Learning Propagators for Sea Surface Height Forecasts Using Koopman Autoencoders

Andrew Brettin[1], Laure Zanna[1], and Elizabeth A. Barnes[2]

[1]New York University
[2]Colorado State University

October 04, 2024

Learning Propagators for Sea Surface Height Forecasts Using Koopman Autoencoders

Andrew E. Brettin[1], Laure Zanna[1], and Elizabeth A. Barnes[2]

[1]Courant Institute of Mathematical Sciences, New York University, New York, NY, USA
[2]Department of Atmospheric Science, Colorado State University, Fort Collins, CO, USA

Key Points:

- We train a neural network to learn a low-dimensional representation of sea surface height that facilitates regional predictions
- The approach can work well in situations where linear inverse models struggle, such as on daily-averaged data
- Reconstruction skill highlights sources of predictability, such as the low-latitudes for North Pacific daily sea surface heights

Corresponding author: Andrew Brettin, `brettin@cims.nyu.edu`

Abstract

Due to the wide range of processes impacting the sea surface height (SSH) on daily-to-interannual timescales, SSH forecasts are hampered by numerous sources of uncertainty. While statistical-dynamical methods like Linear Inverse Modeling have been successful at making forecasts, they often rely on assumptions that can be hard to satisfy given the nonlinear dynamics of the climate. Here, we train convolutional autoencoders with a dynamical propagator in the latent space to generate forecasts of SSH anomalies. Learning a nonlinear dimensionality reduction and the prediction timestepping together results in a propagator that produces better predictions for daily- and monthly-averaged SSH in the North Pacific and Atlantic than if the dimensionality reduction and dynamics are learned separately. The reconstruction skill of the model highlights regions in which better representation results in improved predictions: in particular, the tropics for North Pacific daily SSH predictions and the Caribbean Current for the North Atlantic.

Plain Language Summary

Forecasts of sea surface heights are impacted by numerous sources of uncertainty. While statistical methods for representing temporal changes in the climate system have been useful for making predictions, they often rely on assumptions that do not always hold due to the complex interactions in the climate system. Here, we make a machine learning model that learns a compressed representation of the climate system which facilitates sea surface height predictions. The learned compressed representation of the climate system results in better sea surface height predictions than would occur if the dimensionality reduction and prediction is done separately. Our machine learning model also points to regions where more accurately representing sea level can result in better regional-scale predictions.

1 Introduction

The large variety of processes impacting sea surface heights (SSH) on daily-to-interannual timescales implies that forecasts of SSH on these time horizons are hindered by numerous sources of uncertainty. SSH variability on these timescales is driven by factors including barotropic adjustment to wind stress (Hermans et al., 2022; Kamp et al., 2024; Vinogradova et al., 2007), local air-sea buoyancy fluxes (Cabanes et al., 2006; Gill & Niller, 1973), wind-driven Ekman pumping (Webb, 2021; Cabanes et al., 2006), changes in large-scale Sverdrup balance (Cabanes et al., 2006), advection of density anomalies (Piecuch & Ponte, 2011), Rossby waves (Chelton & Schlax, 1996; Calafat et al., 2018), buoyancy-driven changes in ocean circulation (Roberts et al., 2016), eddy variability due to baroclinic instability (Marques et al., 2022), and the hydrostatic depression of the ocean surface due to atmospheric pressure anomalies (Piecuch et al., 2016). Developing forecasts for SSH amid these numerous drivers thus presents a challenge.

Over the past few decades, statistical-dynamical methods have proven effective for developing forecasts directly from data. Forecasts generated using Linear Inverse Models (LIM, Penland (1989); Penland and Sardeshmukh (1995)) have had substantial success in predicting the large-scale evolution of geophysical fields on these timescales (Newman, Shin, & Alexander, 2011; Zanna, 2012; Fraser et al., 2019; Albers & Newman, 2021). The framework generally involves first applying dimensionality reduction to represent the system state using a low-dimensional state vector, and then determining a linear propagator using the time-lagged covariance statistics between the state variables. This approach is based on the assumption that the state evolution can be represented as the sum of slow, predictable, linear dynamics and fast, unpredictable, nonlinear dynamics modelled by Gaussian noise (Hasselmann, 1976). Despite the simplicity of such models, LIMs have demonstrated skill comparable to operational forecasting models in some cases (Albers & Newman, 2021; Shin & Newman, 2021; Richter et al., 2020).

One appealing aspect of LIMs is the simplified representation of the dynamics as a low-dimensional, linear propagator. While nonlinear dynamical systems can be chaotic, unpredictable, and nontrivial to solve, linear dynamical systems readily admit closed-form solutions and can be solved in a systematic manner. The eigenvalues of the propagator can be used to identify dominant timescales for the dynamics of the system as well as optimal initial conditions for producing anomaly growth (Penland & Sardeshmukh, 1995; von Storch et al., 1995; Vimont et al., 2014; Zanna, 2012). However, ensuring that the state evolution is plausibly described by a linear stochastic dynamical system is often challenging. Whether or not dynamics can be represented as such depends on the processes being represented and the temporal resolution of the data. The computed propagator typically depends on the time lag used to compute it, due to nonstationary statistics (Penland & Sardeshmukh, 1995), unrepresented processes (Penland & Ghil, 1993), fundamental deficiencies in representing dynamical systems using Markov models (DelSole, 2000), and sampling of intrinsic oscillatory modes of the system (Penland, 2019).

Another sensitivity lies in the application of dimensionality reduction. Clearly, the number of dimensions used to represent the state is a parameter (Newman, Alexander, & Scott, 2011). Additionally, the performance of a LIM may depend on the dimensionality reduction technique applied. Typically, Principal Component Analysis (PCA), also known as Empirical Orthogonal Function analysis in the geosciences, is used to reduce the dimensionality of the system (Hotelling, 1933; Pearson, 1901; Lorenz, 1956). However, the requirement that modes are orthogonal can be restrictive (Dommenget & Latif, 2002). Alternatively, neural network autoencoders can relax the assumptions of linearity and orthogonality to obtain more efficient low-dimensional embeddings (Kramer, 1991; Hinton & Salakhutdinov, 2006). Nevertheless, it is unclear whether a more efficient yet complex representation will result in better predictions.

Complementing the linear-stochastic dynamical systems framework in inverse modeling of the earth system is the burgeoning set of data-driven approaches based on the operator-theoretic perspective of nonlinear dynamics. Under Koopman operator theory, nonlinear dynamical systems are represented through the linear (but infinite-dimensional) Koopman operator, which advances measurements of the system through time (Koopman, 1931). Thus, obtaining low-dimensional representations of the Koopman operator is a key goal of data-driven dynamical systems modeling. For instance, Dynamic Mode Decomposition seeks to find the best-fit linear model that advances linear measurements of the system (Schmid, 2010); however, such linear measurements may be insufficient to capture the complexities of nonlinear systems. Therefore, recent deep-learning approaches have modified the autoencoder architecture to learn nonlinear transformations into latent spaces in which the dynamics are approximately linear (Mardt et al., 2018; Lusch et al., 2018; Champion et al., 2019; Yeung et al., 2019; Brunton & Kutz, 2022).

Here, we leverage the Koopman Autoencoder framework in Lusch et al. (2018) to construct a linear propagator for SSH prediction on daily-to-interannual timescales in the North Pacific and North Atlantic. We assess the forecasts made by this model relative to baselines in which the dimensionality reduction and propagator are learned separately. We examine the areas of reconstruction skill to interpret how the Koopman Autoencoder attains its performance.

2 Methods

2.1 Data

We use daily- and monthly-averaged simulated SSH fields from the Community Earth System Model, version 2 (CESM2) Large Ensemble dataset (LENS2, Rodgers et al. (2021); Danabasoglu et al. (2021)). The data is from the 250-year simulation period spanning 1850–2100, with radiative forcing following the historical record from 1850–2014 and the

CMIP6 SSP3–7.0 forcing scenario thereafter (Danabasoglu et al., 2020; O'Neill et al., 2016). Fields are detrended using a locally-fitted fifth-degree polynomial and deseasonalized by removing climatological daily averages.

Sea surface heights η are computed by

$$\eta(x, y, t) = \zeta(x, y, t) + \eta_{\text{ib}}(x, y, t) \tag{1}$$

where ζ is the dynamic sea level simulated by CESM2 and η_{ib} is the inverse barometer contribution to sea level (Ponte, 2006; Gregory et al., 2019), given by

$$\eta_{\text{ib}}(x, y, t) = -\frac{1}{\rho_0 g} p'_a(x, y, t). \tag{2}$$

Here, $p'_a(x, y, t) = p_a(x, y, t) - \frac{1}{A}\int_A p_a(x, y, t)\, dA$ is the sea level pressure deviation from the spatial average over the ocean area A at time t, $\rho_0 = 1025$ kg m^{-3} is the reference sea surface density (Smith et al., 2010; Fofonoff & Millard Jr, 1983), and $g = 9.81$ m s^{-1} is the acceleration due to gravity.

We use nine ensemble members, with seven members for training and one member for validation and testing. We focus on two regions: the North Pacific (15°S–60°N, 115°E–60°W) and the North Atlantic (5°−65°N, 60°W−0°E). For training, fields are standardized using the area-weighted mean and standard deviation averaged over all samples in the training set (LeCun et al., 2002). Locations corresponding to land points are masked with zeros.

2.2 Koopman Autoencoder

Figure 1 illustrates the Koopman Autoencoder (Lusch et al., 2018). The network functions as a propagator for a dynamical system with the entire SSH field as its state variable: it consumes input fields of SSH at a given timestep n (x_n) and outputs the predicted SSH field at the next timestep ($\hat{x}_{n+1}$). We use a timestep of one day for networks trained on daily averages and one month for networks trained on monthly averages.

We employ a convolutional architecture that is well-suited for the spatial fields comprising our system state (Fukushima, 1980; LeCun et al., 1989). The encoder E takes in the state vector x_n, extracts features using convolutional filters and transforms the inputs to a lower dimensional embedding z_n. Then, a linear layer L is applied to the latent embedding, functioning as a single propagation timestep. Finally, the decoder D transforms the encoded prediction back into the state space, using the state at the next timestep x_{n+1} as the target.

During training, parameters in the Koopman Autoencoder are adjusted through backpropagation (Rumelhart et al., 1986) to optimize a combination of different objective functions in accordance with Lusch et al. (2018).

1. The reconstruction error

$$\mathcal{L}_{\text{reconst}}(x_n) = \|x_n - D(E(x_n))\|^2_{2,w}, \tag{3}$$

where $\|\cdot\|_{2,w}$ is the area-weighted ℓ^2-norm (see Supporting Text S2). This loss ensures that the encoder and decoder learns a maximally-efficient representation of the SSH in the d-dimensional latent space.

2. The prediction error

$$\mathcal{L}_{\text{pred}}(x_n, \dots, x_{n+k}) = \frac{1}{k}\sum_{\ell=1}^{k} \|x_{n+\ell} - D(L^\ell E(x_n))\|^2_{2,w} \tag{4}$$

The norm $\|x_{n+1} - D(LE(x_n))\|^2_{2,w}$ indicates the prediction error incurred during a single propagation timestep. In practice, better predictions are obtained by

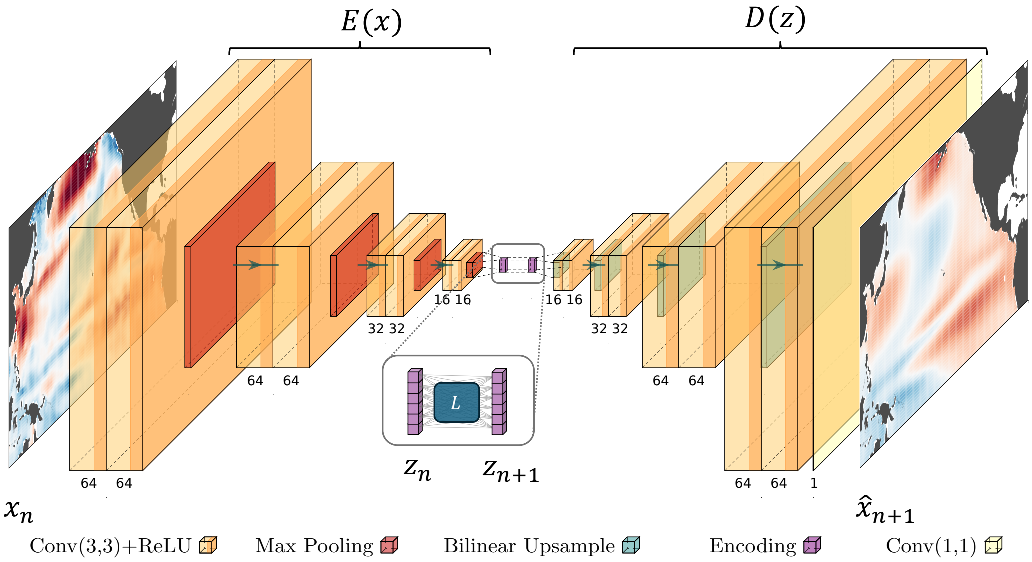

Figure 1. Koopman Autoencoder schematic. The encoder and decoder are denoted by the brackets labelled $E(x)$ and $D(z)$, respectively, and the inset shows the linear propagator. Yellow blocks indicate convolutional layers, and orange shading indicates ReLU activations. Red blocks indicate pooling layers, and green blocks indicate upsampling layers.

penalizing ℓ-timestep predictions for $\ell \in \{1, \dots, k\}$, where the ℓ-timestep prediction $\hat{x}_{n+\ell}$ is given by ℓ applications of the propagator L to the latent embedding: $\hat{x}_{n+\ell} = D(L^\ell E(x_n))$. We use $k = 20$ recurrent passes for all of our networks.

We also add a latent space prediction error

$$\mathcal{L}_{\text{linear}}(x_n, x_{n+1}) = \|LE(x_n) - E(x_{n+1})\|_2^2 \tag{5}$$

which further ensures that the linear prediction $\hat{z}_{n+1} = Lz_n = LE(x_n)$ approximates the latent state at the next timestep $z_{n+1} = E(x_{n+1})$. This term may be redundant as our propagator L is not equipped with activations, but is added for consistency with the proposed methodology of Lusch et al. (2018).

The net loss is given by

$$\mathcal{L}(x_n, \dots, x_{n+k}) = \lambda_1 \mathcal{L}_{\text{reconst}}(x_n) + \lambda_2 \mathcal{L}_{\text{pred}}(x_n, \dots, x_{n+k}) + \lambda_3 \mathcal{L}_{\text{linear}}(x_n, x_{n+1}) \tag{6}$$

where λ_1, λ_2, and λ_3 are hyperparameters. By optimizing this loss, the dimensionality reduction and the timestepping are learned together. This way, the dimensionality reduction is constructed in such a way that predictions are improved.

Separate networks are trained for each region and timescale. Full details about the training architecture and procedure are given in Supporting Text S1.

2.3 Baselines

We contrast the predictions made with our Koopman Autoencoders with baselines in which the dimensionality reduction and predictions are done separately. For dimensionality reduction, we consider Principal Component Analysis (PCA) and Convolutional

Autoencoders (CAE). For forecasting, we apply Damped Persistence (DP) and Linear Inverse Modeling (LIM). Prediction baselines are thus determined by combining the two techniques, and are denoted according to the dimensionality technique and propagator used, e.g. "PCA-LIM" or "CAE-DP."

2.3.1 Dimensionality reduction techniques

As a first baseline, PCA is applied to reduce the dimensionality of the state. In PCA, the data is linearly projected onto the d-dimensional subspace that maximizes the variance of the data. As a result, dimensions describing the data are linear and orthogonal, a restriction that may result in poor representation of nonlinear data manifolds.

As a nonlinear alternative to PCA, we also train Convolutional Autoencoders (CAE). Autoencoders generalize PCA by allowing for nonlinear transformations to a latent space and can learn more efficient representations than PCA (Kramer, 1991; Hinton & Salakhutdinov, 2006; Shamekh et al., 2023; Oommen et al., 2022). For the CAE, we use an encoder and decoder with the same architectures as those of the Koopman Autoencoder, and we train it with nearly identical hyperparameters (see Supporting Text S1).

2.3.2 Predictions in the latent space

We compare the forecasts made by the Koopman Autoencoder to Damped Persistence (DP, Lorenz (1973)). Given a latent state z_n, the prediction at lag τ is given by

$$\hat{z}_{n+\tau} = \mathbf{D}(\tau) z_n \tag{7}$$

where $\mathbf{D}(\tau)$ is a diagonal matrix whose entries give the autocorrelation of each of the latent variables at lag τ. The propagator $\mathbf{D}(\tau)$ is computed iteratively for each time lag by first selecting a training timescale τ_0, computing the lag-τ_0 autocorrelations $\mathbf{D}_0 = \mathbf{D}(\tau_0)$, and then defining $\mathbf{D}(\tau) = (\mathbf{D}_0)^{\tau/\tau_0}$. For a fair comparison with the Koopman Autoencoder, we set τ_0 by fitting DP models using $\tau_0 \in \{1, \ldots, k\}$ and selecting the model with the lowest average prediction error on timesteps 1 to k on the validation dataset.

We also explore predictions made by a Linear Inverse Model (LIM, Penland (1989)). The underlying assumption behind LIM is that the dynamics of a system can be well-represented as a linear dynamical system forced by noise:

$$\frac{dz}{dt} = \mathbf{A}z + \xi \tag{8}$$

where ξ is sampled from a Normal distribution. Then, the evolution matrix $\mathbf{A}$ can be estimated through an error minimization procedure as

$$\mathbf{A} = \frac{1}{\tau_0} \log\left(\mathbf{C}(\tau_0)\mathbf{C}(0)^{-1}\right) \tag{9}$$

where $\mathbf{C}(\tau) = \langle z(t+\tau) z^T(t) \rangle$ gives the time-τ lagged covariance (with angled brackets denoting a time average) and τ_0 is a fitted timescale. Predictions are then given by

$$\hat{z}_{n+\tau} = \mathbf{B}(\tau) z_n \tag{10}$$

with the propagator $\mathbf{B}(\tau)$ given by

$$\mathbf{B}(\tau) = \exp(\mathbf{A}\tau) = \exp\left[\frac{\tau}{\tau_0} \log\left(\mathbf{C}(\tau_0)\mathbf{C}(0)^{-1}\right)\right] \tag{11}$$

The covariance matrix is computed over all ensemble members, and again τ_0 is selected by fitting LIMs for $\tau_0 \in \{1, \ldots, k\}$ and selecting the model with the lowest error over timesteps 1 through k.

In order for a LIM to be valid, several conditions should be met. One basic criterion is that the learned propagator should be stable with decaying eigenvalues. (Similarly, the eigenvalues of the propagator learned by the Koopman Autoencoder should also decay.) Supporting Figure S1 verifies that all propagators considered in this study are stable. Another requirement is that the evolution matrix defined by Equation 9 must be independent of the time lag τ_0 used to compute it. However, this is a strong criterion to meet; common practice is to compute the matrix norm of the propagator $\|A\|_2$ for different τ_0 and to select a propagator based on a timescale τ_0 in which the matrix norm is relatively constant. Supporting Figure S2 shows the ℓ^2-matrix norms of the evolution matrix of the LIM baselines on the range $\tau_0 \in \{1, \ldots, k\}$; over this range, the matrix norm varies by over 300% for all of the regions and timescales considered.

3 Results

In this section, we compare the forecasts made by the Koopman Autoencoder to the other baselines. We use the Mean Square Error (MSE) and Pattern Correlation Coefficient (Legates & Davis, 1997) to assess our predictions, as well as MSE-based skill scores (Murphy, 1988). Metrics are defined explicitly in Supporting Text S2.

3.1 Evaluating prediction performance

Figure 2 compares the area-weighted prediction MSE and pattern correlation of SSH predictions of the Koopman Autoencoder to the baselines using $d = 20$ latent dimensions on forecast lead times τ of up to $\tau_{\max} = 120$ days (daily data) and $\tau_{\max} = 36$ months (monthly data). The CAE generally has the lowest reconstruction error for all dimensionality reduction techniques, beating PCA MSE by a margin of 2–4% at lead $\tau = 0$ for all regions and timescales except in the North Atlantic on monthly data (See Supporting Table S1). The Koopman Autoencoder has the worst reconstructions of all the methods considered: over all regions and timescales, MSE is on average 32% higher for the Koopman Autoencoder than for PCA. However, the better reconstruction error of the CAE does not necessarily result in better predictions. In fact, predictions made by applying propagators to CAE modes are often worse than predictions made using PCA for dimensionality reduction (e.g., using a DP propagator for North Pacific daily SSH, Figure 2a). In contrast, the Koopman Autoencoder generally results in better predictions than the baselines as measured by the area-weighted MSE and pattern correlation. Supporting Table S2 quantitatively summarizes the forecast performance of the models in Figure 2 through the skill score of the different prediction methods relative to PCA-DP, averaged over forecast leads up to $\tau_{\max}$. Skill of the models relative to PCA-DP depends significantly on the region and timescales considered but averaged over all regions and timescales, PCA-LIM has about 6.8% skill over PCA-DP, skill of CAE-LIM is slightly *worse* than PCA-LIM (6.4%), and skill of the Koopman Autoencoder is the highest (8.4%). In effect, by learning the dynamics and the dimensionality reduction together, the Koopman Autoencoder learns a nonlinear latent-space representation of the state that implicitly results in better SSH predictions.

The advantages of using the Koopman Autoencoder over, for example, PCA-LIM are more apparent on daily timescales than on monthly timescales. In the North Pacific, prediction skill of the Koopman Autoencoder relative to PCA-DP on daily-averaged data is 4.5% higher than that of PCA-LIM but is only 3.0% higher for monthly-averaged data; in the North Atlantic, Koopman skill is 1.1% higher than PCA-LIM on daily data but is 1.3% *lower* on monthly data. One potential reason is that the assumptions underlying LIM may be better satisfied for monthly averages than daily averages, because monthly-averaged fields smooth out small-scale, nonlinear features (Sardeshmukh & Sura, 2009; Stephenson et al., 2004). The Koopman Autoencoders also outperform PCA-LIM by a wider margin in the North Pacific than in the North Atlantic. This may be due to the fact that the inverse barometer component constitutes a larger share of the SSH variability in the North Atlantic region considered (about 71% in the North Atlantic on daily timescales vs 32% in the North Pacific; see Supporting Figure S3). This high-frequency variability may be well-represented by white noise, again underpinning the relative success of PCA-LIM.

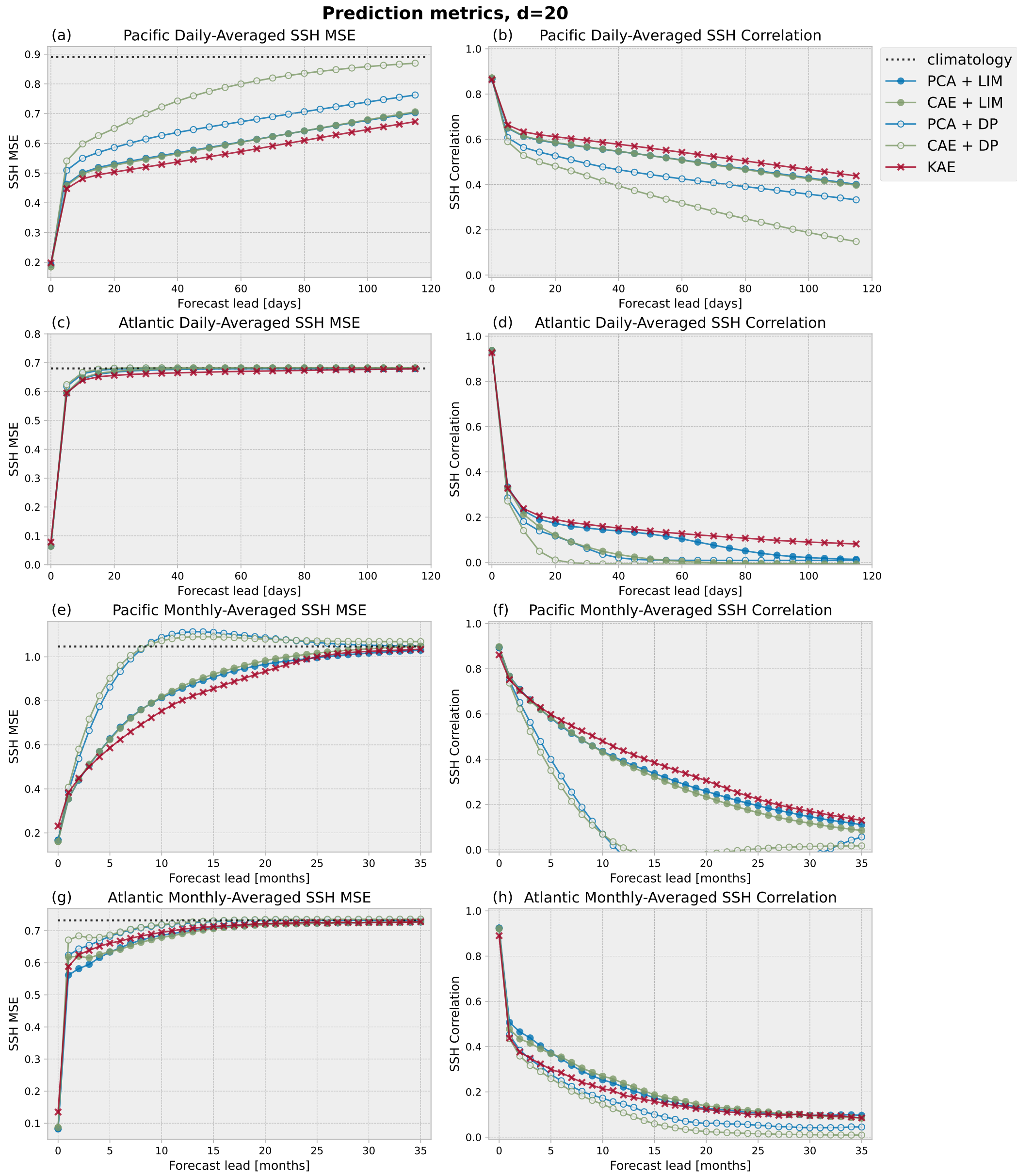

Figure 2. Forecast MSE and Pattern Correlation in the North Pacific and North Atlantic on daily and monthly timescales. Colors indicate dimensionality reduction techniques (red for the Koopman Autoencoder, blue for PCA, and light green for CAE), while markers indicate propagation techniques (x's for the Koopman Autoencoder, filled circles for LIM, and open circles for DP). The black dotted line indicates the climatological MSE of SSH.

3.2 Sensitivity to the number of dimensions

Both the dimensionality reduction and learned propagator's predictions may be sensitive to the dimensionality of the latent space. Figure 3 explores both of these sensitivities. Due to the computational cost of training each network, sensitivity is examined only in one region and timescale; we focus on forecasts of daily-averaged SSH in the North Pacific as the Koopman Autoencoder was shown to generate skillful predictions for these dynamics.

As shown in Figure 3a and Supporting Table S3, reconstruction performance improves as the number of latent dimensions is increased up to $d = 40$ for all dimensionality reduction techniques considered. Just as in Section 3.1, for any given number of dimensions, the CAE has the best reconstructions, outperforming PCA by 2–4%, while the Koopman Autoencoder has the worst reconstructions, with reconstruction MSE 1–13% higher than that of PCA.

Like the reconstruction skill, the predictions of the Koopman Autoencoder also improve as the dimensionality is increased, as shown in Figure 3b. Koopman operator theory suggests that this should be the case, as it states that infinitely many observables must be prescribed to guarantee a nonlinear dynamical system is fully determined. Nevertheless, the utility of using the Koopman Autoencoder for building propagators diminishes as the number of dimensions is increased. Figure 3c shows the domain-averaged prediction skill of the Koopman Autoencoder relative to PCA-LIM predictions using the same dimensionality. For all dimensionalities, the Koopman Autoencoder outperforms PCA-LIM forecasts up to $\tau = 120$ days; however, up to forecast leads of $\tau = 60$ days, the skill of the Koopman Autoencoder decreases as the dimensionality increases. Much of this seems to be simply because the Koopman Autoencoder becomes worse at reconstructions relative to PCA for higher latent dimensionalities (e.g., 1% higher MSE for $d = 10$ vs 13% higher MSE for $d = 40$; see Supporting Table S3). This suggests that the Koopman Autoencoder approach may be most useful for developing low-dimensional propagators.

3.3 Regions of skill

To understand how the Koopman Autoencoder attains its performance, Figure 4 shows the MSE-skill score of the Koopman Autoencoder relative to PCA-based propagators for daily SSH forecasts in the North Pacific and North Atlantic. We focus on PCA-based propagators because of the simplicity and interpretability of linear, orthogonal dimensionality reduction, which the CAE cannot afford. For example, due to the orthogonality of modes, applying damped persistence to the principal components results in purely local dampening of SSH at each location.

Figure 4a shows domain-averaged MSE skill scores for the Koopman Autoencoder and PCA-LIM relative to PCA-DP. Skill scores for the Koopman Autoencoder and PCA-LIM relative to PCA-DP start at about 0, increase to a maximum at a lead of about 30 days, and gradually taper for longer-term forecasts. However, the Koopman Autoencoder skill is much higher than that of PCA-LIM at all lags—by 72% at lead 5 days and by at least 47% for leads up to 120 days.

Figures 4c-e and 4f-h show the regional variations of Koopman Autoencoder skill relative to PCA-DP and to PCA-LIM, respectively, for a few different lead times. Notably, the Koopman Autoencoder is better at reconstructing SSH than PCA at low latitudes but is worse at midlatitudes (Figure 4c). However, by lag $\tau = 5$ days, the negative skill in the midlatitudes has diminished compared to PCA-LIM (Figure 4g), and there is positive skill relative to PCA-DP over the entire domain (Figure 4d). Because the midlatitude SSH variability is dominated by the high-frequency inverse-barometer component (Supporting Figure S3), midlatitude SSH dynamics are inherently less pre-

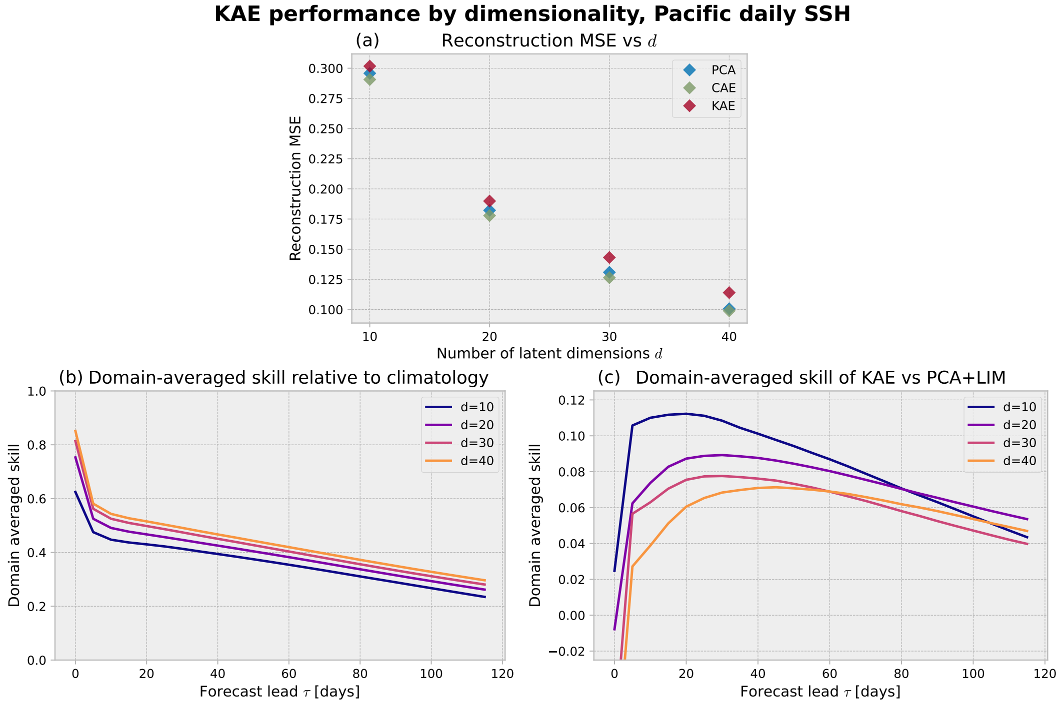

Figure 3. Sensitivity of the Koopman Autoencoder to number of dimensions for predicting North Pacific daily-averaged SSH. (a) Reconstruction error by dimensionality for PCA (blue), CAE (light green), and the Koopman Autoencoder (red). (b) Domain-averaged MSE skill scores of the Koopman Autoencoder predictions relative to climatology for different latent space dimensionalities. (c) Domain-averaged skill score of the Koopman Autoencoder relative to equivalent dimensionality PCA-LIM as a function of forecast lead.

dictable than low-latitude dynamics. Therefore, for North Pacific regional-scale predictions, quality representations of SSH in the tropics are much more helpful for regional-scale predictions than representations in the midlatitudes. Because the dimensionality reduction and propagation are learned together in the Koopman Autoencoder, it can deploy its latent dimensions to focus on representing low-latitude SSH initial conditions particularly well. In contrast, when the dimensionality reduction is done separately, dimensions may be wasted on characterizing variability that is not predictable.

The skill maps also highlight dynamics that the PCA-based propagators do not fully capture. For instance, since PCA-DP characterizes the *local* predictability of SSH, skill of the Koopman Autoencoder relative to PCA-DP indicates that it is capturing *nonlocal* drivers of SSH. Midlatitude skill in the Northeastern Pacific at leads of $\tau = 5$ days (Figure 4d) could come from the advection of sea level pressure anomalies via midlatitude Westerlies, which traverse the Pacific basin on $\mathcal{O}(5–10 \text{ days})$. In the low latitudes, the skill of the Koopman Autoencoder with respect to PCA-DP and PCA-LIM increases until about 30 days (Figures 4a), with the strongest skill occurring in narrow, zonal bands adjacent to the equator (Figure 4h). This timescale and region of enhanced skill is consistent with the timescale and westward propagation of Equatorial Rossby waves.

In the North Atlantic, we note that reconstruction errors for the Koopman Autoencoder at time $\tau = 0$ are poor, with a domain-average skill of -0.14 relative to the PCA reconstructions. However, once again, the latent space representation of the state results in better skill at nonzero time lags up to $\tau = 100$ days (Figure 4b). Figures 4i-k show that the prediction skill of the Koopman Autoencoder occurs primarily in the Atlantic Subtropical Gyre and Gulf Stream separation. Because gyre dynamics are associated primarily with low-variability geostrophic balance, such variability may be underrepresented in variance-targeting PCA-based reconstructions, even though this variability may be predictable on the daily-to-seasonal timescale. Reconstruction skill relative to PCA suggests that the Caribbean Current may be a source of this gyre predictability for SSH predictions in the North Atlantic (Figure 4i).

4 Discussion

Statistical-dynamical models—and linear inverse models in particular—have become indispensable forecasting tools in the past few decades, owing to their simplicity, interpretability, and skill (Penland & Sardeshmukh, 1995; Alexander et al., 2008; von Storch et al., 1995). Modern techniques can help extract more information from data for nonlinear systems. In this study, we trained convolutional neural networks with embedded time-stepping to learn a low-dimensional latent space that facilitates predictions of SSH. Training the network to learn the dimensionality reduction and propagation simultaneously tends to result in better forecasts than if the reduction and propagation are learned separately, as done typically with LIM for example.

We examined some sensitivities of the Koopman Autoencoder method compared to LIM. The skillfulness of the Koopman Autoencoder is most apparent in situations when the assumptions for LIM are least valid (such as on daily data, where the state vector includes highly nonlinear, small-scale features). Additionally, we examined the sensitivity to the dimensionality of the latent space. Our results suggest that the Koopman Autoencoder framework is best for building low-dimensional propagators; however, computational considerations led us to consider only one region and timescale and up to 40 latent dimensions, so the robustness of this result to different dynamics and a wider range of dimensionalities should be further investigated.

Spatial variations in the reconstruction skill of the Koopman Autoencoder point to sources of predictability that the Koopman Autoencoder leverages to make better predictions than LIM. We identified tropical Pacific SSH as a source of predictability for North Pacific daily-averaged SSH and the Caribbean Current SSH for North Atlantic SSH. One limitation of this study is that a univariate field variable is used for SSH predictions. Previous studies have demonstrated that including multiple variables can improve LIM predictions (Newman, Alexander, & Scott, 2011; Capotondi et al., 2022; Brennan et al., 2023). Using multiple input channels to incorporate different fields may improve the Koopman Autoencoder's SSH predictions and reveal additional sources of predictability.

The focus of this study has been to develop an efficient propagator for SSH and to assess its forecasting skill. The imposed linearity of the dynamics in the latent space could be relaxed (for instance, to obtain better predictions). However, the comprehensive theory underpinning linear systems makes the linear propagator potentially appealing for interpretation, yielding possible advantages in applications like predictability (Vimont et al., 2014; Tziperman et al., 2008), emulation (Beucler et al., 2021; Bi et al., 2023), and inference (Baldovin et al., 2020; Falasca et al., 2024).

One question is how the latent state can be physically interpreted (Shamekh et al., 2023; Behrens et al., 2022). In the context of Koopman operator theory, the latent space variables are nonlinear observables of the dynamical system state, but the nonlinearities in the encoder and decoder make it challenging to interpret what these observables measure. One approach to gaining physical understanding of the latent space could be to probe the sensitivity of the decoder to changes in the latent space, either through observing the sensitivity of the outputs to perturbations to the latent space variables (Oring

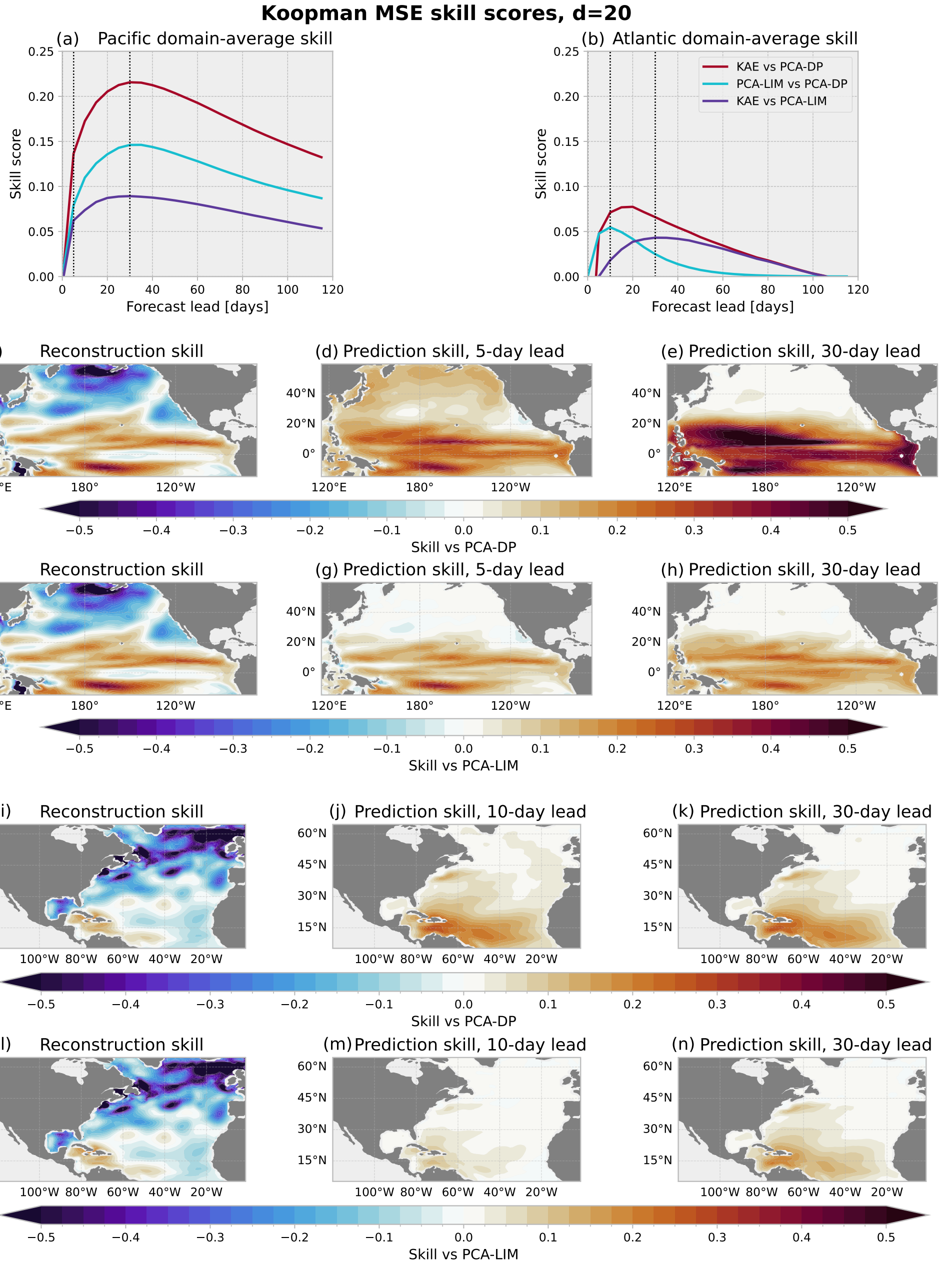

Figure 4. Koopman Autoencoder MSE skill scores for daily-averaged North Pacific (a, c–h) and North Atlantic (b, i–n) SSH predictions. (a, b): Domain-averaged skill as a function of lead time. Red: Skill of Koopman Autoencoder relative to PCA-DP. Purple: Koopman Autoencoder relative to PCA+LIM. Cyan: Skill of PCA-LIM relative to PCA-DP. Black dotted lines indicate forecast leads used for panels c–h. (c, d, e, i, j, k): Skill scores of Koopman Autoencoder relative to PCA-DP at select time lags. (f, g, h, l, m, n): Same but for skill relative to PCA-LIM.

et al., 2021; Leeb et al., 2022) or examining the gradients of the decoder (Mamalakis et al., 2022; Baehrens et al., 2010). Such methods for interpreting the latent space, coupled with eigenanalysis for understanding the timescales for the propagator, could help elucidate the physical processes represented in the latent space, and is left for future work.

Nevertheless, we believe this study has demonstrated a potentially useful approach for developing efficient, low-dimensional linear propagators for climate fields.

Appendix A Open Research

The CESM2 Large Ensemble Dataset is available from the NCAR Climate Data Gateway at `https://doi.org/10.26024/kgmp-c556` (Danabasoglu et al., 2021). The code used for data processing, training, analysis and visualization in this study, as well as the files for reproducing the software environment, are provided under the MIT license at `https://github.com/andrewbrettin/koopman_autoencoders_ssh_prediction` (Brettin, 2024). Figure 1 was built using the PlotNeuralNet software preserved at `https://doi.org/10.5281/zenodo.2526396`, which is available via the MIT license (HarisIqbal88, 2018).

Acknowledgments

This work is supported by the VoLo Foundation. LZ was supported by NOAA grant NOAA-OAR-CPO-2019-2005530,1148, by the KITP Program "Machine Learning and the Physics of Climate" under the National Science Foundation Grant No. NSF PHY-1748958 and by Schmidt Sciences, LLC. EAB is funded, in part, by NOAA grants NA24OARX431C0022 and NA22OAR4310621. We acknowledge high-performance computing support provided by the NSF National Center for Atmospheric Research (NCAR) Computational and Information Systems Laboratory (CISL, 2023). We thank Aurora Basinski-Ferris, Adam Subel, and Chris Pedersen for numerous helpful discussions on this work.

References

Albers, J. R., & Newman, M. (2021). Subseasonal predictability of the North Atlantic Oscillation. *Environmental Research Letters*, *16*(4), 044024.

Alexander, M. A., Matrosova, L., Penland, C., Scott, J. D., & Chang, P. (2008). Forecasting Pacific SSTs: Linear inverse model predictions of the PDO. *Journal of Climate*, *21*(2), 385–402.

Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., & Müller, K.-R. (2010). How to Explain Individual Classification Decisions. *Journal of Machine Learning Research*, *11*, 1803–1831.

Baldovin, M., Cecconi, F., & Vulpiani, A. (2020). Understanding causation via correlations and linear response theory. *Physical Review Research*, *2*(4), 043436.

Behrens, G., Beucler, T., Gentine, P., Iglesias-Suarez, F., Pritchard, M., & Eyring, V. (2022). Non-Linear Dimensionality Reduction With a Variational Encoder Decoder to Understand Convective Processes in Climate Models. *Journal of Advances in Modeling Earth Systems*, *14*(8), e2022MS003130.

Beucler, T., Pritchard, M., Rasp, S., Ott, J., Baldi, P., & Gentine, P. (2021). Enforcing Analytic Constraints in Neural Networks Emulating Physical Systems. *Physical Review Letters*, *126*(9), 098302.

Bi, K., Xie, L., Zhang, H., Chen, X., Gu, X., & Tian, Q. (2023). Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, *619*(7970), 533–538.

Brennan, M. K., Hakim, G. J., & Blanchard-Wrigglesworth, E. (2023). Monthly Arctic Sea-Ice Prediction With a Linear Inverse Model. *Geophysical Research Letters*, *50*(7), e2022GL101656.

Brettin, A. (2024). *Code for "Learning Propagators for Sea Surface Height Forecasts Using Koopman Autoencoders"* [Software]. Zenodo. Retrieved from `https://github.com/andrewbrettin/koopman_autoencoders_ssh_prediction`

Brunton, S. L., & Kutz, J. N. (2022). *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control.* Cambridge University

Press.
Cabanes, C., Huck, T., & Colin de Verdière, A. (2006). Contributions of Wind Forcing and Surface Heating to Interannual Sea Level Variations in the Atlantic Ocean. *Journal of Physical Oceanography*, *36*(9), 1739–1750.
Calafat, F. M., Wahl, T., Lindsten, F., Williams, J., & Frajka-Williams, E. (2018). Coherent modulation of the sea-level annual cycle in the United States by Atlantic Rossby waves. *Nature communications*, *9*(1), 2571.
Capotondi, A., Newman, M., Xu, T., & Di Lorenzo, E. (2022). An Optimal Precursor of Northeast Pacific Marine Heatwaves and Central Pacific El Niño Events. *Geophysical Research Letters*, *49*(5), e2021GL097350.
Champion, K., Lusch, B., Kutz, J. N., & Brunton, S. L. (2019). Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, *116*(45), 22445–22451.
Chelton, D. B., & Schlax, M. G. (1996). Global Observations of Oceanic Rossby Waves. *Science*, *272*(5259), 234–238.
CISL. (2023). *Derecho: HPE Cray EX System (University Community Computing).* Boulder, CO: NSF National Center for Atmospheric Research. doi: https://doi.org/10.5065/qx9a-pg09
Danabasoglu, G., Deser, C., Rodgers, K., & Timmermann, A. (2021). *CESM2 Large Ensemble Dataset* [dataset]. National Center for Atmospheric Research. Retrieved from `https://www.earthsystemgrid.org/dataset/ucar.cgd.cesm2le.output.html` doi: https://doi.org/10.26024/kgmp-c556
Danabasoglu, G., Lamarque, J.-F., Bacmeister, J., Bailey, D., DuVivier, A., Edwards, J., . . . others (2020). The Community Earth System Model Version 2 (CESM2). *Journal of Advances in Modeling Earth Systems*, *12*(2), e2019MS001916.
DelSole, T. (2000). A Fundamental Limitation of Markov Models. *Journal of the Atmospheric Sciences*, *57*(13), 2158–2168.
Dommenget, D., & Latif, M. (2002). A Cautionary Note on the Interpretation of EOFs. *Journal of Climate*, *15*(2), 216–225.
Falasca, F., Perezhogin, P., & Zanna, L. (2024). Data-driven dimensionality reduction and causal inference for spatiotemporal climate fields. *Physical Review E*, *109*(4), 044202.
Fofonoff, N. P., & Millard Jr, R. (1983). Algorithms for the Computation of Fundamental Properties of Seawater. *UNESCO Technical Papers in Marine Sciences*, *44*.
Fraser, R., Palmer, M., Roberts, C., Wilson, C., Copsey, D., & Zanna, L. (2019). Investigating the predictability of North Atlantic sea surface height. *Climate Dynamics*, *53*, 2175–2195.
Fukushima, K. (1980). Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*, *36*(4), 193–202.
Gill, A., & Niller, P. (1973). The theory of the seasonal variability in the ocean. *Deep Sea Research and Oceanographic Abstracts*, *20*(2), 141–177.
Gregory, J. M., Griffies, S. M., Hughes, C. W., Lowe, J. A., Church, J. A., Fukimori, I., . . . others (2019). Concepts and terminology for sea level: Mean, variability and change, both local and global. *Surveys in Geophysics*, *40*, 1251–1289.
HarisIqbal88. (2018). *PlotNeuralNet* [Software]. Zenodo. doi: https://doi.org/10.5281/zenodo.2526396
Hasselmann, K. (1976). Stochastic climate models: Part I. Theory. *Tellus*, *28*(6), 473–485.
Hermans, T. H., Katsman, C. A., Camargo, C. M., Garner, G. G., Kopp, R. E., & Slangen, A. B. (2022). The Effect of Wind Stress on Seasonal Sea-Level Change on the Northwestern European Shelf. *Journal of Climate*, *35*(6), 1745–1759.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, *313*(5786), 504–507.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417.

Kamp, W., Han, W., Zhang, L., Kido, S., & McCreary, J. P. (2024). Tropical Atmospheric Intraseasonal Oscillations Leading to Sea Level Extremes in Coastal Indonesia during Recent Decades. *Journal of Climate*, *37*(9), 2867–2880.

Koopman, B. O. (1931). Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, *17*(5), 315–318.

Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, *37*(2), 233–243.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, *1*(4), 541–551.

LeCun, Y., Bottou, L., Orr, G. B., & Müller, K.-R. (2002). Efficient BackProp. In G. M. et al. (Ed.), *Neural Networks: Tricks of the Trade* (pp. 9–48). Springer.

Leeb, F., Bauer, S., Besserve, M., & Schölkopf, B. (2022). Exploring the Latent Space of Autoencoders with Interventional Assays. *Advances in Neural Information Processing Systems*, *35*, 21562–21574.

Legates, D. R., & Davis, R. E. (1997). The continuing search for an anthropogenic climate change signal: Limitations of correlation-based approaches. *Geophysical Research Letters*, *24*(18), 2319–2322.

Lorenz, E. N. (1956). *Empirical Orthogonal Functions and Statistical Weather Prediction* (Vol. 1). Massachusetts Institute of Technology, Department of Meteorology Cambridge.

Lorenz, E. N. (1973). On the Existence of Extended Range Predictability. *Journal of Applied Meteorology*, 543–546.

Lusch, B., Kutz, J. N., & Brunton, S. L. (2018). Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, *9*(1), 4950.

Mamalakis, A., Ebert-Uphoff, I., & Barnes, E. A. (2022). Neural network attribution methods for problems in geoscience: A novel synthetic benchmark dataset. *Environmental Data Science*, *1*, e8.

Mardt, A., Pasquali, L., Wu, H., & Noé, F. (2018). VAMPnets for deep learning of molecular kinetics. *Nature communications*, *9*(1), 5.

Marques, G. M., Loose, N., Yankovsky, E., Steinberg, J. M., Chang, C.-Y., Bhamidipati, N., … others (2022). NeverWorld2: An idealized model hierarchy to investigate ocean mesoscale eddies across resolutions. *Geoscientific Model Development*, *15*(17), 6567–6579.

Murphy, A. H. (1988). Skill Scores Based on the Mean Square Error and their Relationships to the Correlation Coefficient. *Monthly Weather Review*, *116*(12), 2417–2424.

Newman, M., Alexander, M. A., & Scott, J. D. (2011). An empirical model of tropical ocean dynamics. *Climate Dynamics*, *37*, 1823–1841.

Newman, M., Shin, S.-I., & Alexander, M. A. (2011). Natural Variation in ENSO Flavors. *Geophysical Research Letters*, *38*(14).

O'Neill, B. C., Tebaldi, C., van Vuuren, D. P., Eyring, V., Friedlingstein, P., Hurtt, G., … Sanderson, B. M. (2016). The Scenario Model Intercomparison Project (ScenarioMIP) for CMIP6. *Geoscientific Model Development*, *9*(9), 3461–3482. Retrieved from `https://gmd.copernicus.org/articles/9/3461/2016/` doi: 10.5194/gmd-9-3461-2016

Oommen, V., Shukla, K., Goswami, S., Dingreville, R., & Karniadakis, G. E. (2022). Learning two-phase microstructure evolution using neural operators and autoencoder architectures. *npj Computational Materials*, *8*(1), 190.

Oring, A., Yakhini, Z., & Hel-Or, Y. (2021, 18–24 Jul). Autoencoder Image Interpolation by Shaping the Latent Space. In *Proceedings of the 38th International*

Conference on Machine Learning (Vol. 139, pp. 8281–8290).
Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572.
Penland, C. (1989). Random Forcing and Forecasting Using Principal Oscillation Pattern Analysis. *Monthly Weather Review*, *117*(10), 2165–2185.
Penland, C. (2019). The Nyquist Issue in Linear Inverse Modeling. *Monthly Weather Review*, *147*(4), 1341–1349.
Penland, C., & Ghil, M. (1993). Forecasting Northern Hemisphere 700-mb Geopotential Height Anomalies Using Empirical Normal Modes. *Monthly Weather Review*, *121*(8), 2355–2372.
Penland, C., & Sardeshmukh, P. D. (1995). The optimal growth of tropical sea surface temperature anomalies. *Journal of Climate*, *8*(8), 1999–2024.
Piecuch, C., Dangendorf, S., Ponte, R. M., & Marcos, M. (2016). Annual sea level changes on the north american northeast coast: Influence of local winds and barotropic motions. *Journal of Climate*, *29*(13), 4801–4816.
Piecuch, C., & Ponte, R. (2011). Mechanisms of Interannual Steric Sea Level Variability. *Geophysical Research Letters*, *38*(15).
Ponte, R. M. (2006). Low-frequency sea level variability and the inverted barometer effect. *Journal of Atmospheric and Oceanic Technology*, *23*(4), 619–629.
Richter, I., Chang, P., & Liu, X. (2020). Impact of Systematic GCM Errors on Prediction Skill as Estimated by Linear Inverse Modeling. *Journal of Climate*, *33*(23), 10073–10095.
Roberts, C., Calvert, D., Dunstone, N., Hermanson, L., Palmer, M., & Smith, D. (2016). On the drivers and predictability of seasonal-to-interannual variations in regional sea level. *Journal of Climate*, *29*(21), 7565–7585.
Rodgers, K., Lee, S.-S., Rosenbloom, N., Timmermann, A., Danabasoglu, G., Deser, C., ... others (2021). Ubiquity of human-induced changes in climate variability. *Earth System Dynamics*, *12*(4), 1393–1411.
Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.
Sardeshmukh, P. D., & Sura, P. (2009). Reconciling non-gaussian climate statistics with linear dynamics. *Journal of Climate*, *22*(5), 1193–1207.
Schmid, P. J. (2010). Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, *656*, 5–28.
Shamekh, S., Lamb, K. D., Huang, Y., & Gentine, P. (2023). Implicit learning of convective organization explains precipitation stochasticity. *Proceedings of the National Academy of Sciences*, *120*(20), e2216158120.
Shin, S.-I., & Newman, M. (2021). Seasonal predictability of global and north american coastal sea surface temperature and height anomalies. *Geophysical Research Letters*, *48*(10), e2020GL091886.
Smith, R., Jones, P., Briegleb, B., Bryan, F., Danabasoglu, G., Dennis, J., ... others (2010). The Parallel Ocean Program (POP) Reference Manual. *LAUR-01853*, *141*, 1–140.
Stephenson, D., Hannachi, A., & O'Neill, A. (2004). On the existence of multiple climate regimes. *Quarterly Journal of the Royal Meteorological Society*, *130*(597), 583–605.
Tziperman, E., Zanna, L., & Penland, C. (2008). Nonnormal thermohaline circulation dynamics in a coupled ocean–atmosphere gcm. *Journal of Physical Oceanography*, *38*(3), 588–604.
Vimont, D. J., Alexander, M. A., & Newman, M. (2014). Optimal growth of Central and East Pacific ENSO events. *Geophysical Research Letters*, *41*(11), 4027–4034.
Vinogradova, N. T., Ponte, R. M., & Stammer, D. (2007). Relation between sea level and bottom pressure and the vertical dependence of oceanic variability. *Geo-*

physical Research Letters, *34*(3).
von Storch, H., Bürger, G., Schnur, R., & von Storch, J.-S. (1995). Principal Oscillation Patterns: A Review. *Journal of Climate*, 377–400.
Webb, D. J. (2021). On the low western Pacific sea levels observed prior to strong East Pacific El Niños. *Ocean Science*, *17*(6), 1585–1604.
Yeung, E., Kundu, S., & Hodas, N. (2019). Learning Deep Neural Network Representations for Koopman Operators of Nonlinear Dynamical Systems. In *2019 American Control Conference (ACC)* (pp. 4832–4839).
Zanna, L. (2012). Forecast Skill and Predictability of Observed Atlantic Sea Surface Temperatures. *Journal of Climate*, *25*(14), 5047–5056.

Supporting Information for "Learning Propagators for Sea Surface Height Forecasts Using Koopman Autoencoders"

Andrew E. Brettin[1], Laure Zanna[1], Elizabeth A. Barnes[2]

[1]Courant Institute of Mathematical Sciences, New York University, New York, NY, USA

[2]Department of Atmospheric Science, Colorado State University, Fort Collins, CO, USA

Contents of this file

Introduction

Here, we describe methodological training details and analysis metrics used in this study (Text S1, Text S2), provide supplementary figures describing the validity of the propagators (Figure S1 and S2), show the SSH variability due to different components to give context for the performance differences between regions (Figure S3), and provide tables to quantify the reconstruction and prediction performance of the different dimensionality reduction and propagation techniques (Tables S1, S2, and S3).

Text S1. Architecture and training configurations for the convolutional neural networks

The encoder and decoder of our Convolutional Autoencoder and Koopman Autoencoder are composed of convolutional "blocks," where each block consists of a convolutional layer equipped with ReLU activations followed by another convolutional layer with ReLU activations (Fukushima, 1969, 1980). The convolutional layers use a 3-by-3 filter with a stride of 1 and employ zero-padding to preserve the shape of the input fields. In the encoder, convolutional blocks are succeeded by max-pooling operations using a 2-by-2 kernel, whereas in the decoder, convolutional blocks are preceded by bilinear upsampling using a 2-by-2 kernel. We use an architecture somewhat similar to Oommen, Shukla, Goswami, Dingreville, and Karniadakis (2022), where the number of filters per block is decreased closer to the bottleneck. In the encoder, the first two convolutional blocks contain convolutional layers with 64 channels, the third block contains layers of 32 channels, and the fourth contains layers of 16 channels. The last convolutional layer is fully connected to the latent space encoding. The decoder essentially has the reverse structure of the encoder: the encoding is fully connected to a convolutional block employing layers of 16 channels, followed by a block with layers of 32 channels, and then two blocks of 64 channels. Additionally, the decoder applies a 1-by-1 convolution to the outputs of the last convolutional block in order to return values in the range $(-\infty, \infty)$.

We optimize the parameters of the networks using the Adam optimizer (Kingma & Ba, 2014) with batches of 64 samples and a fixed learning rate of 10^{-4}. For the Koopman autoencoders, L_2 regularization is applied over all network weights to mitigate overfitting. For the daily-averaged data, an L_2 weight of 10^{-3} is applied, whereas for the monthly-averaged data, a higher regularization rate of 10^{-2} was necessary to prevent overfitting. Networks are trained for 500 epochs, with an early stopping threshold of 50 epochs. Checkpoints for the network with the best overall validation loss were saved. Additionally, for the Koopman Autoencoder, we save the checkpoints with the best validation-set prediction MSE such that the learned propagator has decaying eigenvalues. This checkpoint with the best prediction loss is used.

The training capacity of both the Convolutional Autoencoder and Koopman Autoencoder was found to be sensitive to the network weight initializations: for certain initial weights, the network only converged to a constant function. Therefore, for the Convolutional Autoencoder, we initialize weights using Kaiming uniform random values (He et al., 2015), and reinitialize the weights with a different set of Kaiming uniform random values if the network does not converge to a lower loss than that of a constant function. For the Koopman Autoencoder, we leverage information gained about the loss landscape during the training process for the Convolutional Autoencoder. The Koopman Autoencoder's encoder and decoder weights are initialized from the weights of the Convolutional Autoencoder at the 10^{th} epoch of training. This is based on the principle that lower-order features are learned first during training (Kalimeris et al., 2019; Refinetti et al., 2023): by beginning the training from the 10^{th} epoch, the encoder and decoder contain enough complexity to converge to something more expressive than a constant function, but not so much complexity that the KAE overfits. Furthermore, the weights for the linear propagator L are initialized as a multiple of the identity matrix $\alpha\mathbf{I}$, where $\alpha \in (0, 1)$. Thus, the propagator is learned by making gradual adjustments to a type of damped-persistence forecast. We set $\alpha = 0.5$ for these experiments.

The data consists of 32,060 training samples for the daily data, and 21,014 samples for the monthly data (daily data is subsampled by a factor of 20 to reduce the computational cost). We use $k = 20$ recurrent passes for the prediction loss, and set the relative weights of the three different loss functions $\lambda_1 = \lambda_2 = \lambda_3 = 1$. The networks are trained in Pytorch using the distributed data parallel approach on two NVIDIA 32GB V100 GPUs (Paszke et al., 2019; Li et al., 2020).

Text S2. Metrics

Here we define metrics used for assessing reconstruction and prediction performance.

Let $\mathbf{X}$ be the tensor of target values for a specific geophysical field, and let $\hat{\mathbf{X}}$ be the predicted values. These tensors have entries $x_{i,j,n}$, and $\hat{x}_{i,j,n}$, where $i \in \{1, \ldots, M_x\}$ indexes the longitudes, $j \in \{1, \ldots, M_y\}$ indexes the latitudes, and $n \in \{1 \ldots, N\}$ indexes the samples.

We first define domain averaged metrics for a specific sample. Using a wildcard "$*$" to indicate dimensions of aggregation, the area-weighted Mean Squared Error (MSE) for a specific sample is given by

$$\mathrm{MSE}_{(*,*,n)} = \frac{\sum_{i=1}^{M_x} \sum_{j=1}^{M_y} w_{i,j}^2 (x_{i,j,n} - \hat{x}_{i,j,n})^2}{\sum_{i=1}^{M_x} \sum_{j=1}^{M_y} w_{i,j}^2} \tag{1}$$

where $w_{i,j}$ gives the $(i,j)^{\text{th}}$ weight, which is proportional to grid-cell area on nondegenerate points and 0 on masked points. Similarly, the area-weighted pattern Correlation

Coefficient (CC) for a given sample is given by

$$\mathrm{CC}_{(*,*,n)} = \frac{\sum_{i=1}^{M_x} \sum_{j=1}^{M_y} w_{i,j}^2 \ x_{i,j,n} \ \hat{x}_{i,j,n}}{\sqrt{\sum_{i=1}^{M_x} \sum_{j=1}^{M_y} (w_{i,j} \ x_{i,j,n})^2 \sum_{i=1}^{M_x} \sum_{j=1}^{M_y} (w_{i,j} \ \hat{x}_{i,j,n})^2}} \tag{2}$$

Global metrics over all gridpoints and samples can be obtained by averaging over all samples:

$$\mathrm{MSE} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{MSE}_{(*,*,n)} \tag{3}$$

$$\mathrm{CC} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{CC}_{(*,*,n)} \tag{4}$$

The area-weighted ℓ^2-norms $\| \cdot \|_{2,w}$ given in Eqs. (3) and (4) use the globally-averaged area-weighted MSE in Eq. (3).

We can also consider the sample averaged MSE at each location, given by

$$\mathrm{MSE}_{(i,j,*)} = \frac{1}{N} \sum_{n=1}^{N} (x_{i,j,n} - \hat{x}_{i,j,n})^2 \tag{5}$$

It is often useful to assess the predictions of a model relative to another baseline. The skill score is an often used metric that assigns a value between 0 and 1 to assess the performance of the model relative to a baseline (Murphy, 1988). For a prediction model f and a baseline f_0, we define the total skill score by

$$\mathrm{SS} = 1 - \frac{\mathrm{MSE}(f)}{\mathrm{MSE}(f_0)}. \tag{6}$$

where $\mathrm{MSE}(f)$ gives the error given by the model f. This can be interpreted as the percentage of improvement in MSE gained by using model f instead of f_0.

Likewise, the sample-averaged skill score for each location by

$$\mathrm{SS}_{i,j} = 1 - \frac{\mathrm{MSE}_{(i,j,*)}(f)}{\mathrm{MSE}_{(i,j,*)}(f_0)} \tag{7}$$

where $\mathrm{MSE}_{(i,j,*)}(f)$ is the sample-averaged MSE using prediction model f. Finally, domain-averaged skill is found by area-weighted averaging over all spatial indices (i, j):

$$\overline{\mathrm{SS}} = \frac{\sum_{i=1}^{M_x} \sum_{j=1}^{M_y} w_{i,j} \mathrm{SS}_{i,j}}{\sum_{i=1}^{M_x} \sum_{j=1}^{M_y} w_{i,j}} \tag{8}$$

References

Fukushima, K. (1969). Visual Feature Extraction by a Multilayered Network of Analog Threshold Elements. *IEEE Transactions on Systems Science and Cybernetics*, *5*(4), 322–333.

Fukushima, K. (1980). Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. *Biological Cybernetics*, *36*(4), 193–202.

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV).*

Kalimeris, D., Kaplun, G., Nakkiran, P., Edelman, B., Yang, T., Barak, B., & Zhang, H. (2019). SGD on Neural Networks Learns Functions of Increasing Complexity. In *Advances in Neural Information Processing Systems* (Vol. 32). NeurIPS.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. doi: 10.48550/arXiv.1412.6980

Li, S., Zhao, Y., Varma, R., Salpekar, O., Noordhuis, P., Li, T., ... Chintala, S. (2020, aug). PyTorch distributed: experiences on accelerating data parallel training. *Proceedings of the VLDB Endowment*, *13*(12), 3005–3018. Retrieved from https://doi.org/10.14778/3415478.3415530 doi: 10.14778/3415478.3415530

Murphy, A. H. (1988). Skill Scores Based on the Mean Square Error and their Relationships to the Correlation Coefficient. *Monthly Weather Review*, *116*(12), 2417–2424.

Oommen, V., Shukla, K., Goswami, S., Dingreville, R., & Karniadakis, G. E. (2022). Learning two-phase microstructure evolution using neural operators and autoencoder architectures. *npj Computational Materials*, *8*(1), 190.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... others (2019). Pytorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in Neural Information Processing Systems*, *32*.

Penland, C., & Sardeshmukh, P. D. (1995). The optimal growth of tropical sea surface temperature anomalies. *Journal of Climate*, *8*(8), 1999–2024.

Refinetti, M., Ingrosso, A., & Goldt, S. (2023). Neural networks trained with SGD learn distributions of increasing complexity. In *International Conference on Machine Learning* (pp. 28843–28863).

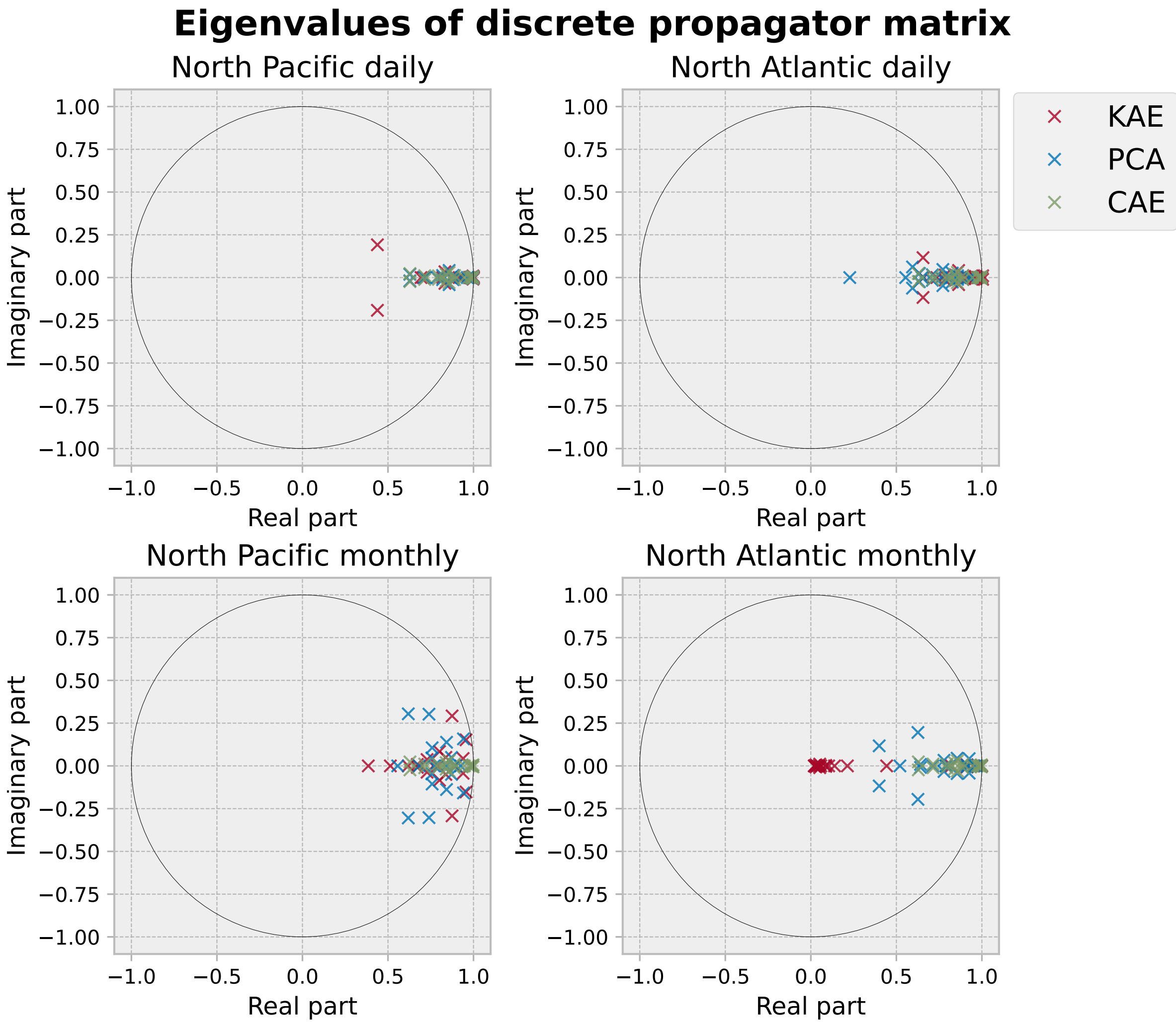

Figure S1. Eigenvalues of discrete propagators of LIM, $\mathbf{B}(1)$, for both PCA and CAE latent modes, as well as the eigenvalues of the Koopman Autoencoder propagator L. The unit circle demarcates the region in which the eigenvalues must lie for the propagator to be stable.

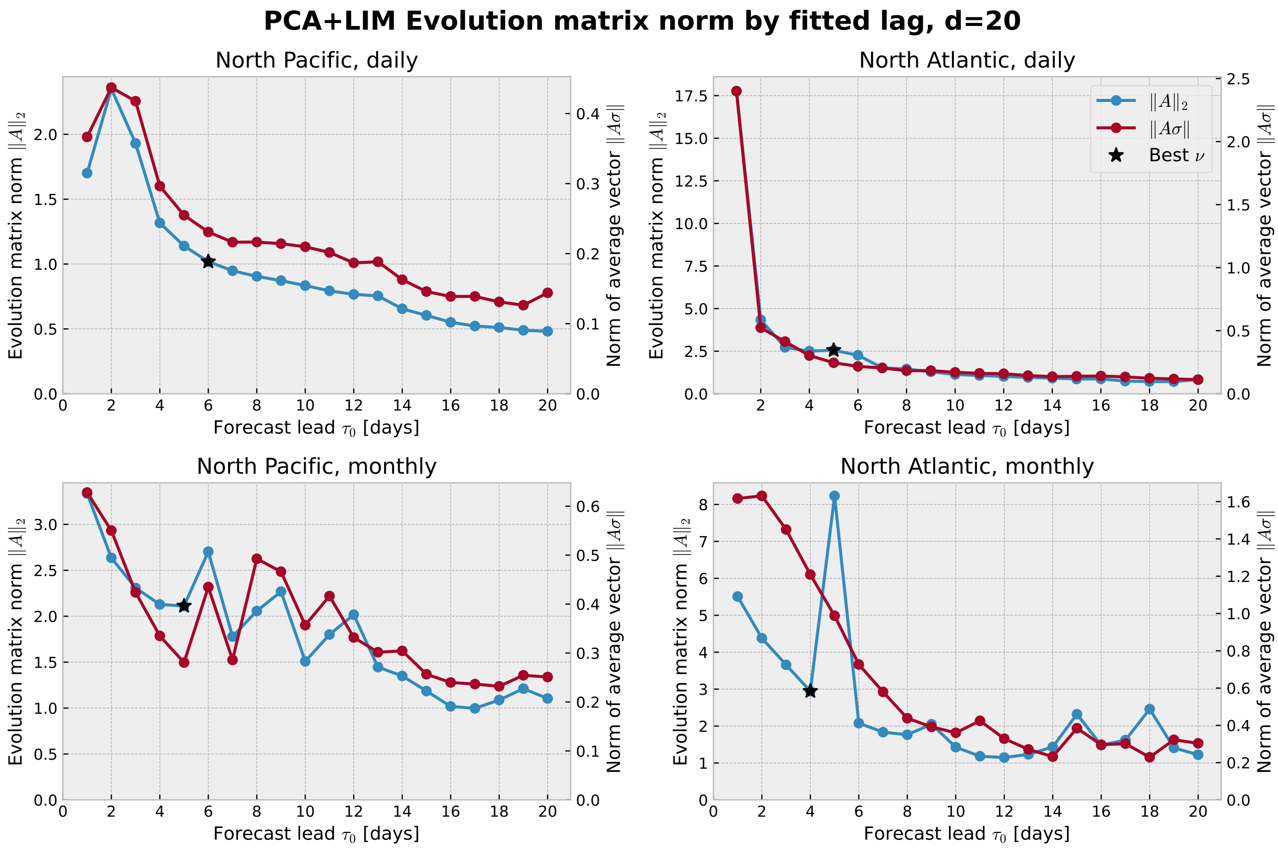

Figure S2. PCA+LIM evolution matrix norms by fitted propagator lead time. The blue line shows the matrix norm itself, with a star indicating the model with the lowest average prediction MSE over timesteps 1–k on the validation dataset. The red line shows the norm of an average propagated latent space vector σ, as in Penland and Sardeshmukh (1995) Fig. 12.

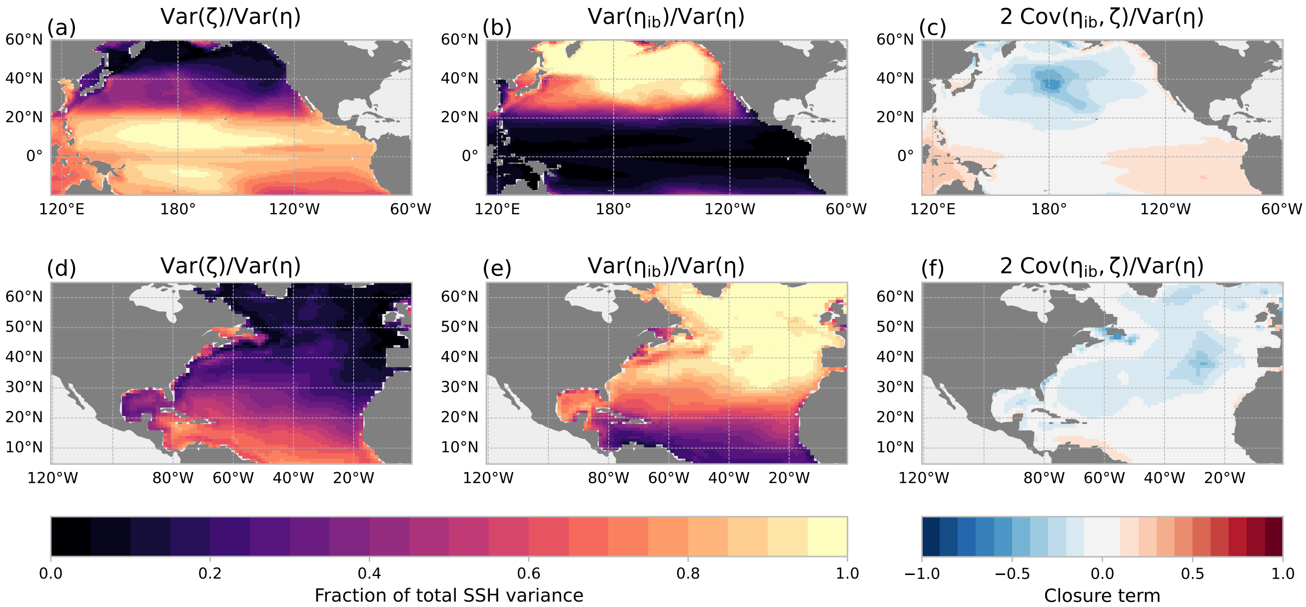

Figure S3. Explained variance of daily SSH variability by component in the North Pacific (a, b, c) and North Atlantic (d, e, f). Panels (a) and (d) show the proportion of SSH variability due to dynamic sea level, while panels (b) and (e) show the proportion due to the inverse barometer component. Because the random variates ζ and η_{ib} are not completely decorrelated, the explained variance by the two terms do not exactly sum to 1. Therefore, the closure term due to covariance $2\text{Cov}(\zeta, \eta_{\text{ib}})/\text{Var}(\eta)$, which is negligible at most locations, is included in panels (c) and (f).

Table S1. Reconstruction MSE for different dimensionality reduction techniques with $d = 20$ latent dimensions. Parentheses show the percent difference in MSE from PCA.

model	Pacific Daily	Atlantic Daily	Pacific Monthly	Atlantic Monthly
PCA	0.191	0.065	0.167	0.082
CAE	0.185 (-3.61%)	0.063 (-2.00%)	0.161 (-3.83%)	0.086 (+5.26%)
KAE	0.198 (+3.27%)	0.078 (+20.69%)	0.231 (+38.39%)	0.135 (+64.10%)

Table S2. Total skill score (expressed as a percentage) of different prediction methods relative to PCA-DP, averaged over forecast leads up to 120 days for daily data and 36 months for monthly data.

Prediction method	Pacific Daily	Atlantic Daily	Pacific Monthly	Atlantic Monthly
CAE-DP	-15.4%	-0.4%	-1.0%	-0.9%
PCA-LIM	9.4%	0.6%	13.9%	3.3%
CAE-LIM	9.6%	0.2%	12.8%	3.0%
KAE	13.9%	1.7%	16.0%	2.0%

Table S3. Reconstruction MSE for different dimensionality reduction techniques in the North Pacific on daily timescales for different numbers of latent dimensions. Lighter shading represents lower MSE. Parentheses show the percent difference in MSE from PCA.

Technique	D=10	D=20	D=30	D=40
PCA	0.308	0.191	0.137	0.106
CAE	0.301 (-2.35%)	0.185 (-3.61%)	0.131 (-4.01%)	0.103 (-2.11%)
KAE	0.311 (+0.92%)	0.198 (+3.27%)	0.149 (+8.80%)	0.119 (+12.72%)