$, $\eta_{\text{gauss}}$, $a$, $J_1$, and $J_2$ to obtain an optimal bandwidth $\epsilon_{\hat{Y},*}$.
  7. Construct the product kernel $\tilde{\kappa}_{\hat{h}} : Z \times Z \to \mathbb{R}_+$ with $\tilde{\kappa}_{\hat{h}}(z, z') = \eta_{\text{gauss}}(\tilde{d}(z, z')/(s_\kappa \epsilon_*))\eta_{\text{gauss}}(\tilde{d}_{\hat{Y}}(\hat{h}z, \hat{h}z')/(s_\kappa \epsilon_{\hat{Y},*}))$.
  8. Execute Algorithm S4 with inputs $\{z_n\}_{n=0}^{N-1}$ and $\tilde{\kappa}_{\hat{Y}}$ to obtain a non-symmetric kernel function $\hat{\kappa} : Z \times Z \to \mathbb{R}_+$.
  9. Form the $N \times N$ kernel matrix $\hat{\boldsymbol{K}} = [\hat{K}_{ij}]_{i,j=0}^{N-1}$ with $\hat{K}_{ij} = \hat{\kappa}(z_i, z_j)$.
  10. If $\hat{k}_{\text{nn}} < N$, set the $N - k_{\text{nn}}$ smallest elements in each row of $\hat{\boldsymbol{K}}$ to 0 and use a sparse array to store $\hat{\boldsymbol{K}}$.
  11. Return: The leading $L$ left singular vectors $\boldsymbol{\phi}_0, \dots, \boldsymbol{\phi}_{L-1}$ of $\hat{\boldsymbol{K}}$, arranged in order of decreasing corresponding singular value, and normalized such that $\|\boldsymbol{\phi}_l\|_2 = \sqrt{N}$.
```

D. Finite-dimensional Hilbert spaces and operator approximation. Given the basis vectors ϕ_l and $\phi_{l,N}$ from section 2.A, we define the L-dimensional Hilbert spaces

$$H_L = \operatorname{span}\{\phi_0, \dots, \phi_{L-1}\} \subset H, \quad H_{L,N} = \operatorname{span}\{\phi_{0,N}, \dots, \phi_{L-1,N}\} \subset \hat{H}_N,$$

where in the case of $H_{L,N}$ L is at most N. As in the main text, we let $\Pi_L : H \to H$ and $\Pi_{L,N} : \hat{H}_N \to \hat{H}_N$ be the orthogonal projections on H and $\hat{H}_N$, respectively, with $\operatorname{ran} \Pi_L = H_L$ and $\operatorname{ran} \Pi_{L,N} = H_{L,N}$. We also let $\boldsymbol{\Pi}_L : \mathfrak{B} \to \mathfrak{B}$ and $\boldsymbol{\Pi}_{L,N} : \hat{\mathfrak{B}}_N \to \hat{\mathfrak{B}}_N$ be the induced projections on the operator algebras $\mathfrak{B} = B(H)$ and $\mathfrak{B}_N = B(\hat{H}_N)$, defined as $\boldsymbol{\Pi}_L A = \Pi_L A \Pi_L$ and $\boldsymbol{\Pi}_{L,N} A = \Pi_{L,N} \hat{A} \Pi_{L,N}$, respectively. Defining $\mathfrak{B}_L = \operatorname{ran} \boldsymbol{\Pi}_L$, we can canonically identify $\mathfrak{B}_L$ with the subalgebra of $\mathfrak{B}$ consisting of all operators A satisfying $\ker A \supseteq H_L$ and $\operatorname{ran} A \subseteq H_L$. The space $\mathfrak{B}_{L,N} := \operatorname{ran} \boldsymbol{\Pi}_{L,N}$ can be canonically identified with a subalgebra of $\hat{\mathfrak{B}}_N$ in a similar manner. We will be making these identifications whenever convenient.

Within this setting, we are interested in the following two types of operator approximation, respectively described in subsections 2.D.1 and 2.D.2.

1. Approximation of an operator $A \in \mathfrak{B}$ by a finite-rank operator $A_L \in \mathfrak{B}_L$.
2. Approximation of $A_L \in \mathfrak{B}_L$ by an operator $A_{L,N} \in \mathfrak{B}_{L,N}$.

Intuitively, we think of an approximation of the first type listed above as a "compression" of an operator $A \in \mathfrak{B}$ of possibly infinite rank to an operator $A_L \in \mathfrak{B}_L \subset \mathfrak{B}$ of at most rank L. Approximations of the second type are of a fundamentally different nature since there are no inclusion relationships between $\mathfrak{B}_L$ and $\mathfrak{B}_{L,N}$. One can think instead of such approximations as data-driven approximations of the *representation* of an operator in a basis.

D.1. Operator compression. Given $A \in \mathfrak{B}$, we define $A_L \in \mathfrak{B}_L$ as

$$A_L := \boldsymbol{\Pi}_L A = \Pi_L A \Pi_L, \tag{S7}$$

Since $\{\phi_0, \phi_1, \dots\}$ is an orthonormal basis of H, the projections Π_L converge strongly to the identity; that is, for every $f \in H$, we have $\lim_{L\to\infty}(\Pi_L - I)f = 0$, where the limit is taken in the norm of H. As a result, the operators $\breve{A}_L := A\Pi_L$ converge strongly to A, $\lim_{L\to\infty}(\breve{A}_L - A)f = 0$ for all $f \in H$. It then follows from standard results in functional analysis that $A_L = \Pi_L \breve{A}_L$ converges strongly to A, i.e,

$$\lim_{L\to\infty} A_L f = Af, \quad \forall f \in H. \tag{S8}$$

Algorithm S6 Tuning of bandwidth-dependent kernels.

```
Inputs
  1. Dataset $x_0, x_1, \dots, x_{N-1} \in \mathcal{X}$; $\mathcal{X}$ is an arbitrary set.
  2. Bandwidth exponent parameter $a > 0$ and range parameters $J_1, J_2 \in \mathbb{N}$.
  3. Distance-like function $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$.
  4. Kernel shape function $\eta : \mathbb{R}_+ \to \mathbb{R}_+$.
Outputs
  1. Optimal bandwidth $\epsilon_* > 0$.
  2. Estimated dataset dimension $m_* \geq 0$.
Steps
  1. Compute the $N \times N$ pairwise distance matrix $\boldsymbol{D} = [D_{ij}]_{i,j=0}^{N-1}$ with $D_{ij} = d(x_i, x_j)$.
  2. Generate logarithmic grid $\{\epsilon_j\}_{j=J_1}^{J_2}$ with $\epsilon_j = 2^{aj}$.
  3. For each $j \in \{J_1, \dots, J_2\}$, compute the kernel sum $S(\epsilon_j) = \sum_{i,l=0}^{N-1} K_{il}/N^2$, where $K_{il} = \eta(D_{il}/\epsilon_j)$.
  4. For each $j \in \{J_1 + 1, \dots, J_2 - 1\}$, compute the logarithmic derivative
        $m(\epsilon_j) = \frac{\log S_{j+1} - \log S_{j-1}}{\log \epsilon_{j+1} - \log \epsilon_{j-1}} = \frac{\log(S_{j+1}/S_{j-1})}{2a}.$
  5. Return: $\epsilon_* = \operatorname{argmax}_{\epsilon_j \in \{\epsilon_{J_1}, \dots, \epsilon_{J_2}\}} m(\epsilon_j)$ and $m_* = m(\epsilon_*)$.
```

As we will see below, this type of strong operator convergence is sufficient to deduce convergence of the matrix mechanical formulation of data assimilation based on $\mathfrak{B}_L$ to the infinite-dimensional quantum mechanical level based on $\mathfrak{B}$ (see the rows labeled Ⓜ and Ⓠ in the schematic of Fig. 1).

D.2. Data-driven operator approximation. In order to facilitate approximation of operators in $\mathfrak{B}_L$ by operators in $\mathfrak{B}_{L,N}$, we use operators acting on the Banach space of continuous functions $C(X)$ as intermediate approximations. In what follows, we will let $\iota : C(X) \to H$ and $\iota_N : C(X) \to \hat{H}_N$ be the canonical linear maps that map continuous functions to their L^2 equivalence classes in H and $\hat{H}_N$, respectively. In addition, we let $\mathfrak{C} = B(C(X))$ be the unital Banach algebra of bounded linear operators on $C(X)$. We assume $L \in \mathbb{N}$ is chosen such that the eigenvalues λ_{L-1} and $\lambda_{L-1,N}$ of K and K_N from Eq. (S2) and Eq. (S3), respectively, are nonzero. This means that all elements of H_L and $H_{L,N}$ have continuous representatives.

With these definitions and assumptions, we restrict attention to approximation of operators $A_L \in \mathfrak{B}_{L,N}$ which are obtained by applying Eq. (S8) to operators $A \in \mathfrak{B}$ that satisfy

$$A \circ \iota = \iota \circ \tilde{A}, \tag{S9}$$

for some $\tilde{A} \in \mathfrak{C}$. In addition, we assume that there is a uniformly bounded family of operators $\hat{A}_N \in \mathfrak{B}_N$ that satisfy an approximate version of Eq. (S9) in the following sense: For every $f \in C(X)$, the norm of the residual $(\hat{A}_N \circ \iota_N - \iota_N \circ \tilde{A})f$ converges to 0. That is, we require

$$\lim_{N\to\infty} \|R_N f\|_{\hat{H}_N} = 0, \quad R_N = \hat{A}_N \circ \iota_N - \iota_N \circ \tilde{A}, \quad \forall f \in C(X), \tag{S10}$$

where the operators $\hat{A}_N$ satisfy the uniform norm bound

$$\|\hat{A}_N\|_{\hat{\mathfrak{B}}_N} \leq a, \tag{S11}$$

for a constant a. As we will see in the ensuing subsections, under Assumption 2, all operators employed in QMDA satisfy Eq. (S9), Eq. (S10), and Eq. (S11).

We have the following approximation lemma for the matrix elements of A in terms of the matrix elements of $\hat{A}_N$.

Lemma 2. *Suppose that $A \in \mathfrak{B}$, $\hat{A}_N \in \hat{\mathfrak{B}}_N$, and $\tilde{A} \in \mathfrak{C}$ satisfy Eq. (S9), Eq. (S10), and Eq. (S11). Then, under Assumptions 1 and 2, and with the notation and assumptions of section 2.A, the matrix elements of $\hat{A}_N$ in the $\{\phi_{l,N}\}$ bases of $\hat{H}_N$ converge almost surely to the matrix elements of A in the $\{\phi_l\}$ basis of H. That is, for μ-a.e. initial condition $x_0 \in X$, and every $i, j \in \mathbb{N}$ such that $\lambda_i, \lambda_j \neq 0$,*

$$\lim_{N\to\infty} \langle \phi_{i,N}, \hat{A}_N \phi_{j,N} \rangle_N = \langle \phi_i, A\phi_j \rangle.$$

Proof. See subsection 2.D.3. □

Let $A_{ij} = \langle \phi_i, A\phi_j \rangle$ and $A_{ij,N} = \langle \phi_{i,N}, \hat{A}_N \phi_{j,N} \rangle_N$. The convergence of $A_{ij,N}$ to A_{ij} from Lemma 2 is not uniform with respect to $i, j \in \mathbb{N}$. However, restricting i and j to the finite index set $\{0, \dots, L-1\}$ associated with the basis vectors of the finite-dimensional spaces H_L and $H_{L,N}$ makes the convergence of $\hat{A}_{ij,N}$ to $\hat{A}_{ij}$ uniform, and we can conclude that the matrix representations of the projected operators $A_{L,N} = \boldsymbol{\Pi}_{L,N} \hat{A}_N$ converge to the matrix representation of $A_L = \boldsymbol{\Pi}_L A$.

Corollary 3. *With notation as above, let $\boldsymbol{A}_L = [A_{ij}]_{i,j=0}^{L-1}$ and $\boldsymbol{A}_{L,N} = [A_{ij,N}]_{i,j=0}^{L-1}$ be the $L \times L$ matrix representations of A_L and $A_{L,N}$ in the $\{\phi_l\}$ and $\{\phi_{l,N}\}$ bases of H_L and $H_{L,N}$, respectively. Then, for μ-a.e. initial condition x_0, we have $\lim_{N\to\infty} \boldsymbol{A}_{L,N} = \boldsymbol{A}_L$ in any matrix norm.*

D.3. Proof of Lemma 2. Recall from section 2.A that the $\phi_{i,N}$ have continuous representatives $\varphi_{i,N}$ which converge μ-a.s. to the continuous representatives φ_i of ϕ_i in the uniform norm of $C(X)$. Note also that for every $N \in \mathbb{N}$, $\iota_N : C(X) \to \hat{H}_N$ has unit operator norm. Using these facts, we get

$$\begin{aligned}
\lvert \langle \phi_{i,N}, \hat{A}_N \phi_{j,N} \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert &= \lvert \langle \phi_{i,N}, \hat{A}_N \iota_N \varphi_{j,N} \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert \\
&\leq \lvert \langle \phi_{i,N}, \hat{A}_N \iota_N (\varphi_{j,N} - \varphi_j) \rangle_N \rvert + \lvert \langle \phi_{i,N}, \hat{A}_N \iota_N \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert \\
&\leq a \lVert \varphi_{j,N} - \varphi_j \rVert_{C(X)} + \lvert \langle \phi_{i,N}, \hat{A}_N \iota_N \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert. \qquad \text{[S12]}
\end{aligned}$$

Moreover, we have

$$\begin{aligned}
\lvert \langle \phi_{i,N}, \hat{A}_N \iota_N \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert &= \lvert \langle \phi_{i,N}, \iota_N \tilde{A} \varphi_j \rangle_N + \langle \phi_{i,N} R_N \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert \\
&\leq \lvert \langle \phi_{i,N}, \iota_N \tilde{A} \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert + \lVert R_N \varphi_j \rVert_{\hat{H}_N} \\
&= \lvert \langle \iota_N \varphi_{i,N}, \iota_N \tilde{A} \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert + \lVert R_N \varphi_j \rVert_{\hat{H}_N} \\
&= \lvert \langle \iota_N (\varphi_{i,N} - \varphi_i), \iota_N \tilde{A} \varphi_j \rangle_N + \langle \iota_N \varphi_i, \iota_N \tilde{A} \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert + \lVert R_N \varphi_j \rVert_{\hat{H}_N} \\
&\leq \lVert \varphi_{i,N} - \varphi_i \rVert_{C(X)} \lVert \tilde{A} \rVert_{\mathfrak{C}} \lVert \varphi_j \rVert_{C(X)} + \lvert \langle \iota_N \varphi_i, \iota_N \tilde{A} \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert + \lVert R_N \varphi_j \rVert_{\hat{H}_N}. \qquad \text{[S13]}
\end{aligned}$$

Now, by Eq. (S9) we have

$$\begin{aligned}
\lvert \langle \iota_N \varphi_i, \iota_N \tilde{A} \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert &= \lvert \langle \iota_N \varphi_i, \iota_N \tilde{A} \varphi_j \rangle_N - \langle \iota \varphi_i, A \iota \varphi_j \rangle \rvert = \lvert \langle \iota_N \varphi_i, \iota_N \tilde{A} \varphi_j \rangle_N - \langle \iota \varphi_i, \iota \tilde{A} \varphi_j \rangle \rvert \\
&= \left\lvert \int_X \varphi_i \tilde{A} \varphi_j d\mu_N - \int_X \varphi_i \tilde{A} \varphi_j \, d\mu \right\rvert,
\end{aligned}$$

so by the weak-* convergence of μ_N to μ (see Eq. (S1)) it follows that for μ-a.e. initial condition x_0,

$$\lim_{N\to\infty} \lvert \langle \iota_N \varphi_i, \iota_N \tilde{A} \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert = 0.$$

Using this result, the uniform convergence of $\varphi_{i,N}$ to φ_i, and Eq. (S10) in Eq. (S13), we obtain

$$\lim_{N\to\infty} \lvert \langle \phi_{i,N}, \hat{A}_N \iota_N \varphi_j \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert = 0.$$

Finally, using the above and the uniform convergence of $\varphi_{i,N}$ to φ_i in Eq. (S12), we arrive at

$$\lim_{N\to\infty} \lvert \langle \phi_{i,N}, \hat{A}_N \phi_{j,N} \rangle_N - \langle \phi_i, A\phi_j \rangle \rvert = 0,$$

which holds again for μ-a.e. initial condition x_0. This completes the proof of the lemma.

E. Approximation of states. Let $\omega_\rho \in S_*(\mathfrak{B})$ be a normal state of $\mathfrak{B}$ induced by a density operator $\rho \in \mathfrak{B}_*$. We recall that the predual $\mathfrak{B}_*$ of $\mathfrak{B}$ is the space of trace-class operators on H (denoted as $B_1(H)$ in the main text), equipped with the trace norm, $\lVert A \rVert_{\mathfrak{B}_*} = \operatorname{tr} \sqrt{A^* A}$. In the case of the finite-dimensional algebras $\mathfrak{B}_L$ and $\mathfrak{B}_{L,N}$, the preduals $\mathfrak{B}_{L*}$ and $\mathfrak{B}_{L,N*}$, respectively, can be identified with the algebras themselves, but we will continue to distinguish them using $_*$ subscripts since they are equipped with a different norm (the trace norm) from the operator norm of the algebras.

As in subsection 2.D, we are interested in two types of state approximation, which can be thought of as state compression and data-driven approximation, respectively:

1. Approximation of ρ by a finite-rank density operator $\rho_L \in \mathfrak{B}_L$; see subsection 2.E.1.
2. Approximation of ρ_L by a data-driven density operator $\rho_{L,N} \in \mathfrak{B}_{L,N*}$; see subsection 2.E.2.

E.1. State compression. Similarly to subsection 2.D.1, for a given density operator $\rho \in \mathfrak{B}_*$ we define the projected operators $\sigma_L = \boldsymbol{\Pi}_L \rho$. Letting $C_L = \operatorname{tr} \sigma_L$, we have $C_L \leq \operatorname{tr} \rho = 1$, so in general the σ_L are not density operators. Nevertheless, the σ_L are positive, finite-rank (and thus trace class) operators that converge to ρ in trace norm (as opposed to merely strongly; cf. Eq. (S8)). Indeed, we have $\rho - \sigma_L = (I - \Pi_L)\rho(I - \Pi_L)$, so $\rho - \sigma_L$ is positive, and

$$\lVert \rho - \sigma_L \rVert_{\mathfrak{B}_*} = \operatorname{tr}(\rho - \sigma_L) = \sum_{l=L}^{\infty} \langle \phi_l, \rho \phi_l \rangle,$$

where the sum in the right-hand side of the last equality is a positive, decreasing function of L, converging to 0 as $L \to \infty$. We also have $C_L = \sum_{l=0}^{L-1} \langle \phi_l, \rho \phi_l \rangle$, which implies that $\lim_{L\to\infty} C_L = 1$, and thus that there exists $L_* \in \mathbb{N}$ such that $C_L > 0$ for all $L > L_*$. For any such L, $\rho_L := \sigma_L / C_L$ is a density operator, and the sequence ρ_L converges to ρ in trace norm,

$$\lim_{L\to\infty} \lVert \rho_L - \rho \rVert_{\mathfrak{B}_*} = 0. \qquad \text{[S14]}$$

In the main text, we denote the map that sends the normal state $\omega_\rho \in S_*(\mathfrak{B})$ to $\omega_{\rho_L} \in S_*(\mathfrak{B}_L)$ as $\mathbf{\Pi}'_L(\rho) = \rho_L$.

Let now A be an element of $\mathfrak{B}$ with corresponding projected elements $A_L = \mathbf{\Pi}_L A \in \mathfrak{B}_L$ from Eq. (S7). By the cyclic property of the trace, we have $\operatorname{tr}(\rho_L A_L) = \operatorname{tr}(\rho_L A)$, and the trace-norm convergence in Eq. (S14) implies $\lim_{L\to\infty} \operatorname{tr}(\rho_L A_L) = \operatorname{tr}(\rho A)$. Equivalently, letting $\omega_\rho \in S_*(\mathfrak{B})$ and ω_{ρ_L} be the states of $\mathfrak{B}$ and $\mathfrak{B}_L$ induced by ρ and ρ_L, respectively, we have

$$\lim_{L\to\infty} \omega_{\rho_L} A_L = \omega_\rho A. \tag{S15}$$

We conclude that evaluation of the projected observables A_L on the projected states ω_{ρ_L} asymptotically recovers the evaluation of A on ρ.

E.2. Data-driven state approximation. Proceeding analogously to subsection 2.D.2, we seek data-driven approximations of projected density operators $\rho_L \in \mathfrak{B}_{L*}$ by density operators $\rho_{L,N} \in \mathfrak{B}_{L,N*}$ for a subset of density operators $\rho \in \mathfrak{B}_*$ that behave compatibly with bounded operators on continuous functions.

First, we recall that every density operator $\rho \in \mathfrak{B}_*$ admits a decomposition (diagonalization) of the form

$$\rho = \sum_{j=0}^{\infty} r_j \langle \xi_j, \cdot \rangle \xi_j, \tag{S16}$$

where $\{\xi_0, \xi_1, \dots\}$ is an orthonormal basis of H, $(r_0, r_1, \dots)$ is an ℓ^1 sequence of real numbers in the interval $[0,1]$, and the sum over j converges in the trace norm of $\mathfrak{B}_*$. In what follows, we shall restrict attention to a subset $S_C(\mathfrak{B}) \subset S_*(\mathfrak{B})$, consisting of all normal states ω_ρ of $\mathfrak{B}$ whose corresponding density operators $\rho \in \mathfrak{B}_*$ are decomposable as in Eq. (S16) with the following additional requirement: The orthonormal basis vectors ξ_j have uniformly bounded continuous representatives; that is, we have

$$\xi_j = \iota \tilde\xi_j, \quad \tilde\xi_j \in C(X), \quad \|\xi_j\|_{C(X)} \le b,$$

for a constant b. Given such an $\omega_\rho \in S_C(\mathfrak{B})$, for each $N \in \mathbb{N}$ we define the positive operator $\hat\sigma_N : \hat H_N \to \hat H_N$, where

$$\hat\sigma_N = \sum_{j=0}^{\infty} r_j \langle \hat\xi_{j,N}, \cdot \rangle_N \hat\xi_{j,N}, \quad \hat\xi_{j,N} = \iota_N \tilde\xi_j.$$

Note that the well-definition of $\hat\sigma_N$ follows from the uniform boundedness of the $\tilde\xi_j$ and the fact that $(r_0, r_1, \dots)$ is an ℓ^1 sequence. It should also be kept in mind that, in general, the σ_N are not normalized as density operators. We then have:

Lemma 4.

(a) $\tilde\rho : f \mapsto g$ *with* $g(x) = \sum_{j=0}^{\infty} r_j \langle \xi_j, \iota f\rangle \tilde\xi_j(x)$ *is well-defined as a linear map from* $C(X)$ *to itself, and it satisfies* $\iota \circ \tilde\rho = \rho \circ \iota$.

(b) For μ*-a.e. initial condition* $x_0 \in X$*, the residual* $R_N f = (\iota_N \circ \tilde\rho) f - (\hat\sigma_N \circ \iota_N) f$ *satisfies*

$$\lim_{N\to\infty} \|R_N f\|_{\hat H_N} = 0, \quad \forall f \in C(X).$$

Proof. See subsection 2.E.3. □

It follows from Lemma 4 that Eq. (S9) and Eq. (S10) hold with $A = \rho$, $\tilde A = \tilde\rho$, and $\hat A_N = \hat\sigma_N$. Thus, Lemma 2 and Corollary 3 apply, and for each $L \in \mathbb{N}$ such that $\lambda_{L-1} > 0$, the matrix representations $\boldsymbol{\sigma}_{L,N} = [\langle \phi_{i,N}, \hat\sigma_N \phi_{j,N}\rangle_N]_{i,j=0}^{L-1}$ of $\sigma_{L,N} = \Pi_{L,N} \hat\sigma_N$ converge to the matrix representation $\boldsymbol{\sigma}_L = [\langle \phi_i, \rho \phi_j \rangle]_{i,j=0}^{L-1}$ of $\sigma_L = \mathbf{\Pi}_L \rho$. If, in addition, L is sufficiently large such that $C_L > 0$, then the density operators $\rho_{L,N} \in \mathfrak{B}_{L,N*}$ defined as $\rho_{L,N} = \sigma_{L,N}/C_{L,N}$ with $C_{L,N} = \operatorname{tr} \sigma_{L,N}$ converge, as $N \to \infty$, in the sense of convergence of the corresponding matrix representations $\boldsymbol{\rho}_{L,N} = \boldsymbol{\sigma}_{L,N}/C_{L,N}$, to the density operator $\rho_L = \sigma_L / C_L$ with matrix representation $\boldsymbol{\rho}_L = \boldsymbol{\sigma}_L / C_L$. As we saw in subsection 2.E.1, the latter converges to ρ as $L \to \infty$ in the trace norm.

Combining the results of this section to those of section 2.D, we conclude that QMDA consistently approximates the action of normal states $\omega_\rho \in S_C(\mathfrak{B})$ on elements $A \in \mathfrak{B}$ satisfying Eq. (S9) and Eq. (S10) by the action of the data-driven states $\omega_{\rho_{L,N}} \in S(\mathfrak{B}_{L,N})$ on the data-driven elements $A_{L,N} \in \mathfrak{B}_{L,N}$ in the sense of the iterated limit

$$\lim_{L\to\infty} \lim_{N\to\infty} \omega_{\rho_{L,N}} A_{L,N} = \lim_{L\to\infty} \omega_{\rho_L} A_L = \omega_\rho A, \tag{S17}$$

where the first equality holds for μ-a.e. initial condition $x_0 \in X$.

E.3. Proof of Lemma 4.

(a) Fix $x \in X$ and $\epsilon > 0$. For any $x' \in X$ and $J \in \mathbb{N}$, we have

$$\left| \sum_{j=J}^{\infty} r_j \langle \xi_j, f \rangle \left(\tilde{\xi}_j(x) - \tilde{\xi}_j(x') \right) \right| \leq \|f\|_{C(X)} \sum_{j=J}^{\infty} r_j \left| \tilde{\xi}_j(x) - \tilde{\xi}_j(x') \right| \leq 2b \|f\|_{C(X)} \sum_{j=J}^{\infty} r_j .$$

Thus, since $(r_0, r_1, \dots) \in \ell^1$, there exists J such that $\left| \sum_{j=J}^{\infty} r_j \langle \xi_j, f \rangle (\tilde{\xi}_j(x) - \tilde{\xi}_j(x')) \right| < \epsilon$, for all $x' \in X$. We therefore have

$$|g(x) - g(x')| = \left| \sum_{j=0}^{\infty} r_j \langle \xi_j, f \rangle \left(\tilde{\xi}_j(x) - \tilde{\xi}_j(x') \right) \right| \leq \left| \sum_{j=0}^{J-1} r_j \langle \xi_j, f \rangle \left(\tilde{\xi}_j(x) - \tilde{\xi}_j(x') \right) \right| + \epsilon,$$

and the continuity of g follows from the fact that the first term in the right-hand side of the last inequality is a finite sum of continuous functions. The boundedness of g can be shown similarly. The relation $\iota \circ \tilde{\rho} = \rho \circ \iota$ follows directly from the definitions of ρ and $\tilde{\rho}$.

(b) Given $f \in C(X)$, we have

$$\|R_N f\|_{\hat{H}_N} = \|(\iota_N \circ \tilde{\rho}) f - (\hat{\sigma}_N \circ \iota_N) f\|_{\hat{H}_N} = \left\| \sum_{j=0}^{\infty} r_j \left(\langle \xi_j, \iota f \rangle - \langle \hat{\xi}_{j,N}, \iota_N f \rangle_N \right) \hat{\xi}_{j,N} \right\|_{\hat{H}_N} \leq \sum_{j=0}^{\infty} r_j \left| \langle \xi_j, \iota f \rangle - \langle \hat{\xi}_{j,N}, \iota_N f \rangle_N \right| b.$$

Since $\left| \langle \xi_j, \iota f \rangle - \langle \hat{\xi}_{j,N}, \iota_N f \rangle_N \right| \leq 2b \|f\|_{C(X)}$ and $(r_0, r_1, \dots) \in \ell^1$, for every $\epsilon > 0$ there exists $J \in \mathbb{N}$ such that

$$\sum_{j=J}^{\infty} r_j \left| \langle \xi_j, \iota f \rangle - \langle \hat{\xi}_{j,N}, \iota_N f \rangle_N \right| b < \epsilon/2.$$

Moreover, by Eq. (S1), for μ-a.e. $x_0 \in X$ and every $j \in \mathbb{N}$ we have $\lim_{N\to\infty} \left| \langle \xi_j, \iota f \rangle - \langle \hat{\xi}_{j,N}, \iota_N f \rangle_N \right| = 0$, so there exists $N_* \in \mathbb{N}$ such that

$$\sum_{j=0}^{J-1} r_j \left| \langle \xi_j, \iota f \rangle - \langle \hat{\xi}_{j,N}, \iota_N f \rangle_N \right| < \epsilon/2, \quad \forall N > N_*.$$

We thus have

$$\|R_N f\|_{\hat{H}_N} \leq \sum_{j=0}^{J-1} r_j \left| \langle \xi_j, \iota f \rangle - \langle \hat{\xi}_{j,N}, \iota_N f \rangle_N \right| + \sum_{j=J}^{\infty} r_j \left| \langle \xi_j, \iota f \rangle - \langle \hat{\xi}_{j,N}, \iota_N f \rangle_N \right| < \epsilon, \quad \forall N > N_*,$$

and since ϵ was arbitrary, we conclude that $\lim_{N\to\infty} \|R_N f\|_{\hat{H}_N} = 0$. This completes the proof of Lemma 4.

F. Approximation of the forecast observable and its spectral measure. In this subsection, we examine the QMDA representation of the forecast observable $f \in \mathfrak{A}$ by projected multiplication operators in $\mathfrak{B}_{L,N}$ which we denote, as in the main text, by $\pi_{L,N} f$. We are interested in two types of asymptotic consistency of our representations, respectively described in subsections 2.F.1 and 2.F.2:

1. Pointwise consistency, meaning that evaluation of $\pi_{L,N} f$ on the states $\omega_{\rho_{L,N}}$ from section 2.E should converge to evaluation of the multiplication operator $\pi f \in \mathfrak{B}$ on the state ω_ρ approximated by $\omega_{\rho_{L,N}}$.
2. Spectral consistency, meaning that the spectral measures of $\pi_{L,N} f$ should converge to the spectral measure of πf in a suitable sense.

In QMDA applications, pointwise consistency is required for consistency of the forecast mean and variance with the theoretical forecast mean and variance, respectively, from the infinite-dimensional data assimilation system based on the algebra $\mathfrak{B}$ (i.e., the quantum mechanical level Ⓠ in Fig. 1). Meanwhile, spectral consistency is required for consistency of the corresponding forecast probabilities (denoted as $\mathbb{P}_{f,t,\tau}$ in the main text).

F.1. Pointwise approximation and its consistency. For a given trajectory $X_N := \{x_0, \dots, x_{N-1}\} \subset X$, let $\hat{\mathfrak{A}}_N$ denote the finite-dimensional, abelian von-Neumann algebra of complex-valued functions on X_N with respect to pointwise function multiplication and complex conjugation, equipped with the maximum norm, $\|u\|_{\hat{\mathfrak{A}}_N} = \max_{x_n \in X_N} |u(x_n)|$. As a vector space, $\hat{\mathfrak{A}}_N$ is isomorphic to the Hilbert space $\hat{H}_N$, but the two spaces have different norms. Every function $f : X \to \mathbb{C}$ induces an element $\hat{f}_N \in \hat{\mathfrak{A}}_N$ by restriction to X_N, $\hat{f}_N(x_n) = f(x_n)$ for all $n \in \{0, \dots, N-1\}$. Reusing notation, we will denote the linear map that maps f to $\hat{f}_N$ by ι_N. Analogously to $\pi : \mathfrak{A} \to \mathfrak{B}$, the algebra $\hat{\mathfrak{A}}_N$ has a regular representation $\hat{\pi}_N : \hat{\mathfrak{A}}_N \to \hat{\mathfrak{B}}_N$ such that, given $u \in \hat{\mathfrak{A}}$, $\hat{\pi}_N u$ is the multiplication operator by u, i.e., $(\hat{\pi}_N u) v = uv$ for all $v \in \hat{\mathfrak{A}}_N$. Moreover, similarly to $\pi_L : \mathfrak{A} \to \mathfrak{B}_L$, for each $L \in \{1, \dots, N\}$ we define the linear map $\pi_{L,N} : \hat{\mathfrak{A}}_N \to \mathfrak{B}_{L,N}$, $\pi_{L,N} = \mathbf{\Pi}_{L,N} \circ \hat{\pi}_N$, which maps elements of $\hat{\mathfrak{A}}_N$ to projected multiplication operators in $\mathfrak{B}_{L,N}$. Note that, in general, neither π_L nor $\pi_{L,N}$ are algebra homomorphisms.

Consider now the C^*-algebra of continuous functions on X, $C(X)$, and its regular representation $\tilde\pi : C(X) \to \mathfrak{C}$, where $\tilde\pi \tilde f$ is the multiplication operator by $\tilde f$, i.e., $(\tilde\pi \tilde f)\tilde g = \tilde f \tilde g$ for all $\tilde g \in C(X)$. One readily verifies that for every $\tilde f \in C(X)$,

$$\iota \circ (\tilde\pi \tilde f) = (\pi f) \circ \iota, \quad \iota_N \circ (\tilde\pi f) = (\hat\pi_N \hat f_N) \circ \iota_N, \tag{S18}$$

where $f = \iota \tilde f \in \mathfrak{A}$ and $\hat f_N = \iota_N \tilde f \in \hat{\mathfrak{A}}_N$. As a result, Eq. (S9) and Eq. (S10) hold for $A = \pi f$, $\tilde A = \tilde\pi \tilde f$, and $\hat A_N = \hat\pi_N \hat f_N$, and by Lemma 2 and Corollary 3 we can consistently approximate πf by the projected multiplication operators $\pi_L f$ and $\pi_{L,N} \hat f_N$.

In the main text, $\pi_L f$ and $\pi_{L,N} \hat f_N$ were used to represent the forecast observable f in the matrix mechanical and data-driven formulations of QMDA, respectively. Under Assumption 2, Eq. (S17) and Eq. (S18) together lead to the following consistency result for these representations,

$$\lim_{L\to\infty} \lim_{N\to\infty} \omega_{\rho_{L,N}}(\pi_{L,N} \hat f_N) = \lim_{L\to\infty} \omega_{\rho_L}(\pi_L f) = \omega_\rho(\pi f),$$

which holds for μ-a.e. initial condition $x_0 \in X$.

As with other linear maps employed in QMDA, in numerical applications we employ the $L \times L$ matrix representation of $\pi_{L,N} \hat f_N$, given by $\boldsymbol{A}_{L,N} = [\langle \phi_{i,N}, (\pi_{L,N} \hat f_N) \phi_{j,N} \rangle_N]_{i,j=0}^{L-1}$. Algorithm S7 describes the computation of this matrix (as well as the spectral measure of $\pi_{L,N} \hat f_N$, which we discuss in subsection 2.F.2 below). By Corollary 3, for μ-a.e. initial condition x_0, $\boldsymbol{A}_{L,N}$ converges as $N \to \infty$ to the matrix representation $\boldsymbol{A}_L = [\langle \phi_i, \pi_L f \phi_j \rangle]_{i,j=0}^{L-1}$ of $\pi_L f$.

Algorithm S7 Projected multiplication operator representing the forecast observable f and evaluation of the associated spectral measure. We suppress L and N subscripts from our notation of $\boldsymbol{A}_{L,N}$ and $\boldsymbol{E}_{L,N,}(S_m)$.

Inputs

1. Training observable values $f_0, \dots, f_{N-1} \in \mathbb{R}$.
2. Basis vectors $\boldsymbol{\phi}_0, \dots, \boldsymbol{\phi}_{L-1}$ from Algorithm S3.
3. Intervals (spectral bins) $S_0, \dots, S_{M-1} \subseteq \mathbb{R}$.

Require: The training data z_n used in the computation of $\boldsymbol{\phi}_l$ are induced by the same dynamical states $x_n \in X$ underlying f_n, i.e., $z_n = z(x_n)$ and $f_n = f(x_n)$.

Outputs

1. $L \times L$ matrix $\boldsymbol{A}$ representing the projected multiplication operator $\pi_{L,N} f$ in the $\phi_{l,N}$ basis of $H_{L,N}$.
2. $L \times L$ projection matrices $\boldsymbol{E}_0, \dots, \boldsymbol{E}_{M-1}$, where $\boldsymbol{E}_m$ is the matrix representation of the spectral projector $E_{\pi_{L,N} \hat f_N}(S_m)$ in the $\{\phi_{l,N}\}$ basis of $H_{L,N}$.

Steps

1. **Return:** $\boldsymbol{A} = [A_{ij}]_{i,j=0}^{L-1}$, where $A_{ij} = \boldsymbol{\phi}_i^\top (\boldsymbol{f} \odot \boldsymbol{\phi}_j)/N$, $\boldsymbol{f} = (f_0, \dots, f_{N-1})^\top$, and $\odot$ denotes elementwise multiplication of column vectors.
2. Compute the eigendecomposition $\boldsymbol{A}\boldsymbol{u}_j = a_j \boldsymbol{u}_j$, where $a_0, \dots, a_L \in \mathbb{R}$ and the eigenvectors $\boldsymbol{u}_0, \dots, \boldsymbol{u}_{L-1} \in \mathbb{R}^L$ satisfy $\boldsymbol{u}_i^\top \boldsymbol{u}_j = \delta_{ij}$.
3. **Return:** The projection matrices $\boldsymbol{E}_0, \dots, \boldsymbol{E}_{M-1}$, where $\boldsymbol{E}_m = \sum_{a_j \in S_m} \boldsymbol{u}_j \boldsymbol{u}_j^\top$.

F.2. Spectral approximation. We are interested in approximating the spectral measure of the multiplication operator $\pi f \in \mathfrak{B}$ associated with the forecast observable f by the spectral measures of the finite-rank operators $\pi_L f \in \mathfrak{B}_L$ and $\pi_{L,N} \hat f_N \in \mathfrak{B}_{L,N}$.

First, we recall that the spectrum of an element a of a unital C^*-algebra is the set of complex numbers z such that $a - z$ does not have an inverse. We denote this set as $\sigma(a)$. In the case of a finite-dimensional operator algebra such as $\mathfrak{B}_L$, and $\mathfrak{B}_{L,N}$, the spectrum of any element a is a finite set consisting of the eigenvalues of a. In the case of the infinite-dimensional operator algebra $\mathfrak{B}$, the spectrum of a multiplication operator πf by an element $f \in \mathfrak{A}$ is equal to the essential range of f, i.e., the support of the pushforward measure $f_*(\mu) : \mathcal{B}(\mathbb{C}) \to \mathbb{C}$ on the Borel σ-algebra on $\mathcal{B}(\mathbb{C})$ on $\mathbb{C}$. Note that $\sigma(\pi f)$ coincides with the spectrum of f as an element of the abelian algebra $\mathfrak{A}$, defined as the set of complex numbers for which $f - z$ is non-invertible.

The point spectrum of πf, i.e., the set of its eigenvalues, consists of all elements $z \in \sigma(\pi f)$ such that the preimage $S = f^{-1}(\{z\}) \subseteq X$ has positive measure, $\mu(S) > 0$. We denote the point spectrum of πf by $\sigma_p(\pi f)$. Points in the complement of $\sigma_p(\pi f)$ in $\sigma(\pi f)$ lie in the continuous spectrum of πf, and have no associated eigenspaces. A challenge with spectral approximation of multiplication operators on infinite-dimensional operator algebras is that generically they have a non-empty continuous spectrum, whereas the continuous spectrum of any finite-rank approximation of these operators is necessarily empty.

Suppose now that $f \in \mathfrak{A}$ is real-valued as per Assumption 1(c), i.e., f is a self-adjoint element of the abelian algebra $\mathfrak{A}$. Then, $\pi f \in \mathfrak{B}$ is a self-adjoint operator, and the spectrum $\sigma(\pi f)$ is a subset of the real line. By the spectral theorem for self-adjoint operators, there is a unique projection-valued measure (PVM) $E : \mathcal{B}(\mathbb{R}) \to \mathfrak{B}$, giving πf through the spectral integral $\pi f = \int_{\mathbb{R}} u \, dE(u)$. By construction, the operators $\pi_L f \in \mathfrak{B}_L$ and $\pi_{L,N} \hat f_N$ are also self-adjoint whenever f is real-valued, and thus have associated PVMs $E_L : \mathcal{B}(\mathbb{R}) \to \mathfrak{B}_L$ and $E_{L,N} : \mathcal{B}(\mathbb{R}) \to \mathfrak{B}_{L,N}$, respectively, such that $\pi_L f = \int_{\mathbb{R}} u \, dE_L(u)$

and $\pi_{L,N}\hat{f}_N = \int_{\mathbb{R}} u \, dE_{L,N}(u)$. Since the spectra $\sigma(\pi_L f)$ and $\sigma(\pi_{L,N} f)$ are finite sets, these spectral measures are discrete. Explicitly, for a given Borel set $S \in \mathcal{B}(\mathbb{R})$, we have

$$E_L(S) = \sum_{l : a_{L,l} \in S} E_{L,l}, \quad E_{L,N}(S) = \sum_{l : a_{L,N,l} \in S} E_{L,N,l}, \tag{S19}$$

where $\{a_{L,j}\} = \sigma(\pi_L f)$ and $\{a_{L,N,j}\} = \sigma(\pi_{L,N} f)$ are the sets of eigenvalues of $\pi_L f$ and $\pi_{L,N} f$ (without multiplicities), and $E_{L,j} \in \mathfrak{B}_L$ and $E_{L,N,j} \in \mathfrak{B}_{L,N}$ are the orthogonal projections onto the corresponding eigenspaces, respectively.

In what follows, we characterize the convergence of $E_{L,N}$ to E_L as $N \to \infty$ (large data limit) and E_L to E as $L \to \infty$ (infinite-dimension limit).

Spectral convergence in the large-data limit. Let $\boldsymbol{A}_L = [\langle \phi_i, (\pi_L f) \phi_j \rangle]_{i,j=0}^{L-1}$ and $\boldsymbol{A}_{L,N} = [\langle \phi_{i,N}, (\pi_{L,N} \hat{f}_N) \phi_{j,N} \rangle_N]_{i,j=0}^{L-1}$ be the matrix representations of $\pi_L f$ and $\pi_{L,N} f$ in the $\{\phi_l\}$ and $\{\phi_{l,N}\}$ bases of H_L and $H_{L,N}$, respectively. In the same bases, the spectral measures E_L and $E_{L,N}$ are represented by matrix-valued measures $\boldsymbol{E}_L : \mathcal{B}(\mathbb{R}) \to \mathbb{M}_L$ and $\boldsymbol{E}_{L,N} : \mathcal{B}(\mathbb{R}) \to \mathbb{M}_L$ such that $\boldsymbol{E}_L(S) = [\langle \phi_i, E_L(S) \phi_j \rangle]_{i,j=0}^{L-1}$ and $\boldsymbol{E}_{L,N}(S) = [\langle \phi_{i,N}, E_{L,N}(S) \phi_{j,N} \rangle_N]_{i,j=0}^{L-1}$ are the matrix representations of the projections $E_L(S)$ and $E_{L,N}(S)$ respectively. Since $\boldsymbol{A}_{L,N}$ converges to $\boldsymbol{A}_L$, it follows from spectral approximation results for finite-rank operators (15) that if the boundary of S does not contain any eigenvalues of $\pi_L f$, then for μ-a.e. initial condition x_0 and Borel set $S \in \mathcal{B}(\mathbb{R})$, $\boldsymbol{E}_{L,N}(S)$ converges as $N \to \infty$ to $\boldsymbol{E}_L(S)$.

Algorithm S7 describes the computation of the spectral projectors $\boldsymbol{E}_{L,N}(S_m)$ on a set of pairwise-disjoint intervals ("spectral bins") $S_0, \dots, S_{M-1} \subseteq \mathbb{R}$ partitioning the range of $\hat{f}_N$ in the training data. We choose the intervals S_m such that they carry equal probability mass under the distribution of $\hat{f}_N$ with respect to the sampling measure μ_N. With this choice, the boundaries of the S_m can be computed from the values of the quantile function of $\hat{f}_N$ on a uniform partition of $[0, 1]$; see Algorithm S8 for further details.

Algorithm S8 Spectral bins for the forecast observable f from the empirical quantile function.

Inputs

1. Training observable values $f_0, \dots, f_{N-1} \in \mathbb{R}$.
2. Number of spectral bins $M \in \mathbb{N}$.

Outputs

1. Intervals (spectral bins) $S_0, \dots, S_{M-1} \subseteq \mathbb{R}$.

Steps

1. Compute the empirical quantile function of f, $Q_f : (0, 1) \to (0, \infty)$, associated with the samples f_n.
2. Define $b_1, \dots, b_{M-1} \in \mathbb{R}$ with $b_m = m/M$.
3. **Return:** The intervals $S_0, \dots, S_{M-1}$, where

$$S_m = \begin{cases} (-\infty, Q_f(b_1)], & m = 0, \\ (Q_f(b_m), Q_f(b_{m+1})], & 1 < m < M - 1, \\ (Q_f(b_{M-1}), \infty), & m = M - 1. \end{cases}$$

Spectral convergence in the infinite-dimension limit. We employ the following results on spectral approximation of self-adjoint operators.

Theorem 5. *Let $A_L : H \to H$ be a sequence of finite-rank, self-adjoint operators on a Hilbert space H converging strongly as $L \to \infty$ to a self-adjoint operator $A : H \to H$. Let $E : \mathcal{B}(\mathbb{R}) \to B(H)$ and $E_L : \mathcal{B}(\mathbb{R}) \to B(H)$ be the spectral measures of A and A_L, respectively. Then, the following hold.*

(a) For every element a of the spectrum of A, there exists a sequence a_L of eigenvalues of A_L such that $\lim_{L \to \infty} a_L = a$.

(b) For every Borel set $S \in \mathcal{B}(\mathbb{R})$ such that $E(\partial S) = 0$ (i.e., the boundary of S does not contain any eigenvalues of A), the spectral projections $E_L(S)$ converge to $E(S)$ in the strong operator topology of $B(H)$, i.e.,

$$\lim_{L \to \infty} E_L(S) f = E(S) f, \quad \forall f \in H.$$

Proof. Strong convergence of bounded self-adjoint operators implies convergence in the strong resolvent sense; see ref. (16), Proposition 10.1.13(a). Strong resolvent convergence of operators implies in turn spectral convergence as stated in Part (a); see ref. (16), Corollary 10.2.2. For Part (b), see ref. (5), Proposition 13(iii), which states the analogous result under strong resolvent convergence of skew-adjoint operators. □

Since the projected multiplication operators $\pi_L f$ converge to πf strongly and they are self-adjoint whenever f is real-valued, it follows from Theorem 5 with $A = \pi f$ and $A_L = \pi_L f$ that the spectra and spectral measures of $\pi_L f$ converge to those of πf in the sense stated in the theorem.

G. Koopman operator approximation. Let $U \equiv U^{\Delta t}$ be the unitary Koopman operator on H associated with the the temporal sampling interval Δt of the data. Following refs. (1–5), we approximate U by a shift operator $\hat{U}_N : \hat{H}_N \to \hat{H}_N$. Here, we define $\hat{U}_N$ as

$$\hat{U}_N f(x_n) = \begin{cases} f(x_{n+1}), & 0 \leq n \leq N-2, \\ f(x_0), & n = N-1. \end{cases} \tag{S20}$$

With this definition, $\hat{U}_N$ is a unitary operator acting as a left circular shift on the sequence of values $f(x_0), \dots, f(x_{N-1})$.

Next, let $\tilde{U} : C(X) \to C(X)$ be the time-Δt Koopman operator on continuous (as opposed to L^2) functions, defined as usual by composition with the time-Δt flow, $\tilde{U} = f \circ \Phi^{\Delta t}$. Since $\Phi^{\Delta t}$ is μ-preserving, we have

$$U \circ \iota = \iota \circ \tilde{U},$$

so Eq. (S9) is satisfied for $A = U$ and $\tilde{A} = \tilde{U}$. Moreover, one verifies that Eq. (S10) is satisfied for $\hat{A}_N = \hat{U}_N$ and $\tilde{A} = \tilde{U}$ by observing that for every $f \in C(X)$ the vectors $g = \hat{U}_N \circ \iota_N f$ and $g' = \iota_N \circ \tilde{U} f$ differ only in the $(N-1)$-th component; i.e., $g(x_n) = g'(x_n)$ for all $n \in \{0, \dots, N-2\}$. This implies that the residual $R_N f = g - g'$ has norm

$$\|R_N f\|_{\hat{H}_N} \leq \|f\|_{C(X)} / \sqrt{N},$$

which verifies Eq. (S10). It therefore follows that Lemma 2 and Corollary 3 apply for U, $\hat{U}_N$, and $\tilde{U}$. Thus, for any μ-a.e. initial condition x_0 and each $L \in \mathbb{N}$ such that $\lambda_{L-1} > 0$, the data-driven shift operator matrices, $\boldsymbol{U}_{L,N} = [\langle \phi_{i,N}, \hat{U}_N \phi_{j,N} \rangle_N]_{i,j=0}^{L-1}$, converge as $N \to \infty$ to the projected time-Δt Koopman operator matrices, $\boldsymbol{U}_L = [\langle \phi_i, U \phi_j \rangle]_{i,j=0}^{L-1}$. The projected Koopman operators U_L converge in turn as $L \to \infty$ to U in the strong operator topology of $\mathfrak{B}$, as described in section 2.D.1.

To approximate the Koopman operator U^t at time $t = t_q := q\,\Delta t$ with $q \in \mathbb{Z}$, we repeat the construction described above using the q-th power of the shift operator, $\hat{U}_N^q$, as the approximating operator on $\hat{H}_N$, which is equivalent to a circular shift by q steps. This leads to projected operators $U_{L,N}^{(q)} = \mathbf{\Pi}_{L,N} \hat{U}_N^q$ whose matrix representations $\boldsymbol{U}_{L,N}^{(q)}$ converge as $N \to \infty$ to the matrix representation of $U_L^{(t_q)} = \mathbf{\Pi}_L U^{t_q}$. As $L \to \infty$, $U_L^{(t_q)}$ converges strongly to U^{t_q} (see Eq. (S8)). Thus, we obtain an asymptotically consistent approximation of the dynamical operators employed in QMDA for any given (finite) time horizon $t_q = q\,\Delta t$.

The construction of the $\boldsymbol{U}_{L,N}^{(q)}$ matrices is described in Algorithm S9. It is important to note that acting with $\hat{U}_N^q$ on elements of $\hat{H}_N$ does not require explicit knowledge of the states x_n. It should also be kept in mind that, unless $H_{L,N}$ is a $\hat{U}_N$-invariant subspace of $\hat{H}_N$, $U_{L,N}^{(q)}$ is not a unitary operator, and it is not equal to the q-th power of $U_{L,N}$. Nevertheless, by unitarity of $\hat{U}_N^q$ we have $U_{L,N}^{(q)*} = U_{L,N}^{(-q)}$. Similarly, $U_L^{(t)}$ is in general not equal to $(U_L)^{t/\Delta t}$, it is not unitary, but it satisfies $U_L^{(t)*} = U_L^{(-t)}$.

Algorithm S9 Koopman operator approximation. We suppress L and N indices from our notation of $\boldsymbol{U}_{L,N}^{(q)}$.

Inputs

1. Basis vectors $\boldsymbol{\phi}_0, \dots, \boldsymbol{\phi}_{L-1}$ from Algorithm S3.
2. Time-shift parameter $q \in \mathbb{Z}$.

Require: The underlying training data $z_0, \dots, z_{N-1}$ are time-ordered and are taken with a uniform sampling interval $\Delta t > 0$.

Outputs

1. $L \times L$ matrix $\boldsymbol{U}^{(q)}$ representing the projected Koopman operator on $U_{L,N}^{q\,\Delta t}$ on $H_{L,N}$.

Steps

1. For each $l \in \{0, \dots, L-1\}$, compute the time-shifted vectors $\boldsymbol{\phi}_i^{(q)} = (\phi_{0l}^{(q)}, \dots, \phi_{N-1,l}^{(q)})^\top \in \mathbb{R}^N$ with
$$\phi_{nl} = \phi_{n'l}, \quad n' = n + l \mod N.$$
2. **Return:** The $L \times L$ matrix $\boldsymbol{U}^{(q)} = [U_{ij}^{(q)}]_{i,j=0}^{L-1}$ with $U_{ij} = \boldsymbol{\phi}_i^\top \boldsymbol{\phi}_j^{(q)} / N$.

Remark. The shift operator in Eq. (S20) differs somewhat from the operators used in refs. (1–5), which employ the non-unitary definition $\check{U}_N : \hat{H}_N \to \hat{H}_N$ with

$$\check{U}_N f(x_n) = \begin{cases} f(x_{n+1}), & 0 \leq n \leq N-2, \\ 0, & n = N-1. \end{cases}$$

The difference between the two approaches is inconsequential in the large-data limit, i.e., both $\hat{U}_{N,L}$ and $\check{U}_{N,L} := \mathbf{\Pi}_L \check{U}_N$ exhibit the $N \to \infty$ convergence in Lemma 2 and Corollary 3. Here, we have opted to work with $\hat{U}_N$ from Eq. (S20), for, as we will see in section 2.H below, the unitarity of this operator ensures that the induced Koopman operator on $\hat{\mathfrak{B}}_N$ is a quantum channel.

H. Finite-dimensional quantum operations. In the main text, we introduced the projected Koopman operator $\mathcal{U}_L^{(t)} : \mathfrak{B}_L \to \mathfrak{B}_L$, defined as

$$\mathcal{U}_L^{(t)} A = U_L^{(t)} A U_L^{(t)*} = U_L^{(t)} A U_L^{(-t)};$$

equivalently, $\mathcal{U}_L^{(t)} = \mathbf{\Pi}_L \circ \mathcal{U}^t$. Letting $\varpi : \mathfrak{B} \to \mathfrak{B}$ be the trivial representation, $\varpi(A) = A$, we have $\mathcal{U}_L^{(t)} = \Pi_L \varpi(U^t) \Pi_L$, so by Stinespring's theorem (see subsection "Positivity-preserving discretization" in the main text) $\mathcal{U}_L^{(t)}$ is completely positive. We also have that $\mathcal{U}_L^{(t)} = (\mathcal{P}_{L*}^{(t)})^*$, where $\mathcal{P}_{L*}^{(t)} : \mathfrak{B}_{L*} \to \mathfrak{B}_{L*}$ is the transfer operator on the predual of $\mathfrak{B}_L$, defined as $\mathcal{P}_{L*}^{(t)} A = U_L^{(-t)} A U_L^{(t)}$.

Next, let $\rho \in \mathfrak{B}_{L*}$ be a density operator. This operator extends to a density operator $\varrho \in \mathfrak{B}_*$ given by $\varrho = \Pi_L \rho \Pi_L$. One can then verify that the transfer operator $\mathcal{P}_L^{(t)} : \mathfrak{B}_{L*} \to \mathfrak{B}_{L*}$ satisfies

$$(\mathcal{P}_L^{(t)} \omega_\rho) \equiv \omega_{\mathcal{P}_{L*}^{(t)} \rho} \mathbf{1}_L = \operatorname{tr}((\mathcal{U}^{-t} \varrho) \Pi_L) \leq \lVert \Pi_L \rVert_{\mathfrak{B}} = 1. \qquad \text{[S21]}$$

By virtue of this fact and the complete positivity of $\mathcal{U}_L^{(t)}$ it follows that $\mathcal{U}_L^{(t)}$ is a quantum operation.

Remark. If H_L happens to be a Koopman-invariant subspace of H, i.e., $U^t H_L = H_L$, then $\operatorname{tr}((\mathcal{U}^{-t} \varrho) \Pi_L) = \operatorname{tr}(\mathcal{U}^{-t} \varrho) = 1$ and $\mathcal{U}_L^{(t)}$ is a quantum channel. This property holds if and only if H_L is an orthogonal direct sum of Koopman eigenfunctions. The existence of such distinguished subspaces of H cannot be assumed for general measure-preserving dynamical systems. For instance, it is a standard result from ergodic theory that if the dynamical flow Φ^t is measure-theoretically mixing, then the Koopman operator $U^t : H \to H$ has only constant eigenfunctions (17).

In the data-driven setting of $\mathfrak{B}_{L,N}$, we employ an analogous construction based on the unitary shift operator $\hat{U}_N^q$. The shift operator induces a unitary $\hat{\mathcal{U}}_N^q : \hat{\mathfrak{B}}_N \to \hat{\mathfrak{B}}_N$ that acts by conjugation by $\hat{U}_N^q$, i.e., $\hat{\mathcal{U}}_N^q A = \hat{U}_N^q A \hat{U}_N^{q*}$. This operator is a quantum channel analogously to the Koopman operator $\mathcal{U}^t : \mathfrak{B} \to \mathfrak{B}$. For $L \leq N - 1$ we define the projected shift operator $\mathcal{U}_{L,N}^{(q)} : \mathfrak{B}_{L,N} \to \mathfrak{B}_{L,N}$ such that $\mathcal{U}_{L,N}^{(q)} A = U_{L,N}^{(q)} A U^{(q)*}$. This operator has entirely analogous properties to the projected Koopman operator $\mathcal{U}_L^{(t)}$; that is, $\mathcal{U}_{L,N}^{(q)}$ is a completely positive map whose associated transfer operator $\mathcal{P}_{L,N}^{(q)} : \mathfrak{B}_{L,N}^* \to \mathfrak{B}_{L,N}^*$ with $\mathcal{P}_{L,N}^{(q)} = \mathcal{U}_{L,N}^{(q)*}$ is trace-non-increasing (i.e., satisfies an analog of Eq. (S21)). Thus, $\mathcal{U}_{L,N}^{(q)}$ is a quantum operation.

```
Algorithm S10 QMDA forecast–analysis step.
Inputs
  1. Forecast timesteps $J_\mathrm{f} \in \mathbb{N}$; observation timesteps $J_\mathrm{o} \in \mathbb{N}$.
  2. Koopman matrices $\boldsymbol{U}^{(1)}, \dots, \boldsymbol{U}^{(J)} \in \mathbb{M}_L$ from Algorithm S9 with $J = \max\{J_\mathrm{f}, J_\mathrm{o}\}$.
  3. Spectral bins $S_0, \dots, S_{M-1} \subseteq R$ from Algorithm S8.
  4. Forecast observable $\boldsymbol{A} \in \mathbb{M}_L$ and spectral projectors $\boldsymbol{E}_0, \dots, \boldsymbol{E}_{M-1} \in \mathbb{M}_L$ from Algorithm S7.
  5. Matrix-valued effect $\boldsymbol{F} : Y \to \mathbb{M}_L$ from Algorithm S12.
  6. Initial density matrix $\boldsymbol{\rho} \in \mathbb{M}_L$.
  7. Observation $y \in Y$ at time $J_\mathrm{o}\, \Delta t$.
Require: All $L \times L$ matrices are representations of operators in $\mathfrak{B}_{L,N}$ in the same data-driven basis $\{\phi_{l,N}\}_{l=0}^{L-1}$ from Algorithm S3. All training data are generated by the same sequence of (unknown) time-ordered states $x_0, \dots, x_{N-1} \in X$ with $x_n = \Phi^{n\,\Delta t}(x_0)$, taken at a fixed sampling interval $\Delta t > 0$.
Outputs
  1. Mean forecast $\bar{f}_0, \dots, \bar{f}_{J_\mathrm{f}} \in \mathbb{R}$ at lead time $\tau_0, \dots, \tau_{J_\mathrm{f}}$ with $\tau_j = j\, \Delta t$.
  2. Forecast uncertainty $\sigma_0, \dots, \sigma_{J_\mathrm{f}} \in \mathbb{R}$ at lead time $\tau_0, \dots, \tau_{J_\mathrm{f}}$.
  3. Forecast probability vectors $\boldsymbol{p}_0, \dots, \boldsymbol{p}_{J_f} \in \mathbb{R}^M$ with $\boldsymbol{p}_j = (p_{0j}, \dots, p_{Mj})$. $p_{mj}$ is the probability that, at lead time $\tau_j$, the forecast observable $f$ lies in spectral bin $S_m$.
  4. Posterior density matrix $\boldsymbol{\rho}^{(+)} \in \mathbb{M}_L$ at time $J_\mathrm{o}\, \Delta t$.
Steps
  1. Set $\boldsymbol{U}^{(0)} = \mathrm{Id}$.
  2. For each $j \in \{0, \dots, J\}$ compute the time-evolved density matrix $\boldsymbol{\rho}_j = \boldsymbol{\sigma}_j / C_j$ with $\boldsymbol{\sigma}_j = (\boldsymbol{U}^{(j)})^\top \boldsymbol{\rho} \boldsymbol{U}^{(j)}$ and $C_j = \operatorname{tr} \boldsymbol{\sigma}_j$.
  3. Return: The mean forecasts $\bar{f}_j = \operatorname{tr}(\boldsymbol{\rho}_j \boldsymbol{A})$ for $j \in \{0, \dots, J_\mathrm{f}\}$.
  4. Return: The forecast uncertainties $\sigma_j = (\operatorname{tr}(\boldsymbol{\rho}_j \boldsymbol{A}^2) - \bar{f}_j^2)^{1/2}$ for $j \in \{0, \dots, J_\mathrm{f}\}$.
  5. Return: The probability vectors $\boldsymbol{p}_j$ with $p_{mj} = \operatorname{tr}(\boldsymbol{\rho}_j E_m)$ for $m \in \{0, \dots, M-1\}$ and $j \in \{0, \dots, J_\mathrm{f}\}$.
  6. Compute the effect matrix $\boldsymbol{E}_y = \boldsymbol{F}'(y)$.
  7. Return: The posterior density matrix
       $$\boldsymbol{\rho}^{(+)} = \frac{\boldsymbol{E}_y \boldsymbol{\rho}_{J_\mathrm{o}} \boldsymbol{E}_y}{\operatorname{tr}(\boldsymbol{E}_y \boldsymbol{\rho}_{J_\mathrm{o}} \boldsymbol{E}_y)}.$$
```

Algorithm S10 describes the QMDA forecast of $f : X \to \mathbb{R}$ via the quantum operation $\mathcal{U}_{N,L}^{(q)}$ applied to the corresponding projected multiplication operators $\pi_{L,N} \hat{f}_N$. Algorithm S11 specializes the forecasting procedure to pure (vector) states, which allow representation of density matrices $\boldsymbol{\rho} \in \mathbb{M}_L$ by their corresponding state vectors, $\boldsymbol{\rho} = \boldsymbol{\xi} \boldsymbol{\xi}^\dagger$ where $\boldsymbol{\xi}$ is a unit vector in $\mathbb{C}^L$. Algorithms S10 and S11 also include the analysis step based on effect-valued maps, which we describe in section 2.J.

```
Algorithm S11 QMDA forecast–analysis step, specialized to pure states.
Inputs
  1. Forecast timesteps $J_\mathrm{f} \in \mathbb{N}$; observation timesteps $J_\mathrm{o} \in \mathbb{N}$.
  2. Koopman matrices $\boldsymbol{U}^{(1)}, \dots, \boldsymbol{U}^{(J)} \in \mathbb{M}_L$ from Algorithm S9 with $J = \max\{J_\mathrm{f}, J_\mathrm{o}\}$.
  3. Spectral bins $S_0, \dots, S_{M-1} \subseteq R$ from Algorithm S8.
  4. Forecast observable $\boldsymbol{A} \in \mathbb{M}_L$ and spectral projectors $\boldsymbol{E}_0, \dots, \boldsymbol{E}_{M-1} \in \mathbb{M}_L$ from Algorithm S7.
  5. Matrix-valued effect $\boldsymbol{F} : Y \to \mathbb{M}_L$ from Algorithm S12.
  6. Initial state vector $\boldsymbol{\xi} \in \mathbb{C}^L$.
  7. Observation $y \in Y$ at time $J_\mathrm{o}\, \Delta t$.

Require: All $L \times L$ matrices are representations of operators in $\mathfrak{B}_{L,N}$ in the same data-driven basis $\{\phi_{l,N}\}$ from Algorithm S3. All training data are induced by the same sequence of (unknown) time-ordered states $x_0, \dots, x_{N-1} \in X$ with $x_n = \Phi^{n\, \Delta t}(x_0)$, taken at a fixed sampling interval $\Delta t > 0$.

Outputs
  1. Mean forecast $\bar{f}_0, \dots, \bar{f}_{J_\mathrm{f}} \in \mathbb{R}$ at lead time $\tau_0, \dots, \tau_{J_\mathrm{f}}$ with $\tau_j = j\, \Delta t$.
  2. Forecast uncertainty $\sigma_0, \dots, \sigma_{J_\mathrm{f}} \in \mathbb{R}$ at lead time $\tau_0, \dots, \tau_{J_\mathrm{f}}$.
  3. Forecast probability vectors $\boldsymbol{p}_0, \dots, \boldsymbol{p}_{J_f} \in \mathbb{R}^M$ with $\boldsymbol{p}_j = (p_{0j}, \dots, p_{Mj})$. $p_{mj}$ is the probability that, at lead time $\tau_j$, the forecast observable $f$ lies in spectral bin $S_m$.
  4. Posterior state vector $\boldsymbol{\xi}^{(+)} \in \mathbb{C}^L$ at time $J_\mathrm{o}\, \Delta t$.

Steps
  1. Set $\boldsymbol{U}^{(0)} = \mathrm{Id}$.
  2. For each $j \in \{0, \dots, J\}$ compute the time-evolved state vector $\boldsymbol{\xi}_j = \boldsymbol{u}_j / \|\boldsymbol{u}_j\|_2$ with $\boldsymbol{u}_j = (U^{(j)})^\top \boldsymbol{\xi}$.
  3. Return: The mean forecasts $\bar{f}_j = \boldsymbol{\xi}_j^\dagger \boldsymbol{A} \boldsymbol{\xi}_j$ for $j \in \{0, \dots, J_\mathrm{f}\}$.
  4. Return: The forecast uncertainties $\sigma_j = (\boldsymbol{\xi}_j^\dagger \boldsymbol{A}^2 \boldsymbol{\xi}_j - \bar{f}_j^2)^{1/2}$ for $j \in \{0, \dots, J_\mathrm{f}\}$.
  5. Return: The probability vectors $\boldsymbol{p}_j$ with $p_{mj} = \boldsymbol{\xi}_j^\dagger \boldsymbol{E}_m \boldsymbol{\xi}_j$ for $m \in \{0, \dots, M-1\}$ and $j \in \{0, \dots, J_\mathrm{f}\}$.
  6. Compute the effect matrix $\boldsymbol{E}_y = \boldsymbol{F}(y)$.
  7. Return: The posterior state vector
       $$\boldsymbol{\xi}^{(+)} = \frac{\boldsymbol{E}_y \boldsymbol{\xi}_{J_\mathrm{o}}}{\|\boldsymbol{E}_y \boldsymbol{\xi}_{J_\mathrm{o}}\|_2}.$$
```

I. Channel consistency. The asymptotic consistency of the quantum operations $\mathcal{U}_{L,N}^{(q)}$ readily follows from the results established in the previous subsections.

First, given $A \in \mathfrak{B}$ and $A_L = \mathbf{\Pi}_L A$, it is a direct consequence of the strong convergence of $U_L^{(t)}$ to U^t and of A_L to A, together with the uniform boundedness of these operators, that $\mathcal{U}_L^{(t)} A_L$ converges to $\mathcal{U}^t A$ in the strong topology of $\mathfrak{B}$, i.e.,

$$\lim_{L\to\infty} (\mathcal{U}_L^{(t)} A_L) f = (\mathcal{U}^t A) f, \quad \forall f \in H. \tag{S22}$$

As a result, for every normal state $\omega_\rho \in S_*(\mathfrak{B})$ induced by a density operator $\rho \in \mathfrak{B}_*$, Eq. (S15) yields

$$\lim_{L\to\infty} (\mathcal{P}_L^{(t)} \omega_{\rho_L}) A_L = \lim_{L\to\infty} \omega_{\rho_L} (\mathcal{U}_L^{(t)} A_L) = \omega_\rho A, \tag{S23}$$

where $\rho_L = \mathbf{\Pi}_L \rho / \operatorname{tr}(\mathbf{\Pi}_L \rho) \in \mathfrak{B}_L$ are the projected density operators induced by ρ. Eq. (S23) establishes the consistency of the quantum operations $\mathcal{U}_L^{(t)}$ with the channel $\mathcal{U}^{(t)}$ in the infinite-dimension limit.

Next, turning to the consistency of $\mathcal{U}_{L,N}^{(q)}$ in the large-data ($N \to \infty$) limit, we restrict attention to states in the subset $S_C(\mathfrak{B}) \subset S_*(\mathfrak{B})$ from subsection 2.E.2 and elements $A \in \mathfrak{B}$ satisfying Eq. (S9) and Eq. (S10). We also consider evolution times $t_q = q\,\Delta t$ with $q \in \mathbb{N}$. Under these assumptions, it follows from Eq. (S17) that for μ-a.e. initial state x_0,

$$\lim_{L\to\infty} \lim_{N\to\infty} (\mathcal{P}_{L,N}^{(q)} \omega_{\rho_{L,N}}) A_{L,N} = \lim_{L\to\infty} (\mathcal{P}_L^{(t_q)} \omega_{\rho_L}) A_L = (\mathcal{P}^t \omega_\rho) A. \tag{S24}$$

Eq. (S24) holds, in particular, for data-driven, projected multiplication operators $A_{L,N} = \pi_{L,N} \hat{f}_N$ associated with continuous functions $f : X \to \mathbb{R}$ and pure states $\hat{\rho}_{N,L}$ induced by continuous state vectors, as in the QMDA experiments described in the main text.

J. Effect-valued feature map. In this subsection, we describe the construction and properties of the effect-valued feature map $\mathcal{F} : Y \to \mathcal{E}(\mathfrak{B})$ and its finite-rank counterparts, $\mathcal{F}_L : Y \to \mathcal{E}(\mathfrak{B}_L)$ and $\mathcal{F}_{L,N} : Y \to \mathcal{E}(\mathfrak{B}_{L,N})$, used in the analysis step of QMDA. As stated in the main text, we build these maps using a continuous, symmetric kernel on observations space $\psi : Y \times Y \to [0,1]$ and a family of data-dependent symmetric, continuous kernels $\psi_N : Y \times Y \to [0,1]$ with $N \in \mathbb{N}$, such that, as $N \to \infty$, the pullback kernels $w_N : X \times X \to [0,1]$ with $w_N(x,x') = \psi_N(h(x), h(x'))$ converge uniformly to $w : X \times X \to [0,1]$ with $w(x,x') = \psi(h(x), h(x'))$.

As a concrete example, in the L96 and ENSO experiments described in the main text we employ variable-bandwidth, bump kernels on $Y = \mathbb{R}^d$,

$$\psi_N(y,y') = \eta_{\text{bump}}\left(\frac{d(y,y')}{\epsilon \sqrt{b_N(y) b_N(y')}} \right). \tag{S25}$$

In Eq. (S25), $\eta_{\text{bump}} : \mathbb{R} \to \mathbb{R}$ is the bump function

$$\eta_{\text{bump}}(u) = \begin{cases} e^{-1/(1-u^2)}, & u \in (-1,1), \\ 0, & \text{otherwise,} \end{cases} \tag{S26}$$

$d : Y \times Y \to \mathbb{R}_+$ is the Euclidean distance, $b_N : Y \to \mathbb{R}_+$ is a bandwidth function obtained by applying Algorithm S2 to the training data $y_0, \dots, y_{N-1} \in Y$, and $\epsilon > 0$. As in section 2.B, we set $\epsilon = s_\psi \epsilon_*$, where $s_\psi > 0$ is a scaling parameter and ϵ_* is the optimal bandwidth parameter determined via Algorithm S6. Note that b_N and ϵ in Eq. (S25) are different from the bandwidth function and bandwidth parameter used in Eq. (S4) and Eq. (S5) (the former are based on data $y_n \in Y$ whereas the latter are based on data $z_n \in Z$).

The choice of kernel in Eq. (S25) is motivated by the fact that the classical Bayesian analysis step of data assimilation can be modeled as an effect-valued map $\check{F} : Y \to \mathcal{E}(\mathfrak{A})$ for the abelian algebra $\mathfrak{A} = L^\infty(X,\mu)$, where $\check{F}(y) = \chi_{Y^{-1}(y)}$ and $\chi_S : X \to \{0,1\}$ denotes the characteristic function of a set $S \subseteq X$. The effect-valued map $F_N : Y \to \mathcal{E}(\mathfrak{A})$ induced by the kernel in Eq. (S25), $F_N(y) = \psi_N(y, h(\cdot))$, can be thought of as a smoothed version of $\check{F}$. In L96 and ENSO experiments not reported here, we found that using a fixed-bandwidth kernel in the analysis step (i.e., setting $b_N(y) = b_N(y') = 1$ in Eq. (S25) led to a noticeable reduction of forecast skill, particularly in the higher-dimensional ENSO case. We also ran experiments using the Gaussian shape function η_{gauss} (as opposed to η_{bump}) in the definition of the kernel ψ_N, using either of the variable- or fixed-bandwidth versions. The results were generally comparable to those reported in the main text, though we found that the bump kernel did provide a modest amount of skill improvement over the Gaussian kernel.

Let $F : Y \to \mathcal{E}(\mathfrak{A})$ be the effect-valued map associated with the kernel ψ,

$$\psi(y,y') = \eta_{\text{bump}}\left(\frac{d(y,y')}{\epsilon \sqrt{b(y) b(y')}} \right),$$

where the bandwidth function $b : Y \to \mathbb{R}_+$ is the uniform limit of b_N (note that we will not need this map in actual numerical applications). As described in the main text, in the setting of the non-abelian algebra $\mathfrak{B}$, we promote F to an operator-valued map $\mathcal{F} : Y \to \mathcal{E}(\mathfrak{B})$, where $\mathcal{F}(y) = \pi(F(y))$. Moreover, we introduce projected and data-driven versions of these maps, given by $\mathcal{F}_L : Y \to \mathcal{E}(\mathfrak{B}_L)$ and $\mathcal{F}_{L,N} : Y \to \mathcal{E}(\mathfrak{B}_{L,N})$, respectively, where $\mathcal{F}_L = \pi_L \circ \mathcal{F}$, $\mathcal{F}_{L,N} = \pi_{L,N} \circ \hat{\mathcal{F}}_N$, and $\hat{F}_N : Y \to \mathcal{E}(\hat{\mathfrak{A}}_N)$ is given by restriction of F_N on the set of training states X_N, $\hat{F}_N(y) = \iota_N(F_N(y))$. State conditioning (analysis) based on these maps has the following consistency properties.

Proposition 6. *Let ω_ρ be a state of $\mathfrak{B}$ in $S_C(\mathfrak{B})$ and y an observation in Y. Let $\boldsymbol{\rho}^{(+)}_{L,N}$ and $\boldsymbol{\rho}^{(+)}_L$ be the matrix representations of the conditional states $\omega_{\rho_{L,N}}|_{\mathcal{F}_{L,N}(y)} \in S(\mathfrak{B}_{L,N})$ and $\omega_{\rho_L}|_{\mathcal{F}_L(y)}$ obtained via Eq. (7) in the $\{\phi_{l,N}\}$ and $\{\phi_l\}$ bases of $H_{L,N}$ and H_L, respectively. Then the following hold.*

(a) For μ-a.e. initial state $x_0 \in X$, $\lim_{N\to\infty} \boldsymbol{\rho}^{(+)}_{L,N} = \boldsymbol{\rho}^{(+)}_L$.
(b) As $L \to \infty$, $\omega_{\rho_L}|_{\mathcal{F}_L(y)}$ converges to $\omega_\rho|_{\mathcal{F}(y)}$ in the trace norm topology of $S_*(\mathfrak{B})$.

Proof. We use an auxiliary map $\tilde{\mathcal{F}} : Y \to \mathfrak{C}$, defined as $\tilde{\mathcal{F}}(y) = \tilde{\pi}(F^{1/2}(y))$. For each $y \in Y$, we have

$$\iota \circ (\tilde{\mathcal{F}}(y)) = (\mathcal{F}(y)) \circ \iota$$

by construction. Moreover, by our assumed uniform convergence of w_N to w, we have that for every $f \in C(X)$, the residual

$$R_N f = \left(\iota_N \circ (\tilde{\mathcal{F}}(y))\right) f - \left(\hat{\mathcal{F}}_N(y) \circ \iota_N\right) f$$

has vanishing $\hat{H}_N$ norm as $N \to \infty$, for μ-a.e. initial condition x_0. Thus, Eq. (S9) and Eq. (S10) hold for $A = \mathcal{F}(y)$, $\tilde{A} = \tilde{\mathcal{F}}(y)$, and $A_N = \hat{\mathcal{F}}_N(y)$, and correspondingly Eq. (S14), Lemma 2, and Corollary 3 also hold. The claims of the proposition follow. □

Computationally, a drawback of using $\mathcal{F}_{L,N}$ for state conditioning is that evaluation of Eq. (7) requires the square root $\sqrt{\mathcal{F}_{L,N}(y)}$. Specifically, we have $\omega_{\rho_{L,N}}|_{\mathcal{F}_{L,N}(y)} = \omega_{\rho^{(+)}_{L,N}}$, where

$$\rho^{(+)}_{L,N} = \frac{\sqrt{\mathcal{F}_{L,N}(y)}\rho_{L,N}\sqrt{\mathcal{F}_{L,N}(y)}}{\operatorname{tr}(\sqrt{\mathcal{F}_{L,N}(y)}\rho_{L,N}\sqrt{\mathcal{F}_{L,N}(y)})}, \qquad \text{[S27]}$$

and computing $\sqrt{\mathcal{F}_{L,N}(y)}$ requires computing the square root of the matrix $\boldsymbol{F}_{L,N}(y) = [\langle \phi_{i,N}, \mathcal{F}_{L,N}(y)\phi_{j,N}\rangle_N]_{i,j=0}^{L-1}$ representing $\mathcal{F}_{L,N}(y)$ in the $\{\phi_{l,N}\}$ basis of $H_{L,N}$. To avoid having to perform this expensive operation at every observational update, in applications we replace $\sqrt{\mathcal{F}_{L,N}(y)}$ by $\mathcal{F}'_{L,N}(y)$, where $\mathcal{F}'_{L,N} : Y \to \mathcal{E}(\mathfrak{B}_{L,N})$ is the effect-valued map defined as

$$\mathcal{F}'_{L,N}(y) = \pi_{L,N}(\hat{F}^{1/2}_N(y)).$$

The construction of this map is described in Algorithm S12. Using $\mathcal{F}'_{L,N}$, the update of a density operator $\rho_{L,N} \in \mathfrak{B}_{L,N*}$ given an observation $y \in Y$ becomes (cf. Eq. (S27))

$$\rho^{(+)}_{L,N} = \frac{\mathcal{F}'_{L,N}(y)\boldsymbol{\rho}_{L,N}\mathcal{F}'_{L,N}(y)}{\operatorname{tr}(\mathcal{F}'_{L,N}(y)\boldsymbol{\rho}_{L,N}\mathcal{F}'_{L,N}(y))}. \qquad \text{[S28]}$$

In the limit of $L \to \infty$ after $N \to \infty$, updating via Eq. (S28) consistently recovers $\omega_\rho|_{\mathcal{F}(y)}$ analogously to Proposition 6. See step 7 of Algorithm S10 for the matrix function that implements Eq. (S28) in the $\{\phi_{l,N}\}$ basis of $H_{L,N}$. Step 7 of Algorithm S11 specializes the state update procedure to pure states.

Algorithm S12 Effect-valued feature map.

Inputs

1. Kernel function $\kappa : Y \times Y \to [0, 1]$.
2. Training data $y_0, \dots, y_{N-1} \in Y$.
3. Basis vectors $\boldsymbol{\phi}_0, \dots, \boldsymbol{\phi}_{L-1}$ from Algorithm S3.

Require: The training data z_n used in the computation of $\boldsymbol{\phi}_l$ are generated by the same dynamical states $x_n \in X$ underlying f_n, i.e., $z_n = z(x_n)$ and $f_n = f(x_n)$.

Outputs

1. Matrix-valued map $\boldsymbol{F}'_{L,N} : Y \to \mathbb{M}_L$ representing the effect-valued function $\mathcal{F}'_{L,N} : Y \to \mathcal{E}(\mathfrak{B}_{L,N})$.

Steps

1. Construct the feature map $\boldsymbol{f}'_N : Y \to \mathbb{R}^N$ where $\boldsymbol{f}'_N(y) = (\psi^{1/2}_N(y, y_0), \dots, \psi^{1/2}_N(y, y_{N-1}))^\top$.
2. **Return:** The function $\boldsymbol{F}'_{L,N} : Y \to \mathbb{M}_L$, where $\boldsymbol{F}'_{L,N}(y) = \boldsymbol{E} = [E_{ij}]_{i,j=0}^{L-1}$ and $E_{ij} = \boldsymbol{\phi}_i^\top(\boldsymbol{f}'_N(y) \odot \boldsymbol{\phi}_j)/N$.

K. Computational cost. The training data requirements and computational cost of QMDA are generally comparable to those of kernel methods for supervised machine learning.

K.1. Training phase. Letting d_Z denote the dimension of the training data space Z, the brute-force computation cost of forming the $N \times N$ kernel matrix $\hat{\boldsymbol{K}}_N$ via Algorithm S3 is $O(d_Z N^2)$. As mentioned in section 2.B, if N is sufficiently large so that the storage and arithmetic cost associated with $\hat{\boldsymbol{K}}_N$ is prohibitive, we approximate $\hat{\boldsymbol{K}}_N$ by a sparse matrix that contains the $\hat{k}_{\text{nn}}$ largest elements of $\hat{\boldsymbol{K}}_N$ in each row. The storage cost and matrix–vector multiplication cost for $\boldsymbol{K}_N$ then become $O(\hat{k}_{\text{nn}} N)$. The arithmetic cost of forming $\hat{\boldsymbol{K}}_N$ can be reduced to $O(N \log N)$ when d_Z and $\hat{k}_{\text{nn}}$ are sufficiently small using randomized approximate nearest neighbor algorithms, e.g., (18). The datasets used in this study were sufficiently small so that all of our experiments were performed without using nearest-neighbor truncation, although in separate calculations we have verified that setting $\hat{k}_{\text{nn}} \simeq 0.1N$ is sufficient to produce good-quality eigenfunctions at a significantly lower computational cost.

We compute the basis vectors $\phi_{l,N}$ associated with $\hat{\boldsymbol{K}}_N$ using iterative SVD solvers (e.g., svds in MATLAB). The cost of this computation depends on the spectral properties of $\hat{\boldsymbol{K}}_N$ and the number L of requested basis vectors, but generally scales linearly with $\hat{k}_{\text{nn}}$ and N. Once the basis $\{\phi_{l,N}\}_{l=0}^{L-1}$ has been computed, we form the $L \times L$ matrices $\boldsymbol{A}_{L,N}$ and $\boldsymbol{U}_{L,N}^{(q)}$ representing the projected forecast observable and Koopman operator via Algorithms S7 and S9, respectively, each with an $O(NL^2)$ computational cost. We also compute the full eigendecomposition of $\boldsymbol{A}_{L,N}$ (see Step 2 of Algorithm S7) at an $O(L^3)$ cost.

K.2. Data assimilation phase. The computational complexity of the data assimilation phase of QMDA depends on whether one employs mixed states (Algorithm S10) or pure states (Algorithm S11). In both cases, the cost of evaluating the matrix-valued feature map $\boldsymbol{F}_{L,N}$ is $O(dNL^2)$, where d is the dimension of the observation space Y. This is the only step in the data assimilation phase whose cost depends on the amount of training data N. If mixed states are used, the cost of the forecast and analysis steps is dominated by matrix–matrix multiplication of $L \times L$ matrices, and is thus $O(L^3)$ (independently of N). In the case of pure states, these operations are replaced by matrix–vector products, which leads to a reduction of cost to $O(L^2)$. More generally, the computational cost of the forecast and analysis steps with rank-r density matrices is $O(rL^2)$.

In applications, it is desirable to work with prediction algorithms where the out-of-sample evaluation cost increases slowly with training size N, or is even independent on N. This motivates exploring approximations of the matrix-valued feature map $\boldsymbol{F}_{L,N}$ with a more favorable scaling in N than the linear scaling of the brute-force approach. To that end, possible approaches include the randomized Nyström method for streaming PCA (19) and landmark methods for manifold learning (20, 21). The recent paper (22) developed a streaming formulation of the kernel analog forecast technique (23) using random Fourier feature methods (24) for kernel approximation in conjunction with the randomized Nyström method. This approach was found to exhibit similar forecast skill to the brute-force approach employed in ref. (25) for predicting the slow variables of the L96 multiscale system at a cost which is independent of the training data. As future work, it would be fruitful to explore applications of these methodologies in the context of QMDA.

3. Quantum circuit implementation

In this section, we describe the construction of the quantum circuit in Fig. 6 that implements the analysis and forecast steps of QMDA. The circuit is composed of three parameterized unitary transformations T_{init}, T_{K}, and T_{rot} acting on the $\mathfrak{n}$-qubit Hilbert space $\mathbb{B}_{\mathfrak{n}}$, whose role is to perform state initialization, Koopman evolution, and eigenbasis rotation, respectively, as discussed in the main text. Our numerical implementation is based on the Qiskit Python library (26), and can be found in the Jupyter notebook /examples/l96Multiscale_qmda/l96MultiscaleQmda.ipynb contained in the GitHub repository https://dg227.github.io/NLSA.

A. Initialization. Suppose that at time $t_n = n\,\Delta t$ the matrix mechanical data assimilation system is in the prior state $\omega_{n-1,1} \in S(\mathfrak{B}_{L,N})$. Suppose also that $\omega_{n-1,1}$ is a pure state with associated state vector $\xi_{n-1,1} \in H_{L,N}$. Then, given an observation $\hat{y}_n \in Y$, the transformation $T_{\text{init}}(\xi_{n-1,1}, \hat{y}_n) \in \mathfrak{M}_{\mathfrak{n}}$ is defined such that $\zeta_n := T_{\text{init}}(\xi_{n-1,1}, \hat{y}_n)\,|\boldsymbol{0}\rangle$ is the quantum computational representation of the state vector $\xi_n \in H_{L,N}$ associated with the posterior state $\omega_n = \omega_{n-1,1}|_{\mathcal{F}_{L,N}(\hat{y}_n)}$. That is, we have $\zeta_n = W_{L,N}\xi_n$, where $W_{L,N} : H_{L,N} \to \mathbb{B}_{\mathfrak{n}}$ is the unitary that maps the $\{\phi_l\}_{l=0}^{L-1}$ basis of $H_{L,N}$ to the quantum computational basis $\{|\boldsymbol{b}\rangle\}_{\boldsymbol{b}\in\{0,1\}^{\mathfrak{n}}}$ of $\mathbb{B}_{\mathfrak{n}}$ where $L = 2^{\mathfrak{n}}$ (see main text). Numerically, we implement $T_{\text{init}}(\xi_{n-1,1}, \hat{y}_n)$ using the QuantumCircuit.initialize method provided by Qiskit.

B. Koopman evolution. Let $j \in \mathbb{N}$ be the number of forecast timesteps corresponding to lead time $\tau_j = j\,\Delta t$. Assuming that the 1-step projected Koopman operator $U_{L,N}^{(1)} \in \mathfrak{B}_{L,N}$ is unitary, $T_{\text{K}}(j) \in \mathfrak{M}_{\mathfrak{n}}$ implements the state vector evolution under the iterated action of the adjoint of $U_{L,N}^{(1)}$ (the projected transfer operator) over j timesteps, i.e., $T_{\text{K}}(j) = W_L(U_{L,N}^{(1)})^{*j} W_L^*$. In Qiskit, we implement this operation using the UnitaryGate class.

As discussed in the main text (see also section 2.H), in general we cannot expect $U_{L,N}^{(1)}$ to be a unitary operator. Thus, in order to use it in a noise-free quantum circuit, where all operations must be unitary, we first pass its matrix representation $\boldsymbol{U}_{L,N}^{(1)}$ through the polar decomposition to extract its unitary part. That is, unless $\boldsymbol{U}_{L,N}^{(1)}$ happens to be unitary to sufficient numerical precision, we first compute a polar decomposition $\boldsymbol{U}_{L,N}^{(1)} = \tilde{\boldsymbol{U}}_{L,N}^{(1)}\tilde{\boldsymbol{P}}_{L,N}^{(1)}$ where $\tilde{\boldsymbol{U}}_{L,N}^{(1)}$ is unitary and $\tilde{\boldsymbol{P}}_{L,N}^{(1)} = (\boldsymbol{U}_{L,N}^{(1)}(\boldsymbol{U}_{L,N}^{(1)})^*)^{1/2}$, and then implement $T_{\text{K}}(j) = W_L(\tilde{U}_{L,N}^{(1)})^{*j} W_L^*$ where $\tilde{U}_{L,N}^{(1)} \in \mathfrak{B}_{L,N}$ is the linear operator represented by $\tilde{\boldsymbol{U}}_{L,N}^{(1)}$.

We should point out that aside from the polar decomposition step, the circuit implementation of QMDA differs from Algorithm S1 in that the time-τ_j Koopman operator is approximated iteratively as $(\tilde{U}_{L,N}^{(1)})^j$ as opposed to the direct approximation $U_{L,N}^{(j)}$ (see also Algorithm S9). While both approaches are asymptotically consistent in the limit of $L \to \infty$ after $N \to \infty$, at finite N and/or L they will generally yield different forecast skill.

C. Eigenbasis rotation. The eigenbasis rotation step T_{rot} is performed so that measurement at the output of the quantum circuit in Fig. 5 is consistent with measurement of the spectral measure $E_{L,N}$ of the projected multiplication operator $A_{L,N} = \pi_{L,N}\hat{f}_N$ associated with the forecast observable f, given the quantum state $\omega_{n,j} \in S(\mathfrak{B}_{L,N})$ for initialization time $t_n = n\,\Delta t$ and lead time $\tau_j = j\,\Delta t$.

Let $A_{L,N}u_{L,N,l} = a_{L,N,l}u_{L,N,l}$ be an eigendecomposition of $A_{L,N}$, where $\{u_{L,N,l}\}_{l=0}^{L-1}$ are orthonormal eigenvectors in $H_{L,N}$ and the eigenvalues $a_{L,N,l} \in \mathbb{R}$ are ordered in increasing order with multiplicities included. With such an eigendecomposition, we can express the spectral measure $E_{L,N}$ from Eq. (S19) as

$$E_{L,N}(S)g = \sum_{l:a_{L,N,l}\in S} \langle u_{L,N,l}, g\rangle_N u_{L,N,l}, \quad \forall S \in \mathcal{B}(\mathbb{R}), \quad \forall g \in H_{L,N}.$$

Next, letting $\boldsymbol{R}_{L,N} \in \mathbb{M}_L$ be the unitary matrix whose l-th column is the column vector representation $\boldsymbol{u}_{L,N,l}$ of $u_{L,N,l}$ in the $\{\phi_{l,N}\}$ basis, we define $R_{L,N} : H_{L,N} \to H_{L,N}$ as the unitary operator represented by that matrix and $\mathcal{R}_{L,N} : \mathfrak{B}_{L,N} \to \mathfrak{B}_{L,N}$ as $\mathcal{R}_{L,N}A = R^*_{L,N}AR_{L,N}$. We then have that $D_{L,N} := \mathcal{R}_{L,N}A_{L,N}$ is a diagonal operator with respect to the $\{\phi_{l,N}\}$ basis, i.e., $D_{L,N}\phi_{l,N} = a_{l,L,N}\phi_{l,N}$. The spectral measure $\tilde{E}_{L,N} : \mathcal{B}(\mathbb{R}) \to \mathfrak{B}_{L,N}$ of this operator satisfies

$$\tilde{E}_{L,N}(S)g = (\mathcal{R}_{L,N}E_{L,N}(S))g = \sum_{l:a_{L,N,l}\in S} \langle \phi_{L,N,l}, g\rangle_N \phi_{L,N,l}, \quad \forall S \in \mathcal{B}(\mathbb{R}), \quad \forall g \in H_{L,N}. \tag*{[S29]}$$

Given a state $\omega_\rho \in S(\mathfrak{B}_{L,N})$ induced by a density operator $\rho \in \mathfrak{B}_{L,N}$, we have $\omega_\rho A = \omega_{\mathcal{R}_{L,N}\rho}(\mathcal{R}_{L,N}A)$ for any $A \in \mathfrak{B}_{L,N}$. The latter, in conjunction with Eq. (S29) implies that evaluation of the spectral measure $E_{L,N}$ given the state ρ is equivalent to evaluation of $\tilde{E}_{L,N}$ given the state $\mathcal{R}_{L,N}\rho$; that is,

$$\omega_\rho(E_{L,N}(S)) = \omega_{\mathcal{R}_{L,N}\rho}(\tilde{E}_{L,N}(S)).$$

If ω_ρ is a pure state induced by a state vector $\xi \in H_{L,N}$, then $\omega_{\mathcal{R}_{L,N}\rho}$ is a pure state induced by the rotated vector $\tilde{\xi} = R^*_{L,N}\xi$, and the above identity reduces to

$$\langle \xi, E_{L,N}(S)\xi\rangle_N = \langle \tilde{\xi}, \tilde{E}_{L,N}(S)\tilde{\xi}\rangle_N. \tag*{[S30]}$$

We will now represent $\tilde{E}_{L,N}$ by an equivalent spectral measure associated with the quantum computational basis of $\mathbb{B}_{\mathfrak{n}}$. Recall that the quantum computational basis $\{|\boldsymbol{b}\rangle\}_{\boldsymbol{b}\in\{0,1\}^{\mathfrak{n}}}$ is indexed by binary strings $\boldsymbol{b} = (b_1, \dots, b_d)$ of length $\mathfrak{n}$. Associated with this basis is a spectral measure $\mathbb{E}_{\mathfrak{n}} : \Sigma(\{0,1\}^{\mathfrak{n}}) \to B(\mathbb{B}_{\mathfrak{n}})$ on the σ-algebra $\Sigma(\{0,1\}^{\mathfrak{n}})$ of all subsets of $\{0,1\}^{\mathfrak{n}}$ such that

$$\mathbb{E}_{\mathfrak{n}}(\hat{S})\hat{g} = \sum_{\boldsymbol{b}\in\hat{S}} \langle \boldsymbol{b}, g\rangle_{\mathfrak{n}} |\boldsymbol{b}\rangle, \quad \forall \hat{S} \in \Sigma(\{0,1\}^{\mathfrak{n}}), \quad \forall \hat{g} \in \mathbb{B}_{\mathfrak{n}}.$$

Measurement at the output of the circuit corresponds to measurement of $\mathbb{E}_{\mathfrak{n}}$; that is, given a state vector $\zeta \in \mathbb{B}_{\mathfrak{n}}$, we obtain a random binary string $\boldsymbol{b} \in \{0,1\}^{\mathfrak{n}}$ with probability $\langle \zeta, \mathbb{E}_{\mathfrak{n}}(\{\boldsymbol{b}\})\zeta\rangle_{\mathfrak{n}}$.

Given a binary string $\boldsymbol{b} \in \{0,1\}^{\mathfrak{n}}$, we let $\ell(\boldsymbol{b}) \in \{0, \dots, L-1\}$ denote the integer with binary representation $\boldsymbol{b}$, i.e., $\ell(\boldsymbol{b}) = \sum_{i=1}^{\mathfrak{n}} 2^{\mathfrak{n}-1}b_i$. To map a measurement of $\tilde{E}_{L,N}$ to a measurement of $\mathbb{E}_{\mathfrak{n}}$, we define the set function $\gamma : \mathcal{B}(\mathbb{R}) \to \Sigma(\{0,1\}^{\mathfrak{n}})$ such that $\gamma(S) = \{\boldsymbol{b} \in \{0,1\}^{\mathfrak{n}} : a_{L,N,\ell(\boldsymbol{b})} \in S\}$. In essence, γ represents an encoding of the eigenvalues $a_{L,N,l}$ by the binary representation of the index l. One can then directly verify that the spectral measures $\tilde{E}_{L,N}$ and $\mathbb{E}_{\mathfrak{n}}$ are related as

$$\mathcal{W}_{L,N}\tilde{E}_{L,N}(S) = \mathbb{E}_{\mathfrak{n}}(\gamma(S)), \quad \forall S \in \mathcal{B}(\mathbb{R}). \tag*{[S31]}$$

Combining Eq. (S30) and Eq. (S31), we get

$$\omega_\rho(E_{L,N}(S)) = \langle \tilde{\zeta}, \mathbb{E}_{\mathfrak{n}}(\gamma(S))\tilde{\zeta}\rangle_{\mathfrak{n}},$$

where $\tilde{\zeta} = W_{L,N}\tilde{\xi}$ is the quantum computational representation of state vector $\tilde{\xi}$. In summary, measurement of the spectral measure $E_{L,N}$ associated with the forecast observable given the state ρ is equivalent to measurement of the quantum computational spectral measure $\mathbb{E}_{\mathfrak{n}}$ given the quantum computational representation $\tilde{\zeta}$ of the *rotated* state vector $\tilde{\xi}$.

In the circuit of Fig. 6, the transformation T_{rot} is introduced to effect that rotation after the Koopman evolution stage. Specifically, we define $T_{\text{rot}} : \mathbb{B}_{\mathfrak{n}} \to \mathbb{B}_{\mathfrak{n}}$ as the unitary map whose matrix representation in the quantum computational basis is $\boldsymbol{R}_{L,N}$, i.e., the same as the representation of $R_{L,N}$ in the $\{\phi_{l,N}\}$ basis of $H_{L,N}$. With this definition, we have $W_{L,N}R_{L,N} = T_{\text{rot}}W_{L,N}$, so application of T_{rot} to the quantum computational representation $W_{L,N}\xi$ of a state vector $\xi \in H_{L,N}$ yields the quantum computational representation $W_{L,N}\tilde{\xi}$ of the rotated state vector $\tilde{\xi} = R_{L,N}\xi$, as desired. As with the implementation of the Koopman evolution T_{K}, we implement T_{rot} using Qiskit's UnitaryGate.

4. Forecast skill metrics

We assess the skill of the QMDA forecasts in the main text using normalized root mean square error (NRMSE) and anomaly correlation (AC) scores. Using throughout the notation of the main text and Algorithm S1, we perform forecasts of f at lead times $\tau_j = j\,\Delta t$ with $j \in \{0, \dots, J_\text{f} - 1\}$ given initial data $\hat{y}_0, \dots, \hat{y}_{\hat{N}-1} \in Y$. We let $\bar{f}_{n,j} \in \mathbb{R}$ be the mean forecast at lead time τ_j initialized with data $\hat{y}_n$. Given the values $\hat{f}_0, \dots, \hat{f}_{\hat{N}+J_\text{f}-1} \in \mathbb{R}$ of f in the verification interval, the error of the forecast mean $\bar{f}_{n,j}$ relative to the true value of f is $\varepsilon_{n,j} = \bar{f}_{n,j} - \hat{f}_{n+j}$.

Let $\mathbb{E}_N\, f$ and $\operatorname{var}_N f$ be the empirical mean and variance of f computed from the training data,

$$\mathbb{E}_N\, f = \int_X f\, d\mu_N = \frac{1}{N} \sum_{n=0}^{N-1} f_n, \quad \operatorname{var}_N f = \int_X (f - \mathbb{E}_N\, f)^2\, d\mu_N = \frac{1}{N} \sum_{n=0}^{N-1} (f_n - \mathbb{E}_N\, f)^2.$$

We define the NRMSE and AC scores for lead time τ_j as

$$\operatorname{NRMSE}(\tau_j) = \sqrt{\frac{1}{\hat{N} \operatorname{var}_N f} \sum_{n=0}^{\hat{N}-1} \varepsilon_{n,j}^2}, \quad \operatorname{AC}(\tau_j) = \frac{1}{\hat{N} \operatorname{var}_N f} \sum_{n=0}^{\hat{N}-1} (\bar{f}_{n,j} - \mathbb{E}_N\, f)(\hat{f}_{n+j} - \mathbb{E}_N\, f),$$

respectively.

NRMSE values close to 0 and AC values close to 1 indicate high forecast skill. NRMSE values approaching 1 indicate loss of skill as the expected forecast error is comparable to the standard deviation of the forecast observable. In climate dynamics applications, such as ENSO forecasting, AC = 0.6 or AC = 0.5 are commonly used thresholds indicating loss of skill.

5. Dataset description

In this section, we describe the properties of the L96 multiscale and CCSM4 datasets used in the experiments presented in the main text. A summary of the attributes of the datasets and the QMDA parameters used in our numerical experiments is provided in Table S3.

A. L96 multiscale. We integrate the L96 multiscale system in Eq. (13) in MATLAB using the built-in stiff solver `ode15s`, sampling the numerical trajectory every $\Delta t = 0.05$ model time units. We note that the use of a stiff solver is important for numerical accuracy due to the timescale separation between the x_k and $\mathsf{y}_{j,k}$ variables occurring at small ε. As noted in the main text, our training and test data are sampled on independent dynamical trajectories. The initial conditions for the trajectory $x_0, \dots, x_{N-1} \in \mathbb{R}^{J(K+1)}$ underlying the training data are $(\mathsf{x}_1, \dots, \mathsf{x}_K) = (1, 0, \dots, 0) \in \mathbb{R}^K$ and $(\mathsf{y}_{1,k}, \dots, \mathsf{y}_{J,k}) = (1, 0, \dots, 0) \in \mathbb{R}^J$ for each $k \in \{1, \dots, K\}$. Similarly, we initialize the test trajectory $\hat{x}_0, \dots, \hat{x}_{\hat{N}-1} \in \mathbb{R}^{J(K+1)}$ at $(\mathsf{x}_1, \dots, \mathsf{x}_K) = (1.2, 0, \dots, 0) \in \mathbb{R}^K$ and $(\mathsf{y}_{1,k}, \dots, \mathsf{y}_{J,k}) = (1.2, 0, \dots, 0) \in \mathbb{R}^J$ for each $k \in \{1, \dots, K\}$. Starting from these initial conditions, we let the two trajectories equilibrate to the attractor over a time interval of $10^4\, \Delta t = 500$ model time units before collecting the first samples, x_0 and $\hat{x}_0$.

B. CCSM4. We sample data every $\Delta t = 1$ month, spanning a 1,300-year period from a control integration of CCSM4 forced with fixed pre-industrial concentrations of greenhouse gases (27). Following ref. (28), the observation map $h : X \to Y \equiv \mathbb{R}^d$ returns the monthly averaged sea surface temperature (SST) field on the model's native ocean grid (of approximately 1°nominal resolution) over the Indo-Pacific longitude-latitude box 28°E–70°W, 30°S–20°N. The number of gridpoints within this domain (which corresponds to the observation space dimension) is $d = 44{,}414$. As the forecast observable $f : X \to \mathbb{R}$, we use the model's Niño 3.4 index—this is defined as the area-averaged SST anomaly over the domain 170°W–120°W, 5°S–5°N relative to a monthly climatology computed over the training period. Specifically, let $\tilde{h} : X \to \mathbb{R}^{\tilde{d}}$ denote the observable representing the SST field over the Niño 3.4 region. For each $m \in \{1, \dots, 12\}$, define the monthly climatology $\bar{h}^{(m)} \in \mathbb{R}^{\tilde{d}}$ as

$$\bar{h}^{(m)} = \frac{1}{N_\text{y}} \sum_{\substack{0 \le n \le N-1 \\ \text{month}(n) = m}} \tilde{h}(x_n),$$

where $\text{month}(n) := (n \mod 12) + 1$ is the calendar month associated with the n-th sample in the training data.

References

1. Berry T, Giannakis D, Harlim J (2015) Nonparametric forecasting of low-dimensional dynamical systems. *Phys. Rev. E.* 91:032915.
2. Giannakis D, Slawinska J, Zhao Z (2015) Spatiotemporal feature extraction with data-driven Koopman operators in *Proceedings of the 1st International Workshop on Feature Extraction: Modern Questions and Challenges at NIPS 2015*, Proceedings of Machine Learning Research, eds. Storcheus D, Rostamizadeh A, Kumar S. (PMLR, Montreal, Canada), Vol. 44, pp. 103–115.
3. Giannakis D (2019) Data-driven spectral decomposition and forecasting of ergodic dynamical systems. *Appl. Comput. Harmon. Anal.* 47(2):338–396.

4. Giannakis D (2019) Quantum mechanics and data assimilation. *Phys. Rev. E* 100:032207.
5. Das S, Giannakis D, Slawinska J (2021) Reproducing kernel Hilbert space compactification of unitary evolution groups. *Appl. Comput. Harmon. Anal.* 54:75–136.
6. von Luxburg U, Belkin M, Bousquet O (2008) Consitency of spectral clustering. *Ann. Stat.* 26(2):555–586.
7. Giannakis D (2021) Delay-coordinate maps, coherence, and approximate spectra of evolution operators. *Res. Math. Sci.* 8:8.
8. Coifman R, Hirn M (2013) Bi-stochastic kernels via asymmetric affinity functions. *Appl. Comput. Harmon. Anal.* 35(1):177–180.
9. Berry T, Harlim J (2016) Variable bandwidth diffusion kernels. *Appl. Comput. Harmon. Anal.* 40(1):68–96.
10. Coifman RR, Lafon S (2006) Diffusion maps. *Appl. Comput. Harmon. Anal.* 21:5–30.
11. Sriperumbudur BK, Fukumizu K, Lanckriet GR (2011) Universality, characteristic kernels and RKHS embedding of measures. *J. Mach. Learn. Res.* 12:2389–2410.
12. Sauer T, Yorke JA, Casdagli M (1991) Embedology. *J. Stat. Phys.* 65(3–4):579–616.
13. Grigor'yan A, Hu J, Lau KS (2014) Heat kernels on metric measure spaces in *Geometry and Analysis of Fractals*, Springer Proceedings in Mathematics and Statistics, eds. Feng DJ, Lau KS. (Springer, Berlin) Vol. 88, pp. 147–207.
14. Coifman RR, Shkolnisky Y, Sigworth FJ, Singer A (2008) Graph Laplacian tomography from unknown random projections. *IEEE Trans. Image Process.* 17(10):1891–1899.
15. Chatelin F (2011) *Spectral Approximation of Linear Operators*, Classics in Applied Mathematics. (Society for Industrial and Applied Mathematics, Philadelphia).
16. de Oliveira CR (2009) *Intermediate Spectral Theory and Quantum Dynamics*, Progress in Mathematical Physics. (Birkhäuser, Basel) Vol. 54.
17. Walters P (1981) *An Introduction to Ergodic Theory*, Graduate Texts in Mathematics. (Springer-Verlag, New York) Vol. 79.
18. Jones PW, Osipov A, Rokhlin V (2011) Randomized approximate nearest neighbors algorithm. *Proc. Natl. Acad. Sci.* 108(38):15679–15686.
19. Halko N, Martinsson PG, Tropp J (2011) Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* 53(2):217–288.
20. Long AW, Ferguson AL (2019) Landmark diffusion maps (L-dMaps): Accelerated manifold learning out-of-sample extension. *Appl. Comput. Harmon. Anal.* 47:190–211.
21. Shen C, Lin YT, Wu HT (2022) Robust and scalable manifold learning via landmark diffusion for long-term medical signal processing. *J. Mach. Learn. Res.* 23:1–30.
22. Giannakis D, Henriksen A, Tropp JA, Ward R (2022) Learning to forecast dynamical systems from streaming data. *SIAM J. Appl. Dyn. Syst.* In press.
23. Alexander R, Giannakis D (2020) Operator-theoretic framework for forecasting nonlinear time series with kernel analog techniques. *Phys. D* 409:132520.
24. Rahimi A, Recht B (2007) Random features for large-scale kernel machines in *Advances in Neural Information Processing Systems*, eds. Platt J, Koller D, Singer Y, Roweis S. (Curran Associates, Inc.), Vol. 20, pp. 1177–1184.
25. Burov D, Giannakis D, Manohar K, Stuart A (2021) Kernel analog forecasting: Multiscale test problems. *Multiscale Model. Simul.* 19(2):1011–1040.
26. Anis MS, , et al. (2021) Qiskit: An open-source framework for quantum computing.
27. CCSM (2010) Community Climate System Model Version 4 (CCSM4) data. https://www.earthsystemgrid.org/dataset/ucar.cgd.ccsm4.joc.b40.1850.track1.1deg.006.html. Accessed March 2019.
28. Wang X, Slawinska J, Giannakis D (2020) Extended-range statistical ENSO prediction through operator-theoretic techniques for nonlinear dynamics. *Sci. Rep.* 10:2636.