LEARN FROM THE CORE, FIT TO THE DATA: DEEP KOOPMAN LEARNING OF THE ATMOSPHERE

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep Koopman learning typically consists of dictionary learning followed by operator fitting, but we find that this pipeline can severely overfit, especially when the rank of the fitted operator is overspecified, because the same stochastic transitions select the features and fit the operator. We study two ways to mitigate this issue: separating the data used for dictionary learning and operator fitting, and using the forecasts of an approximate prediction model, which we call a core model, to train the dictionary. Our main finding is that core-trained dictionaries can substantially improve the held-out VAMP-E score over data-trained ones. On ERA5 and a controlled stochastic shallow-water benchmark, the advantage becomes more pronounced at long lags, and in the benchmark also under stronger transition noise.

1 INTRODUCTION

Forecasting a stochastic dynamical system means characterizing a distribution of its future states, as in probabilistic weather forecasting (Price et al., 2025; Alet et al., 2025). At a fixed lag, some observables of the future are more predictable from the present than others. The Koopman transfer operator maps a future-state observable to its conditional expectation given the present state (Mezić, 2005; Williams et al., 2015). Its leading right singular functions (Wu & Noé, 2020; Mardt et al., 2018) identify the most predictable future observables, with their squared singular values quantifying predictability (Appendix C.1). A few leading singular functions can therefore provide a low-dimensional description of the most predictable structure at a given lag. Estimating that description, rather than forecasting the state itself, is the problem this paper studies.

Modern deep Koopman learning estimates these functions in two stages. First, it learns a dictionary of present- and future-state observables from transition pairs; methods such as VAMPnet, DPNet and NeuralSVD differ in how this representation is parameterized and trained (Takeishi et al., 2017; Mardt et al., 2018; Kostic et al., 2024b; Ryu et al., 2024; Jeong et al., 2025; Turri et al., 2026). Second, it fits the operator over the learned features, for example by EDMD or canonical correlation analysis (CCA). This differs from classical approaches such as the linear inverse model (LIM) (Penland & Sardeshmukh, 1995), which uses a prescribed linear representation based on leading principal components of state anomalies. We call a dictionary learned directly from observed transition pairs a *data-trained dictionary*.

We find that this two-stage pipeline can substantially overfit, especially when the rank of the fitted operator is overspecified. As the rank increases, the training VAMP-E score (Wu & Noé, 2020) continues to improve while the held-out score eventually deteriorates (Figure 1), as McGibbon & Pande (2015) found for variational estimators on finite data. The same stochastic transitions are used both to select predictive features and to fit the operator over them. Consequently, a flexible dictionary can select correlations specific to the observed future realizations, and fitting the operator to those same transitions further amplifies the discrepancy between training and held-out scores. The effect is strongest at long lags, where less of the future is predictable from the present.

We study two ways to decouple dictionary learning from operator fitting. The first is *sample splitting*: train the dictionary and fit the operator on separate transition data. This reduces reuse of the same stochastic realizations, but spends data and still trains the dictionary on noisy futures. It is also difficult for atmospheric data, which are observed as a single temporally correlated trajectory rather than as independent realizations.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

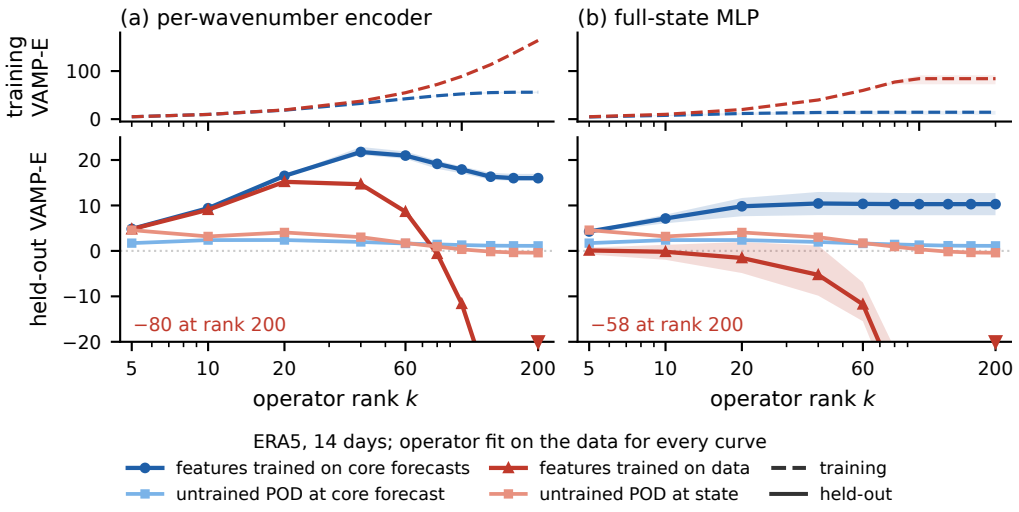

Figure 1: A dictionary learned from the data alone claims more with rank and keeps less of it than the core-trained dictionary. Training (dashed) and held-out (solid) VAMP-E against the operator's rank on ERA5 at fourteen days, with the dictionaries trained on 1979–2011 and scored on 2012–2020, under the per-wavenumber encoder (a) and the full-state MLP (b); the remaining settings are in Section 4.1 and, for (b), Appendix E.1.1.

Our main approach instead uses *model assistance*. We train the dictionary using forecasts from an approximate prediction model, which we call a *core model*, initialized from the observed present states. The observed transitions are then used only to fit the operator over this dictionary, which we call the *core-trained dictionary*. For atmospheric dynamics, the core is a dynamical core that integrates the equations of fluid motion. Unlike hybrid forecasting (Kochkov et al., 2024), the core is not part of the final predictor, and unlike transfer learning from climate simulations (Ham et al., 2019; Rasp & Thuerey, 2021), it does not pretrain the forecaster; it only supplies the target used to learn the representation (Appendix I). Because its forecast is fixed once the present state is given, transition noise, the part of the future the present state does not determine, does not enter the dictionary-training target.

Importantly, the usefulness of the core does not require an accurate state forecast. We show that the fitted operator can correct the component of the core's error represented within the learned feature spaces, so only the component outside those spaces is irrecoverable (Proposition 1). In our atmospheric experiments the dynamical core models only fluid motion, and within days its state forecast is farther from the realized future than the present state itself.

On ERA5 reanalysis, the core-trained dictionary overfits substantially less along rank and achieves a higher held-out VAMP-E than the data-trained dictionary from seven to fourteen days across all seeds (Figure 3; Section 4). A controlled spherical shallow-water benchmark isolates the role of transition noise, and there the advantage of the core-trained dictionary grows as a stochastic forcing omitted by the core becomes stronger (Figure 4; Section 5). Refitting the data-trained dictionary's operator on independent trajectories recovers most of this gap under the stronger forcings, directly implicating transition reuse, while the core-trained dictionary remains ahead without requiring additional independent trajectories (Figure 5; Section 6).

Contributions.

- We identify a generalization failure in deep Koopman learning caused in part by coupling dictionary learning and operator fitting on the same stochastic transitions, and quantify this effect through independent operator refitting.

- We introduce a model-assisted alternative that learns the dictionary from forecasts of an approximate core model while fitting the operator only to observed transitions, together with a projection characterization of which core errors the fitted operator can correct (Section 3).
- We demonstrate a higher held-out VAMP-E than the data-trained dictionary on ERA5 across long lags and on a controlled spherical shallow-water benchmark, where the margin grows with the strength of a stochastic forcing the core omits.

2 PREDICTABLE FUNCTIONS AND THEIR ESTIMATION

2.1 THE TRANSFER OPERATOR AND PREDICTABLE FUNCTIONS

The functions of the future state that the present state predicts best at a given lag are the leading singular functions of the transfer operator, and their squared singular values are their predictabilities. Let $\mathbf{x}_t$ evolve on $\mathcal{X} \subseteq \mathbb{R}^d$, let ρ_{01} be the joint law of the transition pair $(\mathbf{x}, \mathbf{y}) = (\mathbf{x}_t, \mathbf{x}_{t+\tau})$ at lag τ, with marginals ρ_0, ρ_1, and let $\mathcal{K} : L^2_{\rho_1}(\mathcal{X}) \to L^2_{\rho_0}(\mathcal{X})$ act by $(\mathcal{K}g)(\mathbf{x}) = \mathbb{E}[g(\mathbf{y}) \mid \mathbf{x}]$, the best mean-square forecast of $g(\mathbf{y})$ from $\mathbf{x}$. The transition noise of g is $g(\mathbf{y}) - (\mathcal{K}g)(\mathbf{x})$, the part of $g(\mathbf{y})$ that the present state does not determine. The variance of $g(\mathbf{y})$ splits into that of the noise and that of the best forecast $(\mathcal{K}g)(\mathbf{x})$, and the forecast's share is the predictability of g, the most any forecast from $\mathbf{x}$ explains (Appendix C.1). No additive noise model is assumed. A stochastically forced system has transition noise, and so does a deterministic one, chaotic or not, observed at a resolution that does not close, while a deterministic system observed in full has none. When $\mathcal{K}$ is Hilbert–Schmidt it has the singular value decomposition

$$\mathcal{K} = \sum_i \sigma_i \, \phi_i \otimes \psi_i, \tag{1}$$

with orthonormal singular functions ϕ_i of the present state and ψ_i of the future state, $\sigma_1 \geq \sigma_2 \geq \dots$, and $(\phi \otimes \psi)h \triangleq \langle \psi, h \rangle_{\rho_1} \phi$. The constant pair has $\sigma_1 = 1$, and every other ψ_i is centered with predictability σ_i^2, which makes it the i-th most predictable function of the future state (Appendix C.1).

2.2 DEEP KOOPMAN LEARNING IN TWO STAGES

Deep Koopman learning estimates the singular functions in two stages: dictionary learning trains a dictionary, and operator fitting fits an operator over it. A dictionary is a pair of maps $\mathbf{f}, \mathbf{g} : \mathcal{X} \to \mathbb{R}^K$ whose components f_i and g_i, the present-state and future-state features, span $U \subset L^2_{\rho_0}(\mathcal{X})$ and $V \subset L^2_{\rho_1}(\mathcal{X})$. Write the second moments $\mathsf{M}_{\rho_0}[\mathbf{f}] = \mathbb{E}[\mathbf{f}(\mathbf{x})\mathbf{f}(\mathbf{x})^\top]$ and $\mathsf{M}_{\rho_1}[\mathbf{g}] = \mathbb{E}[\mathbf{g}(\mathbf{y})\mathbf{g}(\mathbf{y})^\top]$, taken invertible, and the cross-moment $\mathsf{T}[\mathbf{f}, \mathbf{g}] = \mathbb{E}[\mathbf{f}(\mathbf{x})\mathbf{g}(\mathbf{y})^\top]$. The pairs in the data comprise a training block and a held-out block, and on a block each expectation is an average over its pairs. Let $\mathcal{P}_U$ be the orthogonal projection onto U in $L^2_{\rho_0}(\mathcal{X})$ and $\mathcal{P}_V$ the one onto V in $L^2_{\rho_1}(\mathcal{X})$. Norms and inner products of operators are the Hilbert–Schmidt ones.

Dictionary learning uses the low-rank approximation objective of Ryu et al. (2024), applied to transfer operators by Jeong et al. (2025). For a Hilbert–Schmidt $\mathcal{K}$ the objective is $\|\mathcal{K} - \sum_{i \leq K} f_i \otimes g_i\|^2 - \|\mathcal{K}\|^2$, whose minimizers span the leading K singular subspaces (Appendix C.3), and expanding the square leaves the first line below, which defines $\mathcal{L}$:

$$\mathcal{L}(\mathbf{f}, \mathbf{g}) \triangleq -2 \operatorname{Tr} \mathsf{T}[\mathbf{f}, \mathbf{g}] + \operatorname{Tr}\left(\mathsf{M}_{\rho_0}[\mathbf{f}] \, \mathsf{M}_{\rho_1}[\mathbf{g}]\right) \tag{2}$$

$$= \left\| \sum\nolimits_{i \leq K} f_i \otimes g_i - \mathcal{P}_U \mathcal{K} \mathcal{P}_V \right\|^2 - \|\mathcal{P}_U \mathcal{K} \mathcal{P}_V\|^2. \tag{3}$$

Neither line needs $\mathcal{K}$ to be Hilbert–Schmidt (Appendix C.3). Minimizing $\mathcal{L}$ therefore maximizes $\|\mathcal{P}_U \mathcal{K} \mathcal{P}_V\|^2$ minus the squared distance of $\sum_{i \leq K} f_i \otimes g_i$ from $\mathcal{P}_U \mathcal{K} \mathcal{P}_V$, the part of $\mathcal{K}$ that the two spans hold, and that distance is zero at the minimum.

The best any fit over the dictionary reaches in mean square is $\mathcal{P}_U \mathcal{K} \mathcal{P}_V$. Operator fitting estimates it on the training block by canonical correlation analysis with a shrinkage and a rank, giving the fitted operator $\widehat{\mathcal{K}}$ and its coefficient matrix $\mathsf{K}_{\mathrm{fit}}$; Appendices C.2 and D.1 give the formulas.

2.3 VAMP-E SCORE AND ITS DECOMPOSITION

The paper scores every fit by VAMP-E (Wu & Noé, 2020), which splits into what the spans reach and what the fit misses. For a coefficient matrix A on a block of pairs, with the projections and norms taken under the block's empirical law (Appendix C.2),

$$R_E(\mathsf{A}) \triangleq 2\,\mathrm{Tr}\left(\mathsf{A}^\top \mathsf{T}[\mathbf{f},\mathbf{g}]\right) - \mathrm{Tr}\left(\mathsf{A}^\top \mathsf{M}_{\rho_0}[\mathbf{f}]\,\mathsf{A}\,\mathsf{M}_{\rho_1}[\mathbf{g}]\right)$$
$$= \underbrace{\|\mathcal{P}_U \mathcal{K} \mathcal{P}_V\|^2}_{\text{VAMP-2 of the spans}} - \underbrace{\left\|\mathcal{P}_U \mathcal{K} \mathcal{P}_V - \textstyle\sum_{a,b\le K} \mathsf{A}_{ab}\, f_a \otimes g_b\right\|^2}_{\text{estimation error}}. \tag{4}$$

The first line is what is computed, and completing the square gives the second. The second line's first term, the VAMP-2 score of the spans (Wu & Noé, 2020), is the most any operator on the two spans scores on the block. Its second term, the estimation error, is the squared distance from $\mathcal{P}_U \mathcal{K} \mathcal{P}_V$ to the operator that A represents.

Dictionary learning trains the spans, which set the VAMP-2, and operator fitting keeps them and fits the coefficients, so on any one block it moves only the estimation error. At $\mathsf{A} = \mathsf{K}_{\mathrm{fit}}$, equation 4 scores the fitted operator; the zero operator scores zero, and an operator farther than it from $\mathcal{P}_U \mathcal{K} \mathcal{P}_V$ scores below zero. The margin between two dictionaries on the same block therefore splits the same way. Its span part is the first dictionary's VAMP-2 minus the second's, and its fit part is the second's estimation error minus the first's. When $\mathcal{K}$ is Hilbert–Schmidt, the fitted operator's squared error splits into two approximation terms, which the spans alone fix, and the estimation error (Appendix C.4). A VAMP-2 read on finitely many pairs exceeds its population value by a finite-sample bias (Appendix C.5).

The data-trained dictionary uses the noisy futures in both stages, and each use inflates one part of its generalization gap (Appendix C.6). Taking equation 4 at $\mathsf{K}_{\mathrm{fit}}$ on both blocks and subtracting splits the gap into those two parts, each norm under the empirical law of its block:

$$\underbrace{R_E(\mathsf{K}_{\mathrm{fit}})\big|_{\mathrm{tr}} - R_E(\mathsf{K}_{\mathrm{fit}})\big|_{\mathrm{ho}}}_{\text{generalization gap}} = \underbrace{\|\mathcal{P}_U \mathcal{K} \mathcal{P}_V\|^2_{\mathrm{tr}} - \|\mathcal{P}_U \mathcal{K} \mathcal{P}_V\|^2_{\mathrm{ho}}}_{\text{span part}}$$
$$+ \underbrace{\|\mathcal{P}_U \mathcal{K} \mathcal{P}_V - \widehat{\mathcal{K}}\|^2_{\mathrm{ho}} - \|\mathcal{P}_U \mathcal{K} \mathcal{P}_V - \widehat{\mathcal{K}}\|^2_{\mathrm{tr}}}_{\text{fit part}}. \tag{5}$$

Dictionary learning maximizes the training VAMP-2, so a flexible encoder keeps features whose agreement on the training pairs the noise raised, and the span part grows. Operator fitting fits the coefficients to the same moments, so it weights those features for agreement that does not recur held out, and the fit part grows. The reason is the training block's cross-moment, which includes the present-state features times the transition noise of the future-state features and has mean zero only for features fixed before the pairs are drawn.

3 MODEL ASSISTANCE

Sample splitting removes the reuse only from the fit part of equation 5. Model assistance removes it from both parts, because the futures in the data enter only operator fitting. What remains instead is the core's error outside the learned features, because the fitted operator corrects the part inside them (Proposition 1).

3.1 LEARN FROM THE CORE, FIT TO THE DATA

Model assistance trains the dictionary on each state in the data paired with the core's forecast of it, in place of its future in the data, and we call the result the core-trained dictionary $(\mathbf{f}_{\mathrm{core}}, \mathbf{g}_{\mathrm{core}})$ (Figure 2). A dynamical core is a deterministic map $\Phi_{\mathrm{core}} : \mathcal{X} \to \mathcal{X}$ that integrates approximate equations of motion over the lag. Its operator is composition, $(\mathcal{K}^{\mathrm{core}} g)(\mathbf{x}) = g(\Phi_{\mathrm{core}}\mathbf{x})$ from $L^2_{\rho_{\mathrm{core}}}(\mathcal{X})$, with ρ_{core} the law of $\Phi_{\mathrm{core}}\mathbf{x}$, to $L^2_{\rho_0}(\mathcal{X})$. It is the transfer operator of a transition from $\mathbf{x}$ that lands at $\Phi_{\mathrm{core}}\mathbf{x}$ with certainty and so has no transition noise.

The trained dictionary must supply one structural property, forecast containment, the condition that the core's forecast of every learned future-state feature is a linear combination of the present-state

features,

$$S \triangleq \mathcal{K}^{\text{core}} V = \text{span}\{g_{\text{core},1}(\Phi_{\text{core}}\mathbf{x}), \ldots, g_{\text{core},K}(\Phi_{\text{core}}\mathbf{x})\} \subseteq U. \tag{6}$$

We then fit the operator over the core-trained dictionary to the observed transitions of the same training block, with no split. The fit is the same as over the data-trained dictionary, with the same shrinkage and rank. The core's forecast is fixed by the present state, so the dictionary's training objective does not read which future follows which, and the transition noise reaches only the fit of the coefficients. The core-trained dictionary's future-state features, trained on the core's forecast, are evaluated at the future in the data, so the core supplies the functions and the data their arguments.

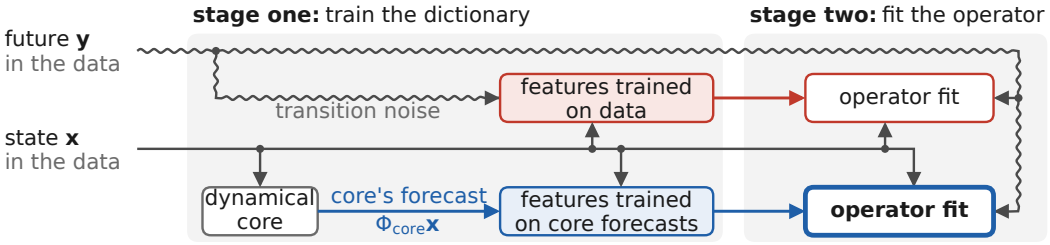

Figure 2: The core-trained dictionary's features are never trained on the data's futures, so the transition noise reaches only the operator fit. Dictionary learning trains the data-trained dictionary (red) on each state in the data paired with its future, and the core-trained dictionary (blue) on each state paired with the core's forecast of it; operator fitting fits both operators to the same states' futures in the data.

3.2 What forecast containment gives

Proposition 1 (What a fit over a containing span misses)**.** *Suppose equation 6, and write* $(\mathcal{K} - \mathcal{K}^{\text{core}})h$ *for the core's error on a future-state function* h*, whose value at* $\mathbf{x}$ *is* $\mathbb{E}[h(\mathbf{y}) - h(\Phi_{\text{core}}\mathbf{x}) \mid \mathbf{x}]$*. For every* h *in the span of the future-state features,* $\mathcal{K}^{\text{core}}h \in U$*, and*

$$\mathcal{P}_U \mathcal{K} h = \mathcal{K}^{\text{core}} h + \mathcal{P}_U(\mathcal{K} - \mathcal{K}^{\text{core}})h, \qquad (\mathcal{I} - \mathcal{P}_U)\mathcal{K}h = (\mathcal{I} - \mathcal{P}_U)(\mathcal{K} - \mathcal{K}^{\text{core}})h. \tag{7}$$

Under forecast containment equation 6, the best fit over the dictionary is the core's forecast plus the part of the core's error inside U (first equality; proof in Appendix C.7). The fit thus corrects the core's error inside U and misses only the part outside U. In the error split of Appendix C.4 that part is the input-side term, the part of the forecast of the future-state features that the present-state features cannot represent, and by the second equality

$$\underbrace{\|(\mathcal{I} - \mathcal{P}_U)\mathcal{K}\mathcal{P}_V\|^2}_{\text{forecast outside } U} \overset{\text{equation 7}}{=} \underbrace{\|(\mathcal{I} - \mathcal{P}_U)(\mathcal{K} - \mathcal{K}^{\text{core}})\mathcal{P}_V\|^2}_{\text{core's error outside } U} \leq \underbrace{\|(\mathcal{K} - \mathcal{K}^{\text{core}})\mathcal{P}_V\|^2}_{\text{core's error}}.$$

The core therefore needs its error to fall inside U, not an accurate forecast, and in the atmosphere most of its error does although its forecast is poor (Section 6). The other approximation term, the output-side term $\|\mathcal{K}(\mathcal{I} - \mathcal{P}_V)\|^2$, is untouched, since it depends on which future-state features the encoder reaches (Appendix C.4).

3.3 Why the core-trained dictionary meets forecast containment

Because the core is deterministic, every minimizer of the population objective of dictionary learning meets forecast containment. By equation 3 with $\mathcal{K}^{\text{core}}$ in place of $\mathcal{K}$ and $\mathcal{P}_V$ taken in $L^2_{\rho_{\text{core}}}(\mathcal{X})$, dictionary learning maximizes $\|\mathcal{P}_U \mathcal{K}^{\text{core}} \mathcal{P}_V\|^2$. Since composition preserves norms, this is at most $\|\mathcal{K}^{\text{core}}\mathcal{P}_V\|^2 = K$ for K linearly independent future-state features, with equality if and only if equation 6 holds. The objective is therefore flat, every V reaching the bound with $U = \mathcal{K}^{\text{core}}V$, and its minimizers are the dictionaries that meet equation 6 with $\sum_{i \leq K} f_i \otimes g_i = \mathcal{K}^{\text{core}}\mathcal{P}_V$. Dictionary learning against the core does not determine which containing span the encoder reaches, so the encoder's inductive bias decides it (Appendix C.8). Operator fitting ranks the functions in that span by the data.

The trained networks meet forecast containment approximately. A least-squares map from $\mathbf{f}_{\mathrm{core}}(\mathbf{x})$ with intercept explains an R^2 near 0.97 of the variance of $\mathbf{g}_{\mathrm{core}}(\Phi_{\mathrm{core}}\mathbf{x})$ on held-out states at seven, ten, and fourteen days. For random features of the same number, R^2 is about zero (Appendix C.7).

3.4 Two testable hypotheses

Model assistance leads to two testable hypotheses about the core-trained dictionary's gain over the data-trained dictionary. The noise hypothesis is that the gain grows with transition noise, which rises with lag and the strength of a forcing the core omits (Sections 4.3, 5.2, and 6). Along the lag, this remains a hypothesis, since a longer lag takes the core's forecast further from the future in the data. The core-error hypothesis is that a core whose error falls more outside the span gains less, although, by Proposition 1, a core with a worse forecast need not give a worse dictionary (Section 5.3).

4 The atmosphere: ERA5 reanalysis

4.1 Setup

The data are ERA5 at triangular truncation T21, with the dictionaries trained on 1979 to 2011 and every score read on 2012 to 2020. The core integrates the primitive equations, which model only the fluid motion, with none of the physical parameterizations, such as radiation and condensation, that a weather model adds. Both dictionaries use the same per-wavenumber encoder and training budget (Appendix B.1), three seeds and lags of one to fourteen days (Table 1 in the appendix). The same fit runs over two untrained POD bases, one read at the state and one at the core's forecast, the second meeting forecast containment by construction (Appendix B).

What is reported. Each dictionary is trained once with 200 features and read at any rank by truncating its fitted operator, so its spans stay fixed along a rank curve (Appendix C.9). Unless its sentence says otherwise, every number in the body is a held-out value under the fixed protocol, rank 60 and shrinkage 0.01 for every dictionary, lag and system. Figure 1 shows the whole rank curve at fourteen days, and Section 4.4 chooses the rank and shrinkage on validation rows instead. Under the selected protocol, named where used, each dictionary picks its shrinkage by two-fold selection on the halves of the held-out block, and the score is then read on the whole block. The margin is the core-trained dictionary's held-out score minus the data-trained dictionary's, and a configuration is one scored combination of system, lag, forcing, data size and seed. Intervals are basic 90% paired-bootstrap intervals (Appendix B.2).

4.2 Overfitting of the data-trained dictionary

From three to fourteen days the data-trained dictionary's held-out score turns down with rank while its training score keeps rising; at one day neither dictionary's turns down within its 200 features. At seven and fourteen days the data-trained dictionary's held-out score falls, at high rank, below zero, the score of the zero operator. The core-trained dictionary's peaks at a higher rank, falls a fraction as far and stays above the data-trained dictionary's past the peak (Figure 1 at fourteen days, Figure 7 in the appendix at both). At the shrinkage each dictionary selects at rank 60, the data-trained dictionary's still turns down with rank (row (b) of Figure 7 in the appendix).

4.3 The margin along the lag

In the seed mean the core-trained dictionary's held-out margin over the data-trained dictionary grows with the lag from three to fourteen days, as the noise hypothesis of Section 3.4 expects (Figure 3). At seven, ten and fourteen days every seed's margin lies between $+5.95$ and $+13.58$ with its basic 90% interval clearing zero; at five days two of the three intervals do, and at one and three days none does (Appendix D.2).

Reading the core's forecast linearly, through the untrained POD basis read at it, is best at three days, and both learned dictionaries beat it from five to fourteen days (Figure 3). It is also costly to use, because it needs the core integrated at every state it is evaluated at and a decomposition of an explicit ensemble of states, while the learned dictionary reads its K numbers from the present state.

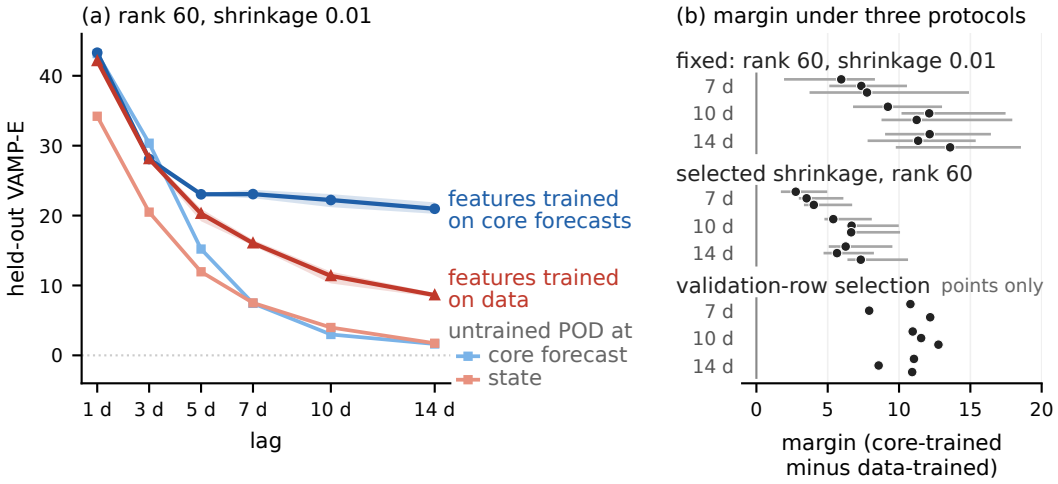

Figure 3: At the fixed protocol and in the seed mean the core-trained dictionary's held-out margin over the data-trained dictionary grows with the lag from three to fourteen days, and from seven to fourteen days it is positive in every seed under each protocol of (b). (a) Held-out VAMP-E against lag, the seed mean with a band spanning the three seeds. (b) The margin, one dot per seed, at the fixed protocol, at each dictionary's selected shrinkage, and with each dictionary's rank and shrinkage chosen on validation rows. Bars are basic 90% paired-bootstrap intervals (Appendix B.2).

4.4 Robustness

Neither each dictionary's selected shrinkage nor early stopping closes the margin. With rank and shrinkage chosen on validation rows of the training period and the held-out block read once, the core-trained dictionary leads at seven, ten and fourteen days in every seed (Appendix D.3). The margin also holds at T42 at seven and fourteen days and under a full-state MLP (Appendix E).

5 A controlled benchmark: spherical shallow water

5.1 Setup

The benchmark varies what the atmospheric data cannot: the strength of a stochastic forcing the core omits, the number of training trajectories and the core's own error. The truth is the one-layer shallow-water equations on the sphere at T42, started from barotropically unstable jets (Galewsky et al., 2004) and driven by stochastic kinetic energy backscatter (Berner et al., 2009). The backscatter ratio r is the share of the dissipated kinetic energy that the forcing returns. It is 0.02 as in Berner et al. (2009), or 0.3 and 1.0 set beyond it to move the axis (Appendix B.4). Two cores run the same equations without the forcing: the accurate core, at the truth's viscosity, and the less viscous core, at a tenth of it. The data are the first N trajectories of a pool, with N of 256 or 1024 (compared in Appendix F.3), and a fixed 64 are held out. Lags are one to fourteen days, and every dictionary and basis is scored by the estimator of Section 4 (Table 1 in the appendix).

5.2 Forcing strength

Wherever two forcings are scored, at either data size, the core-trained dictionary's margin is larger under the stronger one, as the noise hypothesis expects. At seven days and 256 trajectories the margin runs from $+9.35$ at $r = 0.02$ to $+26.00$ at $r = 1.0$ (Figure 4; Appendix F.1). Within each of the two trajectory sets the figure divides, the margin also grows with the lag. At seven to fourteen days under the two stronger forcings the data-trained dictionary's held-out score is below zero in eight of the ten configurations, and at seven days it falls with rank far past its peak (Figure 10 in the appendix). At each dictionary's selected shrinkage the core-trained dictionary stays ahead in every configuration at 256 trajectories from one to seven days, and the margins keep the forcing order there (Appendix D.3.1).

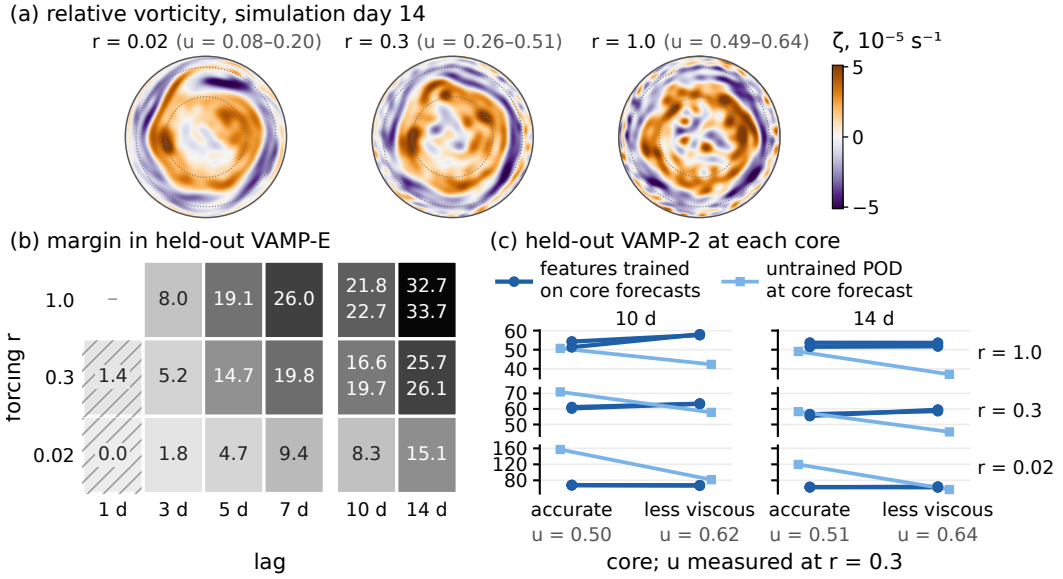

Figure 4: The core-trained dictionary's margin over the data-trained dictionary rises with the forcing the core omits, and under the two stronger forcings a less accurate core leaves the core-trained dictionary no worse in held-out VAMP-2 while the POD basis read at the core's forecast loses it. (a) The truth's relative vorticity on simulation day 14 under the three forcings, seen from above the North Pole; each label gives the accurate core's relative forecast error u over the lags in (b). (b) The margin at the accurate core, at 256 trajectories: seed 0 above seed 1 in a two-value box, hatching where a basic 90% interval covers zero, a dash where a lag is not scored, and a space between the twenty-day and the thirty-day trajectories. (c) Held-out VAMP-2 at each core: dark lines the core-trained dictionary, one per seed, light lines the POD basis read at the core's forecast, and a separate vertical scale for each row.

Under weak forcing the untrained POD basis read at the accurate core's forecast scores above the core-trained dictionary at seven to fourteen days (Figure 4(c); Appendix F.2), because this core integrates the truth's own equations at the truth's own viscosity and so forecasts close to an oracle.

5.3 Core error

As the core-error hypothesis of Section 3.4 allows, under the two stronger forcings a core with the larger relative forecast error $u = \|\mathbf{y} - \Phi_{\text{core}}\mathbf{x}\|/\|\mathbf{y} - \mathbf{x}\|$ need not give a worse dictionary. At ten and fourteen days the core with a tenth of the viscosity has the larger u, 0.62 against 0.50 at ten days and $r = 0.3$. Its dictionary's held-out VAMP-2, which no shrinkage enters, is nevertheless higher in both seeds at $r = 0.3$ at both lags and at $r = 1.0$ at ten days, and within 0.25 of the other core's at $r = 1.0$ and fourteen days (Appendix G.1.2).

6 Where the margin comes from

The features the data-trained dictionary selects look far more predictable on the training pairs than they are, as the span part of equation 5 expects. In the atmosphere they lose about 100 of their training VAMP-2 held out from five to fourteen days, and the core-trained dictionary's features at most 4. In the benchmark the data-trained dictionary's span loses more than the core-trained dictionary's, bias-corrected, in every configuration scored in Appendix G.2. This span loss is the span part of equation 5, which the noise argument expects to be larger when the dictionary is learned from the data's futures than from the core's target (Section 3.1).

Sample splitting. In the benchmark, independent refitting closes about three quarters of the core-trained dictionary's margin over the data-trained dictionary, while the core-trained dictionary's score

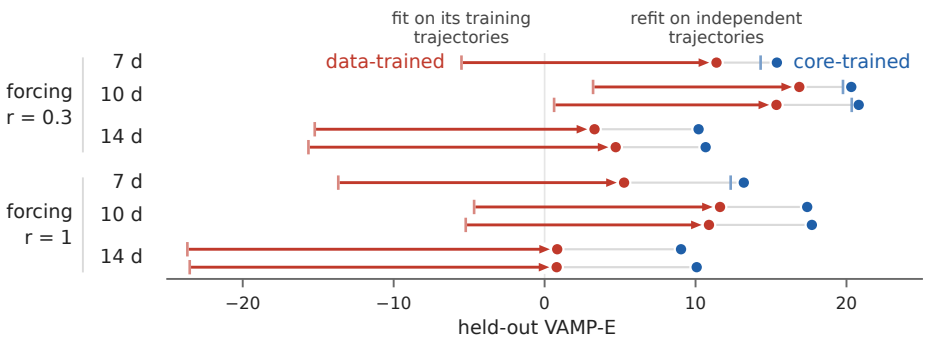

Figure 5: In the spherical shallow-water benchmark of Section 5, refitting the data-trained dictionary's operator on independent trajectories removes most of the core-trained dictionary's margin over it, and the core-trained dictionary stays ahead in every configuration shown. Its features are unchanged, and the core-trained dictionary barely moves under the same refit, so its tick often lies under its dot. A tick is the fit on its own training trajectories, a dot the refit, and a gray line the margin that remains (Appendix G.2.2).

barely moves (Figure 5). It fits the operator's coefficients on trajectories the features were not trained on, with the features frozen, in each configuration of Figure 5 under the two stronger forcings at seven to fourteen days. The two fits of the data-trained dictionary share a span and the same held-out rows, hence a held-out VAMP-2, so the recovery lies in the margin's fit part, not in the span loss above. Most of that margin therefore comes from reusing the transitions the span was trained on, an effect isolated here by intervention. Independent refitting, which is sample splitting and needs no core, still leaves the core-trained dictionary ahead in every one of those configurations (Figure 5; Table 12).

What the core supplies. The core helps as a source of features, not as a forecast. On ERA5 its forecast is further from the future in the data than the present state is at three to fourteen days (Appendix G.1.1). Yet a dictionary trained on its error shows that the error falls inside the core-trained span (Appendix H). At the extreme, the identity target pairs each state with itself and is free of transition noise but without the dynamics. Its dictionary scores below the core-trained dictionary in the benchmark, so the core's benefit depends on what its transitions contain (Appendix G.1.3).

7 Conclusion

We find that fitting a Koopman dictionary and its operator on the same stochastic transitions makes deep Koopman learning overfit, and that model assistance removes the dictionary's share of it without splitting the data. A model that only approximates the dynamics can thus supply the features while the data supply the fit, provided it can be started from states in the data. Its error must fall mostly in the learned span, as a dictionary trained on that error can check (Section 6).

Limitations. The atmospheric study runs at T21 and, at seven and fourteen days, also at T42 (Appendix E.2.1), with one dynamical core and one scalable per-wavenumber encoder, and a full-state MLP as a replication (Appendix E.1.1). The benchmark is barotropic, and its cores are far closer to their truth than the atmospheric core is (Appendix B.4). How far a core's dynamics can depart before its error leaves the span is not mapped, and the identity target marks the extreme. The core is deterministic, so dictionary learning takes no ranking from it, and the encoder's inductive bias picks the span, which reproduces across seeds (Appendices C.8 and E.3). A stochastic core averaged over an ensemble would give that ranking and would also shrink the core's error, since a deterministic forecast misses the mean of a stochastic future.

AI USE STATEMENT

In this work, we used generative AI tools to propose and refine hypotheses, to implement methods in code, and to clean and reformat datasets. We have not used them to generate synthetic data sets, to develop theoretical models or conceptual frameworks, to formulate mathematical claims, to provide ingredients for proofs or to write proofs, to design or give feedback on the research methodology or the experiments, to assist with translation, or to interpret results. Qualitative and thematic data analysis is not applicable to this work. Additionally, we used generative AI tools to edit the paper and to modify figures. We have reviewed all AI-assisted work, and we take responsibility for the final content of this work, including the text, the claims and the artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Appendix B holds the settings: Table 1 gives both systems, their splits, the training and the estimator, Table 2 the statistical procedures, Appendix B.3 the encoder, and Appendix B.4 the benchmark's integration and forcing. The experiments use Python 3.11 with JAX and jaxlib 0.10.1, Flax 0.12.8, Optax 0.2.8, NumPy 2.4.6 and Dinosaur 1.3.6. Each atmospheric dictionary behind Table 3 is trained data-parallel on four GPUs of one node, which split the batch of Table 1 evenly. They are NVIDIA A10 GPUs at one, three and seven days, and NVIDIA V100 GPUs at five, ten and fourteen days. The benchmark's dictionaries are trained the same way on two GPUs of one node, NVIDIA V100 or A40. Code will be released upon acceptance.

REFERENCES

Ferran Alet, Ilan Price, Andrew El-Kadi, Dominic Masters, Stratis Markou, Tom R. Andersson, Jacklynn Stott, Remi Lam, Matthew Willson, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Skillful joint probabilistic weather forecasting from marginals. *arXiv preprint arXiv:2506.10772*, 2025.

Peter J. Baddoo, Benjamin Herrmann, Beverley J. McKeon, J. Nathan Kutz, and Steven L. Brunton. Physics-informed dynamic mode decomposition. *Proceedings of the Royal Society A*, 2023. doi: 10.1098/rspa.2022.0576.

J. Berner, G. J. Shutts, M. Leutbecher, and T. N. Palmer. A spectral stochastic kinetic energy backscatter scheme and its impact on flow-dependent predictability in the ECMWF ensemble prediction system. *Journal of the Atmospheric Sciences*, 66(3):603–626, 2009. doi: 10.1175/2008JAS2677.1.

Kaifeng Bi, Lingxi Xie, Hengheng Zhang, Xin Chen, Xiaotao Gu, and Qi Tian. Accurate medium-range global weather forecasting with 3D neural networks. *Nature*, 619:533–538, 2023.

Cristian Bodnar, Wessel P. Bruinsma, Ana Lucic, Megan Stanley, Anna Allen, Johannes Brandstetter, et al. A foundation model for the Earth system. *Nature*, 641(8065):1180–1187, 2025. doi: 10.1038/s41586-025-09005-y.

Boris Bonev, Thorsten Kurth, Christian Hundt, Jaideep Pathak, Maximilian Baust, Karthik Kashinath, and Anima Anandkumar. Spherical fourier neural operators: Learning stable dynamics on the sphere. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Noah D. Brenowitz and Christopher S. Bretherton. Spatially extended tests of a neural network parametrization trained by coarse-graining. *Journal of Advances in Modeling Earth Systems*, 11: 2728–2744, 2019. doi: 10.1029/2019MS001711.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936. doi: 10.1007/BF02288367.

Bradley Efron. The estimation of prediction error: Covariance penalties and cross-validation. *Journal of the American Statistical Association*, 99(467):619–632, 2004. doi: 10.1198/016214504000000692.

Alban Farchi, Patrick Laloyaux, Massimo Bonavita, and Marc Bocquet. Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, 147(739):3067–3084, 2021. doi: 10.1002/qj.4116.

Joseph Galewsky, Richard K. Scott, and Lorenzo M. Polvani. An initial-value problem for testing numerical models of the global shallow-water equations. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):429–440, 2004. doi: 10.3402/tellusa.v56i5.14436.

Yoo-Geun Ham, Jeong-Hwan Kim, and Jing-Jia Luo. Deep learning for multi-year ENSO forecasts. *Nature*, 573(7775):568–572, 2019. doi: 10.1038/s41586-019-1559-7.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, New York, 2 edition, 2009. doi: 10.1007/978-0-387-84858-7.

Minchan Jeong, J. Jon Ryu, Se-Young Yun, and Gregory W. Wornell. Efficient parametric SVD of Koopman operator for stochastic dynamical systems. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 25564–25600. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/24826a1c623fe41706850748f3dc7370-Paper-Conference.pdf.

Dmitrii Kochkov, Janni Yuval, Ian Langmore, Peter Norgaard, Jamie Smith, Griffin Mooers, Milan Klöwer, James Lottes, Stephan Rasp, Peter Düben, Sam Hatfield, Peter Battaglia, Alvaro Sanchez-Gonzalez, Matthew Willson, Michael P. Brenner, and Stephan Hoyer. Neural general circulation models for weather and climate. *Nature*, 632:1060–1066, 2024. doi: 10.1038/s41586-024-07744-y.

Vladimir Kostic, Pietro Novelli, Andreas Maurer, Carlo Ciliberto, Lorenzo Rosasco, and Massimiliano Pontil. Learning dynamical systems via Koopman operator regression in reproducing kernel Hilbert spaces. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4017–4031, 2022.

Vladimir R. Kostic, Karim Lounici, Grégoire Pacreau, Giacomo Turri, Pietro Novelli, and Massimiliano Pontil. Neural conditional probability for uncertainty quantification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

Vladimir R. Kostic, Pietro Novelli, Riccardo Grazzi, Karim Lounici, and Massimiliano Pontil. Learning invariant representations of time-homogeneous stochastic dynamical systems. In *International Conference on Learning Representations*, 2024b.

Nikolaus Kriegeskorte, W. Kyle Simmons, Patrick S. F. Bellgowan, and Chris I. Baker. Circular analysis in systems neuroscience: the dangers of double dipping. *Nature Neuroscience*, 12(5): 535–540, 2009. doi: 10.1038/nn.2303.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382:1416–1421, 2023. doi: 10.1126/science.adi2336.

Christian Lessig, Ilaria Luise, Bing Gong, Michael Langguth, Scarlet Stadtler, and Martin Schultz. Atmorep: A stochastic model of atmosphere dynamics using large scale representation learning. *arXiv preprint arXiv:2308.13280*, 2023.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018. doi: 10.1038/s41467-018-07210-0.

Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. VAMPnets for deep learning of molecular kinetics. *Nature Communications*, 9(1):5, 2018. doi: 10.1038/s41467-017-02388-1.

Robert T. McGibbon and Vijay S. Pande. Variational cross-validation of slow dynamical modes in molecular kinetics. *The Journal of Chemical Physics*, 142:124105, 2015. doi: 10.1063/1.4916292.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41:309–325, 2005. doi: 10.1007/s11071-005-2824-x.

Cecile Penland and Prashant D Sardeshmukh. The optimal growth of tropical sea surface temperature anomalies. *Journal of Climate*, 8(8):1999–2024, 1995.

David Pfau, Stig Petersen, Ashish Agarwal, David G. T. Barrett, and Kimberly L. Stachenfeld. Spectral inference networks: Unifying deep and spectral learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R. Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, Remi Lam, and Matthew Willson. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044):84–90, 2025. doi: 10.1038/s41586-024-08252-9.

Stephan Rasp and Nils Thuerey. Data-driven medium-range weather prediction with a Resnet pretrained on climate simulations: A new model for WeatherBench. *Journal of Advances in Modeling Earth Systems*, 13(2):e2020MS002405, 2021. doi: 10.1029/2020MS002405.

Stephan Rasp, Michael S. Pritchard, and Pierre Gentine. Deep learning to represent subgrid processes in climate models. *Proceedings of the National Academy of Sciences*, 115(39):9684–9689, 2018. doi: 10.1073/pnas.1810286115.

J. Jon Ryu, Xiangxiang Xu, H. S. Melihcan Erol, Yuheng Bu, Lizhong Zheng, and Gregory W. Wornell. Operator SVD with neural networks via nested low-rank approximation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Ryan J. Tibshirani. Degrees of freedom and model search. *Statistica Sinica*, 25(3):1265–1296, 2015. doi: 10.5705/ss.2014.147.

Giacomo Turri, Luigi Bonati, Kai Zhu, Massimiliano Pontil, and Pietro Novelli. Self-supervised evolution operator learning for high-dimensional dynamical systems. In *International Conference on Learning Representations (ICLR)*, 2026.

Claire Valva and Dimitrios Giannakis. Physics-informed spectral approximation of Koopman operators, 2025. URL https://arxiv.org/abs/2408.05663.

Oliver Watt-Meyer, Noah D. Brenowitz, Spencer K. Clark, Brian Henn, Anna Kwa, Jeremy McGibbon, W. Andre Perkins, and Christopher S. Bretherton. Correcting weather and climate models by machine learning nudged historical simulations. *Geophysical Research Letters*, 48:e2021GL092555, 2021. doi: 10.1029/2021GL092555.

Matthew O. Williams, Ioannis G. Kevrekidis, and Clarence W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015. doi: 10.1007/s00332-015-9258-5.

Hao Wu and Frank Noé. Variational approach for learning Markov processes from time series data. *Journal of Nonlinear Science*, 30(1):23–66, 2020. doi: 10.1007/s00332-019-09567-y.

Janni Yuval and Paul A. O'Gorman. Stable machine-learning parameterization of subgrid processes for climate modeling at a range of resolutions. *Nature Communications*, 11:3295, 2020. doi: 10.1038/s41467-020-17142-3.

CONTENTS OF THE APPENDIX

A Notation . . . 14

B Experimental setting . . . 15

B.1 Systems and estimator . . . 15

B.2 Statistical procedures . . . 16

B.3 Encoder and input normalization . . . 17

B.4 Shallow-water benchmark . . . 18

C Derivations and proofs . . . 19

C.1 Predictability of a singular function . . . 19

C.2 Decomposing VAMP-E . . . 19

C.3 Where the Hilbert–Schmidt condition is needed . . . 20

C.4 The fit's three error terms . . . 20

C.5 The three terms from finite samples . . . 21

C.6 How transition reuse inflates the gap . . . 22

C.7 Proof of Proposition 1 . . . 22

C.8 Flat objective against the core . . . 22

C.9 Operator rank changes only the fit . . . 23

D The margin under each protocol . . . 24

D.1 Shrinkage and rank . . . 24

D.2 Margins at the fixed protocol . . . 25

D.3 Letting each dictionary choose its settings . . . 26

E Robustness of the atmospheric margin . . . 29

E.1 Encoder . . . 29

E.2 Data . . . 30

E.3 Span reproducibility across seeds . . . 31

F Benchmark results . . . 32

F.1 Benchmark margins at one to seven days . . . 32

F.2 Benchmark curves along lag and rank . . . 32

F.3 Generalization gaps by data size . . . 33

G Where the margin comes from . . . 34

G.1 What the core supplies . . . 34

G.2 Span and fit parts of the margin . . . 36

H Additions to the core-trained dictionary . . . 37

H.1 Combined dictionary . . . 37

H.2 Remainder dictionary . . . 38

I Extended related work . . . 38

A NOTATION

symbol	meaning
States, laws and the core	
$\mathbf{x}$, $\mathbf{y} = \mathbf{x}_{t+\tau}$	present and future state, at lag τ
ρ_{01}; ρ_0, ρ_1	joint law of the transition pair; its marginals
Φ_{core}	the dynamical core, a deterministic map over the lag
u	the core's relative forecast error, $\|\mathbf{y}-\Phi_{\text{core}}\mathbf{x}\|/\|\mathbf{y}-\mathbf{x}\|$, each norm taken once over the whole held-out block in the unweighted spectral coefficients
Operators	
$\mathcal{K}$	transfer operator of the observed system, $(\mathcal{K}g)(\mathbf{x}) = \mathbb{E}[g(\mathbf{y}) \mid \mathbf{x}]$
$\mathcal{K}^{\text{core}}$	the core's composition operator, $(\mathcal{K}^{\text{core}}g)(\mathbf{x}) = g(\Phi_{\text{core}}\mathbf{x})$
$\mathcal{K}^{\text{corr}}$	the core's error, $\mathcal{K} - \mathcal{K}^{\text{core}}$
σ_i; ϕ_i, ψ_i	singular values; present-state and future-state singular functions of $\mathcal{K}$
$\mathcal{I}$	identity operator
Dictionaries and spans	
$\mathbf{f}$, $\mathbf{g}$	present-state and future-state features of a dictionary, each in $\mathbb{R}^K$
A; $\mathcal{A}$	a coefficient matrix over the dictionary; the operator it represents, $\sum_{a,b\le K} \mathsf{A}_{ab} f_a \otimes g_b$ (Appendix C.2)
$\mathbf{f}_{\text{core}}$, $\mathbf{g}_{\text{core}}$	the core-trained dictionary's features
B	the untrained POD basis, 200 left singular vectors of the training states (Appendix B)
$\mathbf{f}_{\text{corr}}$, $\mathbf{g}_{\text{corr}}$	the correction dictionary's features (Appendix H)
U; $\mathcal{P}_U$	span of the present-state features; its orthogonal projection, least-squares regression on the features (Appendix C.2)
V; $\mathcal{P}_V$	span of the future-state features; its orthogonal projection, on which the error equation 10 is measured
S	span of the core's forecast of the future-state features
U_0	the frozen block a remainder dictionary is trained against: a constant with a prefix of $\mathbf{f}_{\text{core}}$
Objective, fit and scores	
$\mathsf{M}_{\rho_0}[\mathbf{f}]$, $\mathsf{M}_{\rho_1}[\mathbf{g}]$	second moments of the features
$\mathsf{T}[\mathbf{f}, \mathbf{g}]$	cross-moment $\mathbb{E}[\mathbf{f}(\mathbf{x})\mathbf{g}(\mathbf{y})^\top]$
$\mathcal{L}(\mathbf{f}, \mathbf{g})$	the low-rank approximation objective equation 2
$\bar{\mathsf{C}}_{01}$	whitened cross-covariance equation 9
$\widehat{\sigma}_i$; $\mathbf{u}_i$, $\mathbf{v}_i$	its singular values; its present-state and future-state singular vectors in $\mathbb{R}^K$
$\widehat{\mathcal{K}}$; K_{fit}	the fitted operator, which acts only on V and maps into U; its matrix, fitted on training moments
R_E	VAMP-E equation 4; primes mark moments of another block
$\|\mathcal{P}_U\mathcal{K}\mathcal{P}_V\|^2_{\text{HS}}$	the largest score the two spans allow; its estimate on held-out rows is the held-out VAMP-2 (Appendix C.5)
Widths, keys and benchmark settings	
K	the number of features a dictionary is trained with; 200 unless another number is stated
k	the rank the fitted operator is truncated to; 60 in the body
λ	the shrinkage added to each second moment before whitening, relative to the mean eigenvalue; 0.01 in the body
k_c	the number of leading core features a second block is added to
r	the benchmark's backscatter ratio, the share of the dissipated kinetic energy its forcing returns
N	the number of training trajectories in the benchmark, 256 or 1024
configuration	one scored combination of system, lag, forcing, data size and seed

B EXPERIMENTAL SETTING

B.1 SYSTEMS AND ESTIMATOR

Both systems' cores are integrated with Dinosaur 1.3.6,[1] the spectral dynamical core released with Kochkov et al. (2024). Table 1 collects the settings of both systems. The benchmark's rows list what each setting ranges over and not a crossed grid: not every lag is scored at every forcing and data size, and Table 8 lists every configuration scored at one to seven days, and Figure 4 draws every lag at 256 trajectories.

Atmosphere. The atmospheric core advects humidity, which enters through the virtual temperature, and has no condensation, evaporation, radiation or surface fluxes, over a flat surface. The atmospheric state is the core's spectral coefficients at T21, 32,637 entries of which 16,687 vary, sampled every six hours from 1979. That gives 48,180 training pairs at every lag and, after an embargo of fourteen days plus the lag, 13,120 held-out pairs at one day falling to 13,016 at fourteen. Where a dictionary is early-stopped, the validation block is the last tenth of the training window behind the same embargo, training stops after 20,000 steps without a better validation score, and the best checkpoint is read.

Training. The margin compares two training targets, so the two dictionaries must otherwise be trained alike. At each of the six lags the core-trained and data-trained dictionaries of Table 3 share the step budget (120,000), the learning rate (0.005) and the capacity (200 features, width 512). Every dictionary in either system is trained once under the nested form of the objective (Appendix B.3), with 200 features except for the additions of Appendix H, and every score is read from that one training as Appendix D.1 describes. A seed sets the two encoders' initial parameters and the order of the minibatches; neither the data nor the split is randomized, the split being by date in the atmosphere and by trajectory in the benchmark.

POD bases. Both untrained bases are built from one matrix B of 200 columns: the leading left singular vectors, taken without removing their mean, of about 3,000 training states at a stride of 16, an economical estimate of the training block's proper orthogonal decomposition, over the coordinates whose variance there is nonzero. The POD basis read at the state has present-state features $\mathbf{f}(\mathbf{x}) = \mathsf{B}^\top \mathbf{x}$ and future-state features $\mathbf{g}(\mathbf{y}) = \mathsf{B}^\top \mathbf{y}$, and the POD basis read at the core's forecast keeps that $\mathbf{g}$ and has $\mathbf{f}(\mathbf{x}) = \mathsf{B}^\top \Phi_{\text{core}} \mathbf{x}$. Neither is trained or pairs a state in the data with its future, and the operator over each is the fit of Section 2.2 on the same training pairs, with the same shrinkage and rank as over the data-trained and core-trained dictionaries. Without shrinkage and at full rank, that fit over a fixed dictionary is the least-squares estimate of extended dynamic mode decomposition (Williams et al., 2015), and for the POD basis read at the core's forecast it is the same regression with $\Phi_{\text{core}} \mathbf{x}$ in place of the present state.

The POD basis read at the state is the dictionary of the linear inverse model (Penland & Sardeshmukh, 1995), which forms it from anomalies rather than from the states themselves. The POD basis read at the core's forecast is the untrained counterpart of the core-trained dictionary: its present-state features are its future-state features evaluated at the core's forecast, $\mathbf{f} = \mathbf{g} \circ \Phi_{\text{core}}$, so it meets forecast containment equation 6 by construction and Proposition 1 applies to it exactly. It tests whether training on the core adds anything beyond reading that forecast linearly. Both need the explicit ensemble of training states, which exists at this resolution and not at an operational one.

Early stopping. The configurations of Table 3 are trained to the full budget, without early stopping. Early stopping is used only in two checks. At ten and fourteen days (Appendix D.3.3) each data-trained dictionary stops at the step its own training-period validation selects, between 12,000 and 14,000, while the core-trained dictionaries keep the full budget. The configurations of different training lengths in Appendix E.2.2 stop both dictionaries, and the benchmark uses no early stopping.

[1] https://github.com/neuralgcm/dinosaur, tag v1.3.6, commit 31d5527.

Table 1: The two systems. A value that spans both columns holds in both systems. The estimator row is the protocol the body reports in both systems; Appendix D.3.1 gives the same configurations at each dictionary's own selected shrinkage.

	atmosphere	shallow water
System		
data	ERA5 at T21	simulated at T42
truth	reanalysis	one-layer shallow water with stochastic forcing at backscatter r of 0.02, 0.3 or 1.0
core	the primitive equations, fluid motion only (no physical parameterizations), over a flat surface	the same equations without the forcing, at the truth's viscosity or a tenth of it
Split and lags		
training block	1979 to 2011	the first N of the trajectories, N of 256 or 1024
held out	from 14 days plus the lag after the last training state (2012-01-08 at one day to 2012-01-21 at fourteen) to 2020-12-31	64 further trajectories
lags	1, 3, 5, 7, 10 and 14 days	
Encoder and training		
encoder	per-wavenumber, width 512, 200 features	
optimizer	Adam at a constant learning rate of 0.005 (0.01 for the correction dictionary of Appendix H), no warmup or decay	
steps and batch	120,000 steps; batch of 512 pairs split evenly across the devices; per-sample input normalization	
seeds	3	1, with a second at 10 and 14 days at N of 256 at every forcing
Estimator		
fixed protocol	rank 60, shrinkage 0.01	
selected protocol	shrinkage chosen from 0, 0.01, 0.1, 0.3 and 1 (Appendix D.3.1)	

B.2 Statistical procedures

Table 2 gives the resampling, the bias correction, the selections of rank and shrinkage and the random baselines used in this paper, with their settings.

In the paired moving-block bootstrap, block starts are uniform, and a benchmark block may cross a trajectory boundary, since the held-out rows are ordered trajectory by trajectory, at 44 to 72 pairs each. For the point estimate θ and the replicates' quantiles $q_{0.05}$ and $q_{0.95}$, the basic interval $[2\theta - q_{0.95},\ 2\theta - q_{0.05}]$ reflects the quantiles about θ, and the percentile interval is $[q_{0.05}, q_{0.95}]$.

Three consequences of these procedures decide how a number here is read.

No bootstrap interval resamples the training: a bootstrap interval says how far a score moves when the held-out rows move, and nothing about how far it moves when the training does. Seed-to-seed variation is the larger of the two in several cases, which is why seeds are reported separately rather than pooled into one interval.

The replicates of a VAMP-E score are biased, and the bias is not the same for two dictionaries of different numbers of features or conditioning. A basic interval at the level of one configuration is therefore not a reliable statement about that configuration, since it can exclude its own point estimate. At the fixed and selected protocols that happens in two configurations of Table 4, five of Table 5, and one at the benchmark's fixed protocol, three days at $r = 1.0$ (Table 8). At the atmosphere's fixed protocol it happens in none of the eighteen configurations, under either kind of interval; what those configurations show instead is the disagreement about zero that the next paragraph counts. An interval that behaves this way is evidence that the replicates sit away from the point estimate by more than their spread, the bias above, not that the point estimate is wrong.

Table 2: The statistical procedures and their settings, and where each is used.

procedure	what it answers	settings
paired moving-block bootstrap	how far a score or a margin moves when the held-out rows move; every interval in the paper	operator fitted once and held fixed; only the held-out rows resampled, in moving blocks of 120 rows, 30 days in the atmosphere; 400 replicates, the same rows for every dictionary in a replicate; basic 90% intervals, with the percentile interval beside them where a table gives both
two-fold selection	which shrinkage, or rank and shrinkage, a dictionary picks for itself on the held-out block (Sections 4.2 to 4.4 and 5.2; Appendices D.3.1 and D.3.2)	each dictionary separately, over the shrinkage grid 0 to 1; the selected protocol holds rank 60, picks over the two halves of the held-out block and reads the whole block; the joint variant adds ten ranks from 5 to 200, picks on one half and scores the other
validation-row selection	which rank and shrinkage a dictionary picks without reading the held-out block (Sections 4.3 and 4.4; Appendix D.3.3)	ten ranks from 5 to 200 with the shrinkage grid, picked on the validation block of Appendix B, the last 4,818 rows of the training window, each dictionary separately; the held-out block scored once and whole; each bootstrap replicate re-makes the pick on resampled validation rows
jackknife	the finite-sample bias of a VAMP-2 (Section 6; Appendix G.2)	delete one unit: a calendar year in the atmosphere, of 33 training and 9 held-out years; in the benchmark one of the 64 held-out trajectories, or one of 32 groups of eight training trajectories; with θ the value on all n units and $\bar{\theta}$ the mean of the n delete-one values, the bias-corrected value is $n\theta - (n-1)\bar{\theta}$
random baselines	whether an ordering, a containment or a span overlap exceeds what random features or subspaces give (Section 3.3; Appendices C.8 and E.3)	for the rank correlation between per-pair scale and persistence, the 95th percentile over 1000 permutations of the feature index, with the persistence of 500 random unit directions of the same span beside it; for the containment R^2, the same number of i.i.d. standard normal features; for the overlap of two seeds' leading-k spans, random k-dimensional subspaces of the two seeds' measured 59-dimensional spans, whose expected overlap is $k/59$ times that of the full spans

The body reads the basic interval, and both kinds are computed from the same replicates. At the fixed protocol the two differ only at one day, where 0 against 2 of 3 seeds separate, and at five days, where 2 against 1 of 3 do. At the selected protocol, whose percentile interval Table 4 gives beside the basic one, they differ only at five days, 2 against 0 of 3. They differ where the replicates sit away from the point estimate, which the basic interval corrects and the percentile interval does not; from seven days they give the same verdict in every seed. Basic and percentile intervals move in opposite ways from the replicates' shift, so where the shift is large they can disagree about whether a margin excludes zero. We therefore read a sign that repeats across configurations rather than any one configuration's interval, and we say which when a single interval is quoted.

B.3 Encoder and input normalization

The encoder reads the state's modal coefficients directly, and nothing standardizes them before it. The T21 state has 33 channels in the core's own order: vorticity, divergence and temperature variation on eight sigma levels each, the logarithm of surface pressure, and specific humidity on eight levels. Each channel is a 43×23 block indexed by zonal wavenumber from -21 to 21 and total wavenumber l from 0 to 22, so a state has 32,637 entries.

The network has one hidden layer and no residual path. It normalizes the sample, then mixes the channels at each total wavenumber through a separate $33 \to 512$ map per l, shared across zonal

wavenumbers and with no bias, and applies a GELU. It then pools over the sphere, summing over the coefficients with one learned weight per l and dividing by 989, and maps the 512 hidden units that result to the dictionary's 200 features. The encoder therefore mixes channels within each spectral location before pooling across locations. Nothing inside the network normalizes the representation after the input.

Per-sample normalization takes one mean and one variance over all $32{,}637$ entries of a state, structural zeros included, and follows them with a learned scale and shift per entry. It is taken over the sample and not over the batch, and it equalizes a state's overall scale rather than its channels: temperature keeps 0.999 of its temporal variance through it.

No constant feature is prepended to the 200, and the first feature converges to the constant direction on its own, with a scale the training solves for of one in every seed. Each feature is divided by its uncentered root-mean-square over the global batch before the loss is taken, with gradients passing through the division. The two networks output unit-norm directions, and the per-pair scales, one for each pair (f_i, g_i), are solved in closed form at each step from a moving average of the batch moments rather than learned by gradient. Training computes the objective in its nested form, in which each feature is updated against the features before it, held fixed, so every prefix of the features is itself a dictionary and the ordering is part of what is learned. The present-state and future-state networks are of the same class with the same settings, share no parameters, and start from different initialization keys.

B.4 Shallow-water benchmark

The forcing is the constant-dissipation variant of spectral stochastic kinetic energy backscatter. The relative forecast error that bounds the axis is read at one day and at a finer resolution than the grid scored here, so it stands beside these configurations and not in one series with them. It is 0.1368, 0.1655, 0.2830, 0.4128 and 0.5158 at r from 0.02 to 1, and 0.5443, 0.5549 and 0.5678 at r of 2, 3 and 5. A fiftyfold rise in the injection to $r = 1$ buys $+0.379$ of that fraction and a fivefold rise past it buys $+0.052$, which is why the axis stops there.

The truth and both cores are integrated as one-layer shallow water at T42, with a third-order semi-implicit Runge–Kutta step at a timestep of 300 seconds, and the state is saved every six hours. The Laplacian diffusion is applied at each step, at the viscosity Section 5 gives for each core. Beside it the model applies an exponential step filter, which multiplies the coefficient at total wavenumber l by $\exp\{-(\Delta t/\tau)(l/l_{\max})^{36}\}$ with $\tau = 75$ seconds, so that the smallest resolved scale is damped by e^{-4} at each step and the large scales are left alone. The truth and both cores share the filter, so a core differs from the truth only through the forcing and, for the less viscous core, through the viscosity. A core is advanced over the lag from every state in the data of the truth.

The forcing is added by operator splitting: after each unforced step it adds $\Delta t\, a\, \nabla^2\psi$ to the vorticity. Here ψ is a random field of unit surface-mean variance whose amplitude at total wavenumber n is proportional to $n^{-1.27}$ for $n \geq 1$, with $n = 0$ excluded, and it advances one step of a first-order autoregressive process with a correlation time of six hours. The scale a is set so that the energy the forcing injects is r times what the unforced run dissipates, which is what fixes the ratios Section 5 quotes.

The truth carries a Laplacian viscosity $\nu = 10^5\ \mathrm{m^2\,s^{-1}}$, and the perturbation's latitude and longitude are drawn at random. The two weaker ratios measure 0.021 and 0.300; the amplitude of the strongest is interpolated between two amplitudes whose ratios measure 0.989 and 1.029, and the ratio at that amplitude is not itself measured. Each trajectory runs 20 days, or 30 for the ten- and fourteen-day configurations, and the first two days are dropped in either case.

The step and the filter are the model's own defaults, and the benchmark does not change them. The initial state is that of the barotropic-instability case of Galewsky et al. (2004), as the model implements it, and the dissipation is a Laplacian viscosity. That diffusion was chosen there so that the day-six vorticity converges near T85, and the case is tested from T42 upwards, which is why T42 is the native resolution here and nothing is truncated.

Two things are ours rather than inherited. The height bump's longitude is drawn at random as well as its latitude, where the model's own routine draws the latitude alone; with no orography a shift in longitude gives a rotated solution, so it adds variety and no new dynamics. And the forcing's

amplitude is anchored on energy. Under every forcing used, the large-scale energy spectrum keeps its shape, and the jet, the growth of the instability and its roll-up stay recognizable in the vorticity fields.

What the benchmark stands for is one-layer barotropic dynamics: a midlatitude jet going unstable and rolling up into vortices, with a stochastic small-scale forcing in place of the processes a model does not resolve. It has no baroclinic instability, no moisture, no radiation and no topography. Its cores solve the truth's own equations without the forcing, so they stay far closer to their truth than the atmospheric core does.

Each trajectory set holds 4160 trajectories from one generator seed, with the longitude of the initial perturbation drawn at random. The training trajectories are the first N of the first 4096 and the held-out ones are the last 64, so no trajectory is in both. A pair is formed at every six-hourly save after the spin-up whose partner at the lag exists. A trajectory of d days at a lag of τ days therefore gives $4d - 8 - 4\tau$ pairs: 60, 52 and 44 at three, five and seven days, and 72 and 56 at ten and fourteen. Consecutive pairs from one trajectory are six hours apart and are not independent.

C Derivations and proofs

C.1 Predictability of a singular function

Let g be a future-state function with $\mathbb{E}_{\rho_1}[g] = 0$ and $\mathbb{E}_{\rho_1}[g^2] = 1$. The law of total variance gives $\mathrm{Var}(g(\mathbf{y})) = \mathbb{E}_{\rho_0}[\mathrm{Var}(g(\mathbf{y}) \mid \mathbf{x})] + \mathrm{Var}_{\rho_0}(\mathbb{E}[g(\mathbf{y}) \mid \mathbf{x}])$. The conditional mean is $(\mathcal{K}g)(\mathbf{x})$, whose mean is $\mathbb{E}_{\rho_1}[g] = 0$, so its variance is $\|\mathcal{K}g\|^2_{\rho_0} = \langle \mathcal{K}^*\mathcal{K}g, g\rangle_{\rho_1}$, and the unit variance splits into a part no function of the present state predicts and a part $\mathcal{K}$ explains,

$$1 = \mathrm{Var}_{\rho_1}(g(\mathbf{y})) = \underbrace{\mathbb{E}_{\rho_0}\big[\mathrm{Var}(g(\mathbf{y}) \mid \mathbf{x})\big]}_{\text{irreducible}} + \underbrace{\langle \mathcal{K}^*\mathcal{K}\, g,\, g\rangle_{\rho_1}}_{\text{predictability}}. \tag{8}$$

The constant is a singular pair with $\sigma_1 = 1$, since $\mathcal{K}1 = 1$ and $\mathcal{K}$ is a contraction, and every other ψ_i is orthogonal to it and so centered; for such $g = \psi_i$, equation 1 gives $\mathcal{K}\psi_i = \sigma_i\phi_i$ with $\|\phi_i\|_{\rho_0} = 1$, so the predictability is σ_i^2. Both terms of equation 8 are non-negative and sum to one, so $\sigma_i \le 1$. No forecast of $g(\mathbf{y})$ from $\mathbf{x}$ has a smaller mean-square error than $\mathcal{K}g$: the transition noise $g(\mathbf{y}) - (\mathcal{K}g)(\mathbf{x})$ is orthogonal to every function of the present state, so $\mathbb{E}[(g(\mathbf{y}) - h(\mathbf{x}))^2] = \mathbb{E}[(g(\mathbf{y}) - (\mathcal{K}g)(\mathbf{x}))^2] + \|\mathcal{K}g - h\|^2_{\rho_0}$ for every $h \in L^2_{\rho_0}(\mathcal{X})$. The predictability is therefore the most any forecast of $g(\mathbf{y})$ from $\mathbf{x}$ explains.

C.2 Decomposing VAMP-E

Let $\mathcal{P}_U$ and $\mathcal{P}_V$ project onto the spans of the two feature sets under the law the expectations are taken over, let A be any $K \times K$ coefficient matrix, and write $\mathcal{A} = \sum_{a,b\le K} \mathsf{A}_{ab}\, f_a \otimes g_b$ for the operator it represents. For one pair of features, $\langle f_a \otimes g_b, \mathcal{K}\rangle_{\mathrm{HS}} = \langle f_a, \mathcal{K}g_b\rangle_{\rho_0} = \mathbb{E}[f_a(\mathbf{x})(\mathcal{K}g_b)(\mathbf{x})] = \mathbb{E}[f_a(\mathbf{x})g_b(\mathbf{y})]$, the last step by the tower property, and $\langle f_a \otimes g_b, f_c \otimes g_d\rangle_{\mathrm{HS}} = \langle f_a, f_c\rangle_{\rho_0}\langle g_b, g_d\rangle_{\rho_1} = (\mathsf{M}_{\rho_0}[\mathbf{f}])_{ac}(\mathsf{M}_{\rho_1}[\mathbf{g}])_{bd}$. Summing both over the coefficients gives $\mathrm{Tr}(\mathsf{A}^\top \mathsf{T}[\mathbf{f}, \mathbf{g}]) = \langle \mathcal{A}, \mathcal{K}\rangle_{\mathrm{HS}}$ and $\mathrm{Tr}(\mathsf{A}^\top \mathsf{M}_{\rho_0}[\mathbf{f}]\, \mathsf{A}\, \mathsf{M}_{\rho_1}[\mathbf{g}]) = \|\mathcal{A}\|^2_{\mathrm{HS}}$, so the score equation 4 is $2\langle \mathcal{A}, \mathcal{K}\rangle_{\mathrm{HS}} - \|\mathcal{A}\|^2_{\mathrm{HS}}$. Since $\mathcal{A}$ has its present-state functions in U and vanishes off V, $\langle \mathcal{A}, \mathcal{K}\rangle_{\mathrm{HS}} = \langle \mathcal{A}, \mathcal{P}_U\mathcal{K}\mathcal{P}_V\rangle_{\mathrm{HS}}$, and completing the square gives

$$R_E(\mathsf{A}) = \|\mathcal{P}_U\mathcal{K}\mathcal{P}_V\|^2_{\mathrm{HS}} - \|\mathcal{P}_U\mathcal{K}\mathcal{P}_V - \mathcal{A}\|^2_{\mathrm{HS}}.$$

Every term is finite for any bounded $\mathcal{K}$: $\mathcal{A}$ has rank at most K, so $\langle \mathcal{A}, \mathcal{K}\rangle_{\mathrm{HS}}$ is the finite sum $\sum_{a,b\le K} \mathsf{A}_{ab}\langle f_a, \mathcal{K}g_b\rangle_{\rho_0}$, and $\mathcal{P}_U\mathcal{K}\mathcal{P}_V$ has rank at most K and so is Hilbert–Schmidt. At $\mathsf{A} = \mathsf{I}$, the $K \times K$ identity matrix, the operator is $\mathcal{A} = \sum_{i\le K} f_i \otimes g_i$, the one the features form before any fit, and the first line of equation 4 is minus equation 2, so dictionary learning maximizes $R_E(\mathsf{I})$ on the pairs it is trained on. For a Hilbert–Schmidt $\mathcal{K}$, expanding the plain objective $\|\mathcal{K} - \mathcal{A}\|^2_{\mathrm{HS}} - \|\mathcal{K}\|^2_{\mathrm{HS}}$ cancels $\|\mathcal{K}\|^2_{\mathrm{HS}}$ and leaves $-2\langle \mathcal{A}, \mathcal{K}\rangle_{\mathrm{HS}} + \|\mathcal{A}\|^2_{\mathrm{HS}} = -R_E(\mathsf{I})$, the computed form equation 2. The identity above at $\mathsf{A} = \mathsf{I}$ then gives equation 3, whose second term is the largest score the two spans allow and whose first is how far the operator the dictionary represents falls from the projection. The identity is for the score over the law the moments are taken on, in the plain form and in the nested one we train; a single minibatch adds a term.

The projection's matrix. The projection $\mathcal{P}_U$ is least-squares regression on the present-state features, so $\mathcal{P}_U \mathcal{K} g_b = \mathbf{f}^\top \mathsf{M}_{\rho_0}[\mathbf{f}]^{-1} \mathbb{E}[\mathbf{f}(\mathbf{x})(\mathcal{K} g_b)(\mathbf{x})]$ for each future-state feature g_b. By the tower-property step above, $\mathbb{E}[\mathbf{f}(\mathbf{x})(\mathcal{K} g_b)(\mathbf{x})]$ is the b-th column of the cross-moment, so the pairs alone give $\mathcal{P}_U \mathcal{K} g_b$, the function in U closest to $g_b(\mathbf{y})$ in mean square. That is the first line below, and $\mathcal{P}_V h = \mathbf{g}^\top \mathsf{M}_{\rho_1}[\mathbf{g}]^{-1} \mathbb{E}[\mathbf{g}(\mathbf{y}) h(\mathbf{y})]$, the same regression on the future-state features applied before $\mathcal{K}$, contributes $\mathsf{M}_{\rho_1}[\mathbf{g}]^{-1}$ in the second:

$$\begin{aligned} \mathcal{P}_U \mathcal{K} g_b &= \sum\nolimits_{a \le K} \left(\mathsf{M}_{\rho_0}[\mathbf{f}]^{-1} \mathsf{T}[\mathbf{f}, \mathbf{g}] \right)_{ab} f_a, \\ \mathcal{P}_U \mathcal{K} \mathcal{P}_V &= \sum\nolimits_{a,b \le K} \left(\mathsf{M}_{\rho_0}[\mathbf{f}]^{-1} \mathsf{T}[\mathbf{f}, \mathbf{g}]\, \mathsf{M}_{\rho_1}[\mathbf{g}]^{-1} \right)_{ab} f_a \otimes g_b, \\ \bar{\mathsf{C}}_{01} \triangleq \mathsf{M}_{\rho_0}[\mathbf{f}]^{-1/2} \mathsf{T}[\mathbf{f}, \mathbf{g}]\, \mathsf{M}_{\rho_1}[\mathbf{g}]^{-1/2} &= \sum\nolimits_{i \le K} \widehat{\sigma}_i\, \mathbf{u}_i \mathbf{v}_i^\top. \end{aligned} \tag{9}$$

Orthonormalizing the features to $\mathsf{M}_{\rho_0}[\mathbf{f}]^{-1/2}\mathbf{f}$ and $\mathsf{M}_{\rho_1}[\mathbf{g}]^{-1/2}\mathbf{g}$, the whitening, turns the coefficient matrix of the second line into the whitened cross-covariance $\bar{\mathsf{C}}_{01}$ of the third. The singular values $\widehat{\sigma}_i$ of $\bar{\mathsf{C}}_{01}$ are the uncentered canonical correlations of $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{y})$, and $\mathbf{u}_i$ and $\mathbf{v}_i$ in $\mathbb{R}^K$ are its present-state and future-state singular vectors. Putting the second line's matrix into the trace form of $\|\mathcal{A}\|_{\mathrm{HS}}^2$ above gives $\|\mathcal{P}_U \mathcal{K} \mathcal{P}_V\|_{\mathrm{HS}}^2 = \mathrm{Tr}\left(\mathsf{M}_{\rho_1}[\mathbf{g}]^{-1} \mathsf{T}[\mathbf{f}, \mathbf{g}]^\top \mathsf{M}_{\rho_0}[\mathbf{f}]^{-1} \mathsf{T}[\mathbf{f}, \mathbf{g}] \right) = \|\bar{\mathsf{C}}_{01}\|_F^2 = \sum_i \widehat{\sigma}_i^2$, the VAMP-2 of the spans. The functions $\mathbf{u}_i^\top \mathsf{M}_{\rho_0}[\mathbf{f}]^{-1/2}\mathbf{f}$ and $\mathbf{v}_i^\top \mathsf{M}_{\rho_1}[\mathbf{g}]^{-1/2}\mathbf{g}$ are orthonormal, and with the $\widehat{\sigma}_i$ they write the projection in the form of equation 1, so they estimate ϕ_i and ψ_i. If the two spans hold the leading singular subspaces, equation 9 recovers the leading K terms of equation 1; otherwise it is $\mathcal{K}$ compressed to the dictionary.

C.3 Where the Hilbert–Schmidt condition is needed

The computed form of equation 2 is a polynomial in the moment matrices, and the identity equation 3 holds for any bounded operator (Appendix C.2), so neither needs a Hilbert–Schmidt assumption. The form that does is the difference $\|\mathcal{K} - \sum_{i \le K} f_i \otimes g_i\|_{\mathrm{HS}}^2 - \|\mathcal{K}\|_{\mathrm{HS}}^2$, which is finite only when $\mathcal{K}$ is Hilbert–Schmidt, that is when the pair law has a square-integrable density against the product of its marginals. We do not verify that condition for either system; in the atmosphere at the longest lag the core's relative forecast error u exceeds one. The core's operator is a composition whose singular values all equal one when it is read from the law of $\Phi_{\mathrm{core}}\mathbf{x}$ to ρ_0, so it is not Hilbert–Schmidt: against the core's target the difference form is formal, while equation 3 holds as written. Unlike the VAMP-2 score of Mardt et al. (2018) and the projection score of Kostic et al. (2024b), which place inverse square roots inside the loss, it holds no inverse, so minibatch training does not backpropagate through an ill-conditioned one. When the law of $\Phi_{\mathrm{core}}\mathbf{x}$ has a bounded density with respect to ρ_1, $\mathcal{K}^{\mathrm{core}}$ maps a function to one with the same norm under that law, whereas $\mathcal{K}$ contracts every direction that the present state does not determine exactly.

The objective targets singular functions rather than eigenfunctions because the transfer operator of an irreversible system is not self-adjoint, and only its singular functions are, in order, the most predictable functions (Section 2.1). The span of the minimizers is unique only when the spectrum has a gap at the number of features, $\sigma_K > \sigma_{K+1}$, and the nested form orders the pairs only when the singular values are distinct. When $\mathcal{K}$ is Hilbert–Schmidt, the minimizers of equation 2 span its leading K singular subspaces by the Eckart–Young–Mirsky theorem (Eckart & Young, 1936), and a nested form of the objective orders them by singular value; this is a property of the objective for the operator it is written against, not a condition on training. The objective sees the two sides only through their product, so $\mathbf{f}$ and $\mathbf{g}$ are fixed only up to an invertible $K \times K$ map. What a finite record of transition noise leaves in the sample moments we call sampling noise.

C.4 The fit's three error terms

When $\mathcal{K}$ is Hilbert–Schmidt, the error of the fitted operator splits orthogonally into two approximation terms and the estimation error,

$$\|\mathcal{K} - \widehat{\mathcal{K}}\|_{\mathrm{HS}}^2 = \underbrace{\|\mathcal{K}(\mathcal{I} - \mathcal{P}_V)\|_{\mathrm{HS}}^2}_{\text{output side}} + \underbrace{\|(\mathcal{I} - \mathcal{P}_U)\mathcal{K}\mathcal{P}_V\|_{\mathrm{HS}}^2}_{\text{input side}} + \underbrace{\|\mathcal{P}_U \mathcal{K} \mathcal{P}_V - \widehat{\mathcal{K}}\|_{\mathrm{HS}}^2}_{\text{estimation error}}. \tag{10}$$

The output-side term is the part of $\mathcal{K}$ on future-state functions outside V, and the input-side term its part on V that lands outside U. The fitted operator satisfies $\widehat{\mathcal{K}} = \mathcal{P}_U \widehat{\mathcal{K}} \mathcal{P}_V$. For any Hilbert–Schmidt

A, $\|A\|_{\mathrm{HS}}^2 = \|A\mathcal{P}_V\|_{\mathrm{HS}}^2 + \|A(\mathcal{I} - \mathcal{P}_V)\|_{\mathrm{HS}}^2$, since the two pieces act on orthogonal subspaces. With $A = \mathcal{K} - \widehat{\mathcal{K}}$, the second piece is $\mathcal{K}(\mathcal{I} - \mathcal{P}_V)$ because $\widehat{\mathcal{K}}$ vanishes off V. The first piece is $(\mathcal{I} - \mathcal{P}_U)\mathcal{K}\mathcal{P}_V + (\mathcal{P}_U\mathcal{K}\mathcal{P}_V - \widehat{\mathcal{K}})$, whose two summands take values in the orthogonal complement of U and in U, so their Hilbert–Schmidt inner product is zero. Together these give equation 10.

At $\widehat{\mathcal{K}} = 0$, equation 10 writes the VAMP-2 of the spans, $\|\mathcal{P}_U\mathcal{K}\mathcal{P}_V\|_{\mathrm{HS}}^2$, as $\|\mathcal{K}\|_{\mathrm{HS}}^2$, the summed predictability of all the system's singular functions, less the two approximation terms. By equation 4 under the population law, a fitted operator therefore scores $\|\mathcal{K}\|_{\mathrm{HS}}^2 - \|\mathcal{K} - \widehat{\mathcal{K}}\|_{\mathrm{HS}}^2$. The approximation terms depend on the spans alone, so they are what the spans miss and the only terms a further dictionary lowers. The estimation error is what the fit misses, and it is the only term a fit over the dictionary lowers. Since $\|\mathcal{K}\|_{\mathrm{HS}}^2$ is unknown, the approximation terms are identified only as a difference between two dictionaries (Appendix C.5). Section 3.2 uses the input-side term, which under forecast containment keeps only the core's error outside U.

C.5 The three terms from finite samples

The three terms of equation 10 are not equally available from a finite record. One is read directly, and the other two are identified as a difference between dictionaries.

The estimation error. On a block of held-out pairs, the largest score the two spans allow is the sum of the squared canonical correlations of $\mathbf{f}(\mathbf{x})$ with $\mathbf{g}(\mathbf{y})$ on those rows, and it is the supremum of the held-out VAMP-E over operators with present-state functions in U that vanish off V. The held-out score of a fitted operator lies below it, and the difference, the estimation error of Section 2.3, is $\|\mathcal{P}_U\mathcal{K}\mathcal{P}_V - \widehat{\mathcal{K}}\|_{\mathrm{HS}}^2$ under the empirical law of those rows. That is an identity on the rows it is read on, so the term needs no assumption beyond the two spans and the block; as an estimate of the population term it holds the bias this appendix describes below.

The approximation terms. Their sum is $\|\mathcal{K}\|_{\mathrm{HS}}^2$ minus the VAMP-2, and the first quantity is unknown and can be infinite, so neither term nor their sum is estimable for one dictionary. What is identified is a difference between two dictionaries read on the same block: the difference of their approximation sums is minus the difference of their VAMP-2 scores, and every approximation statement in this paper has that form. Splitting a difference into its input and output parts needs a dictionary with wider present-state features, whose VAMP-2 bounds $\|\mathcal{K}\mathcal{P}_V\|_{\mathrm{HS}}^2$ from below. We do not measure that, so nothing here attributes an approximation difference to one side.

The four blocks. The split in equation 10 comes from four mutually orthogonal blocks of the operator: the block the fit reaches, the block only U misses, the block only V misses, and the corner both miss. Grouping them into two approximation terms gives the corner to whichever side is divided off first, so an input-side and an output-side term are not properties of the operator alone; the four blocks are the statement that does not depend on the order. The split of equation 10 divides V off first.

The finite-sample bias. A VAMP-2 from finitely many rows exceeds its population value. With independent rows and no dependence between present and future it is near pq/n, for p and q the widths of the two sides and n the number of rows; serial correlation replaces n by a smaller effective count, which we do not measure for the VAMP-2 scores. Appendix B.2 gives the settings of the correction we apply to it, a jackknife, which under-corrects, in each dictionary, components slower than the unit it deletes. Two dictionaries of the same number of features on the same block therefore have biases of the same order, and a VAMP-2 difference cancels them to leading order rather than exactly. The margin does not rest on that correction, since the same bias enters a VAMP-2 and the estimation error beneath it and cancels in their difference. The fit's estimation error holds that bias and, in addition, the shift between the law the operator was fitted on and the law it is read on: features decorrelate between the two blocks, and the score this loses appears in the estimation error and not in the approximation terms.

C.6 How transition reuse inflates the gap

Regressors chosen on the same noisy sample that then estimates their coefficients are selected in part for fitting that sample's noise, so they score well in sample and worse on the held-out block. Regressors chosen without the observed responses, with those responses used only for the coefficients, avoid that selection. A dictionary learned from the data alone is the first case; a dictionary learned from the core is the second, its features coming from the core's transitions out of present states in the data.

For squared error, the expected excess of test over training error is the fit's optimism, twice the sum of covariances between fitted values and responses divided by the number of samples (Efron, 2004), (Hastie et al., 2009, Chapter 7). With features fixed before the responses are seen, the optimism of least squares without shrinkage or truncation counts only its coefficients, K^2 of them for a $K \times K$ fit. With features chosen on the same responses, the optimism also counts the search; that surplus is proven positive for selectors such as best subset, though not in general, and shrinkage can offset it (Tibshirani, 2015). Dictionary learning against the core does not read which future follows which state, so the core-trained dictionary's span part of equation 5 gains no search term from the pairing, and its fit part is the ordinary estimation error of its $K \times K$ coefficients.

Both dictionaries are then read through a rank truncation that keeps the singular functions an operator fitted on pairs in the data ranks highest. For the core-trained dictionary it is therefore the training of the features, and not the truncation, that pairs no state in the data with its future in the data.

Two things the accounting does not give. The identity is exact for squared error with fixed covariates and an independent test copy, and VAMP-E is a squared error on the whitened cross-covariance only once the whitening is fixed, so it applies to our score approximately. And it does not give the size of the core-trained dictionary's own generalization gap.

C.7 Proof of Proposition 1

Write $h = \mathbf{c}^\top \mathbf{g}$ for a coefficient vector $\mathbf{c}$. Then $(\mathcal{K}^{\text{core}} h)(\mathbf{x}) = h(\Phi_{\text{core}}\mathbf{x}) = \mathbf{c}^\top \mathbf{g}(\Phi_{\text{core}}\mathbf{x})$, a linear combination of the functions spanning S, so $\mathcal{K}^{\text{core}} h \in S \subseteq U$. Hence $\mathcal{P}_U \mathcal{K}^{\text{core}} h = \mathcal{K}^{\text{core}} h$ and $(\mathcal{I} - \mathcal{P}_U)\mathcal{K}^{\text{core}} h = 0$. Applying $\mathcal{P}_U$ and $\mathcal{I} - \mathcal{P}_U$ to $\mathcal{K}h = \mathcal{K}^{\text{core}} h + (\mathcal{K} - \mathcal{K}^{\text{core}})h$ and using linearity gives equation 7. Summing the second equality's squared norm over an orthonormal basis of V gives the input-side term of Section 3.2. Its bound holds because $\mathcal{I} - \mathcal{P}_U$ is an orthogonal projection, and its right side is the core's squared error summed over that basis. The least-squares fit of $h(\mathbf{y})$ on the functions spanning U attains $\mathcal{P}_U \mathcal{K}h$: its normal equations set the residual orthogonal to U, and $h(\mathbf{y}) - (\mathcal{K}h)(\mathbf{x})$ is already orthogonal to every function of $\mathbf{x}$, so the fitted function is the projection of $\mathcal{K}h$ onto U. Since $S \subseteq U$, the fitted function contains $\mathcal{P}_S(\mathcal{K} - \mathcal{K}^{\text{core}})h$, the part of the core's error that is linear in $\mathbf{g}_{\text{core}}(\Phi_{\text{core}}\mathbf{x})$.

Containment in the trained networks. Forecast containment equation 6, the proposition's hypothesis, holds less closely at three days than at the lags Section 3.3 reports: there the least-squares map from $\mathbf{f}_{\text{core}}(\mathbf{x})$, with an intercept, reproduces $\mathbf{g}_{\text{core}}(\Phi_{\text{core}}\mathbf{x})$ with held-out R^2 between 0.895 and 0.942, read on the core-trained dictionaries that the combined dictionary of Appendix H is built on. At seven, ten and fourteen days the pooled held-out R^2 is 0.963 to 0.974, against -0.0045 to -0.0037 for as many i.i.d. standard normal features, and the median feature's is 0.50 to 0.70, since the pooled share is weighted toward the high-variance features. At those lags 47 to 57 of the 200 canonical correlations between the held-out spans exceed 0.9.

C.8 Flat objective against the core

Against the core the population objective fixes no order of the per-pair scales, so the order a trained core-trained dictionary shows comes from its encoder and its training on the states in the data, and it follows each feature's persistence along the data. The fitted operator does not use that order: it is fitted over all K features and orders its own singular functions by their singular values, equation 9.

The per-pair scale. The scale the training solves for a future-state feature, read over the whole training block, is the cosine over the present states in the data between the present-state feature paired with it and the core's forecast of that feature, less a deflation by the earlier pairs. That

cosine is uncentered, like the moments the objective is written with, and the expression is exact for orthonormal features. The scale the training solves from its moving average of batch moments differs from this whole-block one by a per-batch term, and the numbers here are the whole-block scales in magnitude. Each scale is an achieved value of this encoder and this optimization, neither a singular value of $\mathcal{K}^{\mathrm{core}}$ nor a verified optimum over the encoder class.

Unlimited capacity. If the present-state encoder can represent $\mathbf{g} \circ \Phi_{\mathrm{core}}$ for some future-state features $\mathbf{g}$ orthonormal under the law of $\Phi_{\mathrm{core}}\mathbf{x}$, then $\mathbf{f} = \mathbf{g} \circ \Phi_{\mathrm{core}}$ makes the cross-moment and both second moments the identity, so every cosine is one, no deflation remains, and equation 2 takes the value $-K$. For any $\mathbf{g}$ with $\mathsf{M}_{\rho_0}[\mathbf{g} \circ \Phi_{\mathrm{core}}]$ invertible, equation 2 is minimized over $\mathbf{f}$ pointwise by $\mathbf{f} = \mathsf{M}_{\rho_0}[\mathbf{g} \circ \Phi_{\mathrm{core}}]^{-1}\, \mathbf{g} \circ \Phi_{\mathrm{core}}$, where it equals $-K$, and a singular second moment gives at best minus its rank. So $-K$ is the minimum, every such $\mathbf{g}$ attains it and none is preferred, which is the flatness of Section 3.3. With $\mathbf{g}$ orthonormal, $\mathbf{f} = \mathbf{g} \circ \Phi_{\mathrm{core}}$ minimizes the objective of every prefix at once, so the nested form is flat as well; the argument uses the moments alone and needs no Hilbert–Schmidt assumption (Appendix C.3).

Measured order. Every number in this paragraph is taken over the training block, on the atmospheric dictionaries at seven and fourteen days with three seeds each, and a feature's persistence is taken along the trajectory in the data at that dictionary's own lag. For the trained core-trained dictionaries the deflation is under one percent of the cosine in median over features 2 to 60, and at most six percent for any one of them. At the trained capacity the scales are not all one, and their ranking is what is left: the first scale's size exceeds the mean size of the 51st to 60th by 0.065 to 0.096. The scales fall with the nested index, and their order agrees with how persistent each feature is along the data, its correlation with itself one lag later, $\mathrm{corr}(f_k(\mathbf{x}_t), f_k(\mathbf{x}_{t+\tau}))$. The rank correlation between the two, over features 2 to 60 with feature 1 excluded, is $+0.90$ to $+0.95$ at seven days and $+0.68$ to $+0.71$ at fourteen, against permutation-null 95th percentiles of 0.21 to 0.23 (Table 2), and the leading features are slower than random directions of the same span. The same ordering appears in the data-trained dictionary, whose objective reads the data's futures rather than the core's, so persistence orders both dictionaries and not only the core's. These measurements characterize the ordering the specified encoder and training procedure produce; they do not identify what aspect of the parametrization or the optimization selects it.

Stochastic core. A stochastic core would make $\mathcal{K}^{\mathrm{core}}$ a contraction with a nontrivial spectrum, so dictionary learning would have a ranking by the model's own predictability, and a narrow core-trained dictionary would then be expected to lose less. That ranking would carry to the observed system only if the two operators were close on the leading singular functions, and a single sampled future per state would put model noise back into the objective that trains the span unless the conditional expectation is estimated from an ensemble. We leave the choice of model stochasticity to future work.

C.9 Operator rank changes only the fit

A held-out score is the operator's own norm on that block minus the risk of the fitted operator, so reading a curve of held-out scores is reading the risk of equation 10 upside down. Every dictionary here is trained once at $K = 200$ and read at rank k by truncating the fitted operator, so U and V do not move along the curve and both approximation terms are the same number at every rank. The truncation therefore falls within the estimation error, since the split assumes only that the fitted operator lies in the two spans, and a fit over the dictionary can lower that term alone (Appendix C.4). Both approximation terms set each curve's level and neither can move a peak; whatever separates the two peaks is in the estimation error, where a falling truncation bias meets a rising sampling error. The two dictionaries differ there by construction rather than by degree: the data-trained dictionary's features were chosen on the same pairs the fit then uses, while the core-trained dictionary's were trained on the core's transitions and pair no state in the data with its future in the data.

The core shapes the features while they are learned, and the fitted operator then reads only the state in the data, so the core is a training-time cost and not part of the inference path.

D THE MARGIN UNDER EACH PROTOCOL

The body reports the margin at one fixed protocol, rank 60 and shrinkage 0.01. This appendix states how the two keys are applied and gives the margin at that protocol seed by seed, and again with each dictionary choosing its own shrinkage, or its rank and shrinkage together.

D.1 SHRINKAGE AND RANK

Every fitted operator in this paper is scored under two keys, the same for every dictionary, lag and system and applied to a trained dictionary without retraining it: the rank at which the operator is truncated, and the shrinkage added to the second moments before they are inverted. Each feature is first divided by its training root-mean-square, and a shrinkage λ times the identity is added to each second-moment matrix before whitening, so that λ is relative to the mean eigenvalue. A rank of k means the operator fitted over all 200 features and truncated to its leading k singular functions, not a dictionary trained with fewer than 200 features. It keeps the leading k terms of the decomposition equation 9, so its coefficient matrix is $\mathsf{K}_{\mathrm{fit}} = \mathsf{M}_{\rho_0}[\mathbf{f}]^{-1/2}\big(\sum_{i\le k}\widehat{\sigma}_i \mathbf{u}_i\mathbf{v}_i^{\top}\big)\mathsf{M}_{\rho_1}[\mathbf{g}]^{-1/2}$, with every moment, and so $\bar{\mathsf{C}}_{01}$, taken on the training block after the shrinkage is added.

Without shrinkage and at full rank, $\mathsf{K}_{\mathrm{fit}}$ is the matrix of $\mathcal{P}_U\mathcal{K}\mathcal{P}_V$ on the training block, so there the estimation error is zero. The fit then scores the training VAMP-2, the sum $\sum_i \widehat{\sigma}_i^2$ of the predictabilities the dictionary claims, with $\widehat{\sigma}_1^2 = 1$ when both spans contain the constant. The fitted operator $\widehat{\mathcal{K}} = \sum_{a,b\le K}(\mathsf{K}_{\mathrm{fit}})_{ab}\, f_a \otimes g_b$ acts only on V and maps into U.

The whitening and the shrinkage act on the second moments of all 200 features at every rank, so the rank axis varies how many singular functions are kept and not what the fit costs to estimate. Both dictionaries are compared on this one rank axis, which favors the core-trained dictionary at small ranks; every dictionary here is read at its trained number of features, 200.

Figure 6 varies the shrinkage with the rank held at 60, and Figure 7 varies the rank at seven and fourteen days. At the fixed protocol the data-trained dictionary's held-out score peaks at rank 80 at three days, 40 to 60 at five, 40 at seven, 20 to 40 at ten and 20 at fourteen, and at one day neither dictionary's turns down before rank 200. Figures 1 and 7 show the whole rank axis. With each dictionary choosing its own rank and shrinkage on validation rows the core-trained dictionary stays ahead from seven to fourteen days in every seed (Appendix D.3.3). With the choice made on one half of the held-out block, it stays ahead at ten and fourteen days and falls behind at five and seven (Appendix D.3.2). The margin falls as the shrinkage grows, and at the long lags the core-trained dictionary is ahead at every value drawn.

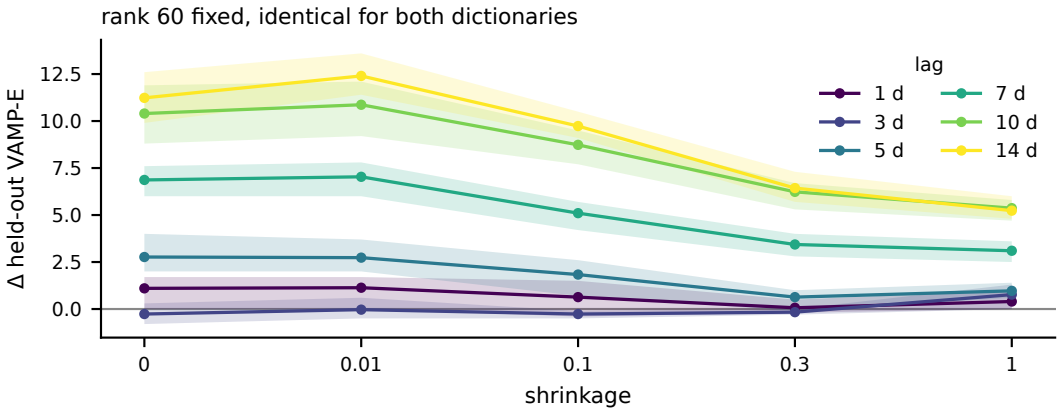

Figure 6: The margin falls as the shrinkage grows, and at seven, ten and fourteen days the core-trained dictionary is ahead at every value drawn. Held-out margin against shrinkage at rank 60, on the whole held-out block, one line per lag with the band spanning the three seeds; nothing is chosen on the rows being scored for either dictionary, and the column at 0.01 is the protocol Table 3 gives.

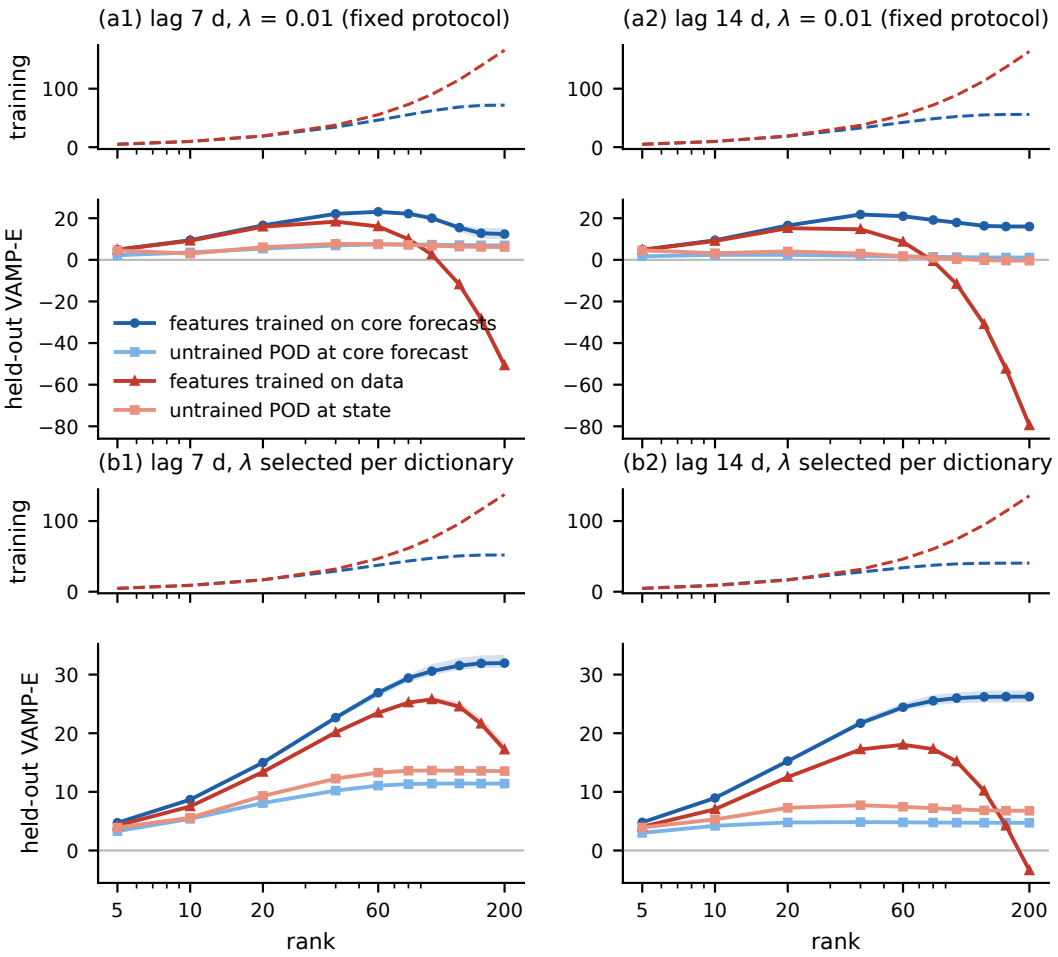

Figure 7: The data-trained dictionary claims more with rank and keeps less. VAMP-E against rank at seven and fourteen days, each held-out panel under its own training strip and both untrained bases drawn; row (a) is at shrinkage 0.01 and row (b) at each dictionary's two-fold shrinkage, held at its rank-60 selection along the whole curve.

D.2 Margins at the fixed protocol

Table 3 lists the configurations behind the margin's growth with the lag (Section 4.3): the held-out margin of the core-trained dictionary over the data-trained dictionary at the fixed protocol, seed by seed.

Table 3: In the seed mean the core-trained dictionary's held-out margin over the data-trained dictionary grows with the lag from three to fourteen days, and at one and three days it is not separated from zero. Rank 60, shrinkage 0.01, seeds 0, 1 and 2 separately, with basic 90% intervals from the paired bootstrap of Appendix B.2.

lag	margin, seeds 0 / 1 / 2	basic 90% intervals excluding 0
1 d	+1.76 / +0.05 / +1.70	0 of 3
3 d	−0.44 / −0.21 / +0.61	0 of 3
5 d	+2.07 / +3.72 / +2.51	2 of 3
7 d	+5.95 / +7.36 / +7.76	3 of 3
10 d	+9.22 / +12.12 / +11.24	3 of 3
14 d	+12.15 / +11.34 / +13.58	3 of 3

D.3 Letting each dictionary choose its settings

Each dictionary chooses for itself here, and the three subsections below differ in what is chosen and in which rows the choice reads.

D.3.1 The selected protocol

The body scores every configuration at one fixed weak shrinkage, because the claim is about the dictionaries' spans, which a weak shrinkage leaves alone and a strong one does not. This appendix gives the same configurations under the other protocol: the shrinkage is chosen per dictionary by a two-fold selection over the two halves of the held-out block, over the full grid, and the operator is then read on the whole held-out block. Selecting on the block a score is then read on biases that score upward, and the bias falls on whichever dictionary the selection helps more, so this protocol is a check on the fixed one rather than a replacement for it.

Table 4: Held-out VAMP-E of the core-trained dictionary minus the data-trained dictionary at rank 60, each dictionary at the shrinkage its own two-fold selection picks over the full grid. Both interval kinds come from the same paired moving-block bootstrap.

lag	seed	margin	basic 90%	percentile 90%
1 d	0	$+4.54$	$[+5.43, +8.63]$	$[+0.45, +3.65]$
	1	-0.10	$[-0.39, +1.19]$	$[-1.38, +0.20]$
	2	$+3.72$	$[+4.33, +7.28]$	$[+0.16, +3.11]$
3 d	0	-0.26	$[-1.06, +3.88]$	$[-4.39, +0.55]$
	1	-0.00	$[-0.41, +3.63]$	$[-3.63, +0.41]$
	2	-0.31	$[-1.14, +1.10]$	$[-1.73, +0.51]$
5 d	0	-0.09	$[-1.21, +2.61]$	$[-2.79, +1.02]$
	1	$+0.99$	$[+0.19, +4.15]$	$[-2.17, +1.79]$
	2	$+1.05$	$[+0.22, +3.46]$	$[-1.36, +1.88]$
7 d	0	$+2.76$	$[+1.72, +4.97]$	$[+0.55, +3.80]$
	1	$+3.52$	$[+2.97, +6.10]$	$[+0.94, +4.06]$
	2	$+4.03$	$[+3.33, +6.71]$	$[+1.34, +4.72]$
10 d	0	$+5.39$	$[+4.77, +8.09]$	$[+2.69, +6.01]$
	1	$+6.68$	$[+6.54, +9.98]$	$[+3.37, +6.81]$
	2	$+6.65$	$[+6.62, +10.07]$	$[+3.22, +6.67]$
14 d	0	$+6.26$	$[+5.07, +9.53]$	$[+2.98, +7.45]$
	1	$+5.65$	$[+4.71, +8.24]$	$[+3.06, +6.60]$
	2	$+7.32$	$[+6.38, +10.63]$	$[+4.00, +8.25]$

Benchmark. Table 5 scores the benchmark's configurations of Table 8 the same way, each dictionary at the shrinkage its own two-fold selection picks. The core-trained dictionary keeps the ordering the fixed protocol gives it in all but one of these configurations, and that one is a one-day configuration at the larger data size, -0.07 at the selected protocol, whose basic interval just excludes zero and whose percentile interval does not. What the margins lose is size: at five and seven days they run about half the fixed protocol's, and at one and three days they stay closer to it. The reason is the data-trained dictionary rather than the core. In the configurations where it falls furthest at shrinkage 0.01 the two-fold selection picks 0.3 or 1 for it, and that shrinkage returns much of what it had lost, which is where the two protocols part. In five of these nineteen configurations a basic interval excludes its own point estimate, marked below, so a single configuration's basic interval is not a reliable statement about that configuration and the sign that repeats across configurations is what we read.

D.3.2 Held-out-half selection of rank and shrinkage

Appendix D.3.1 lets each dictionary choose its shrinkage and holds the rank at 60. Here each chooses both, on one half of the held-out block and scored on the other, so nothing is chosen on the rows a score is read on. The two selections here differ in which block the choice reads and in which encoders they choose for. This one uses halves of the held-out block, for the encoders of Table 3, trained on the whole training window; the one below uses validation rows inside the training

Table 5: Shallow water, seed 0: the configurations of Table 8 with each dictionary at the shrinkage its own two-fold selection picks, read at rank 60 on the whole held-out block. The margin is the accurate core's dictionary minus the data-trained dictionary, and † marks a configuration whose basic interval excludes it, which Appendix B.2 takes up.

lag	r	N	shrinkage, core / data	margin	basic 90%	percentile 90%
1 d	0.02	256	0.1 / 0.1	$+0.140^{\dagger}$	$[-0.995, +0.022]$	$[+0.259, +1.275]$
	0.02	1024	0.1 / 0.1	-0.067	$[-1.289, -0.011]$	$[-0.122, +1.155]$
	0.3	256	0.1 / 0.1	$+1.404$	$[+0.418, +1.684]$	$[+1.123, +2.389]$
	0.3	1024	0.1 / 0.1	$+0.502$	$[-1.596, +0.623]$	$[+0.380, +2.599]$
3 d	0.02	256	0.1 / 0.1	$+1.503$	$[+0.500, +1.640]$	$[+1.366, +2.505]$
	0.02	1024	0.1 / 0.1	-0.335	$[-1.399, +0.496]$	$[-1.166, +0.730]$
	0.3	256	0.1 / 0.1	$+4.400$	$[+3.380, +4.523]$	$[+4.277, +5.420]$
	0.3	1024	0.1 / 0.1	-0.252	$[-2.051, -0.045]$	$[-0.458, +1.547]$
	1.0	256	0.1 / 0.3	$+7.318$	$[+7.252, +8.276]$	$[+6.360, +7.384]$
5 d	0.02	256	0.1 / 0.3	$+3.973^{\dagger}$	$[+4.195, +5.903]$	$[+2.042, +3.750]$
	0.02	1024	0.1 / 0.1	-1.225	$[-1.889, -0.982]$	$[-1.468, -0.561]$
	0.3	256	0.1 / 0.3	$+8.660$	$[+8.343, +9.537]$	$[+7.784, +8.977]$
	0.3	1024	0.1 / 0.1	$+0.790$	$[-0.748, +0.980]$	$[+0.601, +2.328]$
	1.0	256	0.1 / 0.3	$+10.368$	$[+10.069, +10.965]$	$[+9.771, +10.667]$
7 d	0.02	256	0.3 / 0.3	$+4.947$	$[+3.497, +5.432]$	$[+4.462, +6.396]$
	0.02	1024	0.1 / 0.1	$+1.948^{\dagger}$	$[+0.243, +1.805]$	$[+2.090, +3.653]$
	0.3	256	0.1 / 1	$+8.376^{\dagger}$	$[+8.425, +10.014]$	$[+6.738, +8.327]$
	0.3	1024	0.1 / 0.3	$+3.708$	$[+3.293, +4.361]$	$[+3.054, +4.122]$
	1.0	256	0.1 / 1	$+9.073^{\dagger}$	$[+9.073, +10.025]$	$[+8.120, +9.073]$

period, with encoders early-stopped on those rows. Table 6 gives the atmosphere, configuration by configuration.

The core-trained dictionary stays ahead at ten and fourteen days in every seed and falls behind at five and seven. The seed means of the margin at each dictionary's own pick are -5.15 and -1.99 at five and seven days, against $+4.62$ and $+5.44$ at ten and fourteen. Three protocols of the same configurations stand beside each other: the margin at each dictionary's own pick, the margin on the whole block, which selection biases upward, and the margin at rank 60 with each dictionary's own leak-free shrinkage.

With the rank fixed at 60 the margin is positive from seven days in every seed, so it is letting the rank be chosen that turns five and seven days toward the data-trained dictionary. That turn comes from the data-trained dictionary's gain from the free rank: its two-fold score rises by 6.98 to 7.95 at five days, more than the turn, and by 3.35 to 4.10 at seven. The gain does not need the grid's corner: read on the whole block at seven days, its rank-100 score at shrinkage 0.3 is 26.15, 25.34 and 25.81 over the three seeds, about what rank 200 at shrinkage 1 gives, while rank 200 at 0.3 falls to 18.29, 17.04 and 16.28. At five and seven days the core-trained dictionary's own two-fold score changes by -0.23 to $+4.14$, except at seven days and seed 1, where it falls by 3.05, about half the turn there.

Four things in the table bound what it says. The data-trained dictionary picks the grid's corner, rank 200 at shrinkage 1, on both halves at five and seven days in every seed. A wider grid could therefore only raise each half's own best, and the two-fold score only where the halves' picks still agree; the core-trained dictionary picks rank 200 on half A in 11 of the 12 configurations. Where the halves' picks differ, the two-fold scores a half at a pick that half would not make, and that happens to the core-trained dictionary in all twelve configurations and to the data-trained dictionary in three. The core-trained dictionary scores 4.1 to 9.7 higher on half A than on half B at its own picks. Intervals are not read here, since the replicates sit far below their points and the basic interval excludes its own point estimate in 9 of the 12 configurations. The evidence is the sign of the point margin, which agrees across the three seeds at every lag.

In the benchmark the same selection keeps the core-trained dictionary ahead. Thirteen configurations are read there, the accurate core at seven, ten and fourteen days and r of 0.02, 0.3 and 1.0, with 256 trajectories. The core-trained dictionary is ahead at its own rank and shrinkage in all thirteen,

Table 6: Each dictionary chooses its rank and shrinkage on one half of the held-out block and is scored on the other. A pick is written rank/shrinkage, A and B are the two halves, and a two-fold score is the mean of each half scored at the other half's pick. The margins are the core-trained dictionary's score minus the data-trained dictionary's: at each dictionary's own pick, on the whole held-out block, which selection biases upward, and at rank 60 with each dictionary's own leak-free shrinkage. Intervals are not given here, for the reason the text beneath states.

	core picks		core	data picks		data	margin		
configuration	A	B	two-fold	A	B	two-fold	own	whole	rank 60
5 d, s0	200/0.3	80/0.3	23.58	200/1	200/1	30.39	−6.81	+2.14	−0.49
5 d, s1	200/0.3	80/0.3	24.11	200/1	200/1	29.67	−5.56	+3.15	+0.66
5 d, s2	200/0.3	160/0.3	27.07	200/1	200/1	30.16	−3.09	+3.13	+0.72
7 d, s0	200/0.3	80/0.3	22.21	200/1	200/1	23.50	−1.28	+2.64	+2.69
7 d, s1	200/0.3	80/0.3	19.12	200/1	200/1	23.04	−3.92	+3.95	+2.47
7 d, s2	200/0.3	100/0.3	22.20	200/1	200/1	22.98	−0.77	+6.31	+2.84
10 d, s0	200/0.3	80/0.3	21.95	200/1	200/1	17.54	+4.41	+8.88	+4.85
10 d, s1	200/0.3	80/0.3	21.00	60/0.3	160/1	15.78	+5.22	+9.92	+5.64
10 d, s2	200/0.3	60/0.3	20.95	60/0.3	200/1	16.70	+4.24	+10.92	+5.44
14 d, s0	160/0.3	60/0.3	20.07	60/0.3	40/0.3	14.55	+5.53	+8.74	+6.03
14 d, s1	200/0.3	60/0.3	19.66	40/0.3	40/0.3	15.31	+4.35	+7.36	+5.59
14 d, s2	200/0.3	80/0.3	21.51	40/0.3	40/0.3	15.07	+6.44	+9.37	+7.03

by $+4.12$ to $+9.42$, every basic interval above zero, six of them excluding their own point estimate by less than 0.5.

D.3.3 Choosing on validation rows

The selection above still reads the held-out block, on one half at a time. Here each dictionary chooses its rank and shrinkage on its own training-period validation rows instead, the last 4,818 of the training window behind the embargo, and the held-out block is read once and whole. At seven days both dictionaries are the early-stopped pair of Appendix E.2.2 rather than the pair of Table 3, so neither encoder trained on the rows its choice is made on. The core-trained dictionary is ahead in every seed, by $+10.80$, $+7.91$ and $+12.19$, a seed mean of $+10.30$. At ten and fourteen days only the data-trained dictionary is early-stopped, and the core-trained dictionary is the one of Table 3, whose encoder trained on those states and never on their futures in the data. It is ahead in every seed at both lags, by $+10.96$ to $+12.76$ at ten days and $+8.57$ to $+11.05$ at fourteen.

Three things bound that conclusion. The early-stopped data-trained dictionary scores lower on the held-out block than the one of Table 3 at the same pick: at seven days, at rank 200 and shrinkage 1, 18.93, 19.37 and 18.42 against 26.05, 25.57 and 25.45. A block of 4,818 rows favors heavy shrinkage, so a pick made on it need not be the grid's best. At seven days the data-trained dictionary at rank 200 and shrinkage 0.3 reads 2.50 to 3.00 above its validation pick on the whole held-out block, less than the margin. The basic interval again excludes its own point estimate in all nine configurations, so the evidence is the sign in every seed. The percentile interval lies above zero in seven of the nine; the two that cover zero are at seven days, in seeds 0 and 1. With the picks re-made on resampled validation rows the core-trained dictionary is ahead in 95.2 to 100 percent of replicates.

Early stopping. Stopped early, between 12,000 and 14,000 steps, the data-trained dictionary trails the core-trained dictionary at the selected protocol by $+8.57$ to $+10.40$ at ten days and $+7.72$ to $+9.34$ at fourteen, with both intervals excluding zero in all six configurations. The core-trained dictionaries are not early-stopped, and the margin does not grow from ten to fourteen days here. At seven days, the thirty-three-year configurations of Appendix E.2.2 stop both dictionaries early on their own rows, and there the core-trained dictionary is ahead in every seed at the fixed protocol, $+4.35$, $+4.41$ and $+5.65$.

E ROBUSTNESS OF THE ATMOSPHERIC MARGIN

Each subsection here checks one way the atmospheric margin could be an artifact: of the encoder, the training budget, the resolution, the amount of data, the annual cycle, the held-out period, or spans that differ from seed to seed.

E.1 ENCODER

E.1.1 A FULL-STATE MLP

To check that neither the overfitting nor the margin belongs to the per-wavenumber encoder alone, both dictionaries are also trained with a full-state MLP, fully connected over the whole state, of width 256 in place of the per-wavenumber encoder's 512, at the same 120,000 steps and the same batch. Every configuration is scored at the body's fixed protocol, rank 60 and shrinkage 0.01 over the whole held-out block.

The overfitting replicates at one, three, seven, ten and fourteen days, in every seed. At shrinkage 0.01 and rank 60 the data-trained dictionary's training score reaches the rank itself; its held-out score peaks below rank 200, and past that peak the core-trained dictionary's held-out score is above it at every rank. At seven, ten and fourteen days the data-trained dictionary's held-out score peaks at rank 40 or below and is below zero at rank 60 in every seed; at three days it is below zero at rank 60 in two seeds of three. At fourteen days both training scores stop rising past an effective rank, near 60 for the core-trained dictionary and near rank 100 for the data-trained dictionary. At rank 200 the held-out scores are 7.9 to 12.7 and -76.1 to -34.2, the data-trained dictionary's curve being clipped at -20 in Figure 1(b).

The margin replicates at seven, ten and fourteen days, and Table 7 gives the configurations: at the three lags every seed's basic interval lies above zero, as in Table 3, while at one and three days not every seed's does.

Table 7: With the full-state MLP, the core-trained dictionary's held-out margin over the data-trained dictionary clears zero in every seed at seven, ten and fourteen days, as with the per-wavenumber encoder, and not in every seed at one and three days. Rank 60, shrinkage 0.01, seeds 0, 1 and 2 separately, with basic 90% intervals from the paired bootstrap of Appendix B.2.

lag	margin, seeds 0 / 1 / 2	basic 90% intervals above 0
1 d	$+1.49$ / -2.53 / -5.87	0 of 3
3 d	$+20.41$ / -3.43 / $+26.27$	2 of 3
7 d	$+15.05$ / $+22.67$ / $+34.43$	3 of 3
10 d	$+19.27$ / $+19.73$ / $+20.34$	3 of 3
14 d	$+25.66$ / $+14.81$ / $+25.84$	3 of 3

E.1.2 ENCODER WIDTH

To follow the margin as the encoders' capacity grows, the seven- and fourteen-day configurations of Table 3 are trained at one, two and four times the encoder's width, with the global batch held at 512, and read at rank 60 and shrinkage 0.01. The one-times dictionaries here are trained separately from those of Table 3.

At fourteen days the margin does not fall as the width grows, $+12.123$, $+13.563$ and $+10.581$ across the three widths against a pooled seed standard deviation of 1.139, and the data-trained dictionary's generalization gap does not rise either, 46.340, 50.399 and 50.015. The core-trained dictionary's own generalization gap rises, 20.977 to 22.838, and the spread of its per-pair scales falls, 0.084 to 0.049, the first scale's size minus the mean size of the 51st to 60th (Appendix C.8). Every margin at four times the width is above zero with its basic interval, at $+9.950$, $+11.084$ and $+10.708$.

At seven days the margin falls, $+6.398$, $+6.363$ and $+2.517$ against a pooled seed standard deviation of 0.535, and the spread of the per-pair scales falls with it, 0.071 to 0.033. That spread is the quantity Appendix C.8 reports as not all one at the capacity we train. The fall is the core-trained dictionary's, from 21.5 to 22.8 at the reported width down to 19.0 to 19.4 at four times it, while the data-trained

dictionary edges up. Every seed's margin at four times the width is still above zero, at $+2.865$, $+2.124$ and $+2.563$, each basic interval above zero.

Two quantities reverse sign at four times the width when the learning rate is quartered, so neither is read here: the data-trained dictionary's generalization gap at seven days, where the Spearman correlation holds, and the Spearman correlation at fourteen. A configuration is flagged as not converged when its held-out score moves by more than its seed spread over its last 20,000 steps. At fourteen days no configuration at four times the width is flagged, against ten of the twelve at the other widths, and at seven days three of six are, against seven of twelve. Width moves the encoder's capacity and its optimization together, so this sweep does not separate them.

E.2 Data

The three subsections below vary the resolution, the length of the training data and the period they are read on.

E.2.1 The margin at T42

To check that the margin does not depend on the T21 truncation of Section 4, the seven- and fourteen-day configurations are also trained and scored at T42, with everything else as Table 1 gives. The core is the same on eight levels with six-hourly states, training from 1979-01-01 to 2011-12-23 and holding out 2012 to 2020, and encoders of width 512 with 200 features trained for 120,000 steps. At the fixed protocol, rank 60 and shrinkage 0.01 over the whole held-out block, the margin of the core-trained dictionary over the data-trained dictionary is $+11.551$, $+11.782$ and $+10.802$ at seven days, and $+22.659$, $+21.487$ and $+23.021$ at fourteen. The sign and the growth with the lag are those of the T21 configurations of Table 3, and the sizes are about 1.6 and 1.8 times as large. At fourteen days seed 1 fails in training and is left out, so the values there are seeds 0, 2 and 3, and at seven days seeds 0 to 2.

The values above are at the fixed protocol and are not to be read against the selected-shrinkage values of Appendix D.3.1; at T42 the selected protocol is scored for the seven-day seeds and for fourteen-day seeds 0 and 2. At half the budget, 60,000 steps, the margin is read at each dictionary's own two-fold shrinkage: at seven days it is negative in every seed, and at fourteen days its basic interval lies above zero in one seed of three. At 120,000 steps and the same shrinkage it is positive with every basic interval above zero at seven days, and at fourteen days in seeds 0 and 2.

E.2.2 Years of training data

A margin that came from scarce data would be largest at four years and gone by thirty-three, so both dictionaries are trained at seven days on four, eight, sixteen and thirty-three years, each early-stopped on its own rows, with the protocol held fixed. The benchmark varies the same axis by trajectory count, in the subsection on gaps by data size below. That is not what we read: the core-trained dictionary is behind at four years, and from eight years on the mean margin over seeds is positive and does not shrink as the data grow. The seed means are -0.18, $+3.68$, $+3.90$ and $+4.80$ at four, eight, sixteen and thirty-three years, and seed 0 reads -0.62 at eight. The four-year configuration locates nothing, however, because neither dictionary fits its own training block there, both training scores being a fraction of their thirty-three-year values. These dictionaries are early-stopped, so the thirty-three-year value is not the one Appendix D.2 reports.

The spread over seeds is wide at eight and sixteen years and narrow at thirty-three. The two configurations responsible are opposite cases: at one the data-trained dictionary fits the training block and keeps almost nothing of it, at the other it barely fits at all and the core-trained dictionary is the one with the large generalization gap.

E.2.3 Season and period

The score is uncentered in time, so a pair locked to the annual cycle would carry a near-one correlation without predicting anything else. Reading each pair's contribution beside the share of its training projection that its own climatology explains, only the leading three or four of either dictionary are seasonal, and together they hold about 1.5 to 2 in every dictionary and basis alike. That

is a fraction of the long-lag margins. A period is one of three consecutive thirds of the held-out block, with the embargo removed at each internal boundary. Read period by period, the core-trained dictionary is ahead in all 27 configurations of three long lags, three periods and three seeds, so a handful of calendar-locked pairs does not account for the margin.

Pair by pair, for the core-trained dictionary at seven, ten and fourteen days, 19 to 26 pairs have a training correlation above 0.9 and hold 12.0 to 14.1 of a score of 20.3 to 23.7. The median seasonal share of those pairs is 0.03 to 0.11, while the leading three or four are at shares of 0.9 to 0.99. The POD basis read at the state is more seasonal still: four of its first five coefficients are almost entirely explained by their own climatology, and its leading pair contributes negatively held out.

Both dictionaries score lower over 2018 to 2020 than over the earlier thirds. At the body's fixed protocol, on that third of the held-out block and averaged over three seeds without an interval, the margin is $+11.0$, $+13.4$ and $+17.6$ at seven, ten and fourteen days.

E.3 Span reproducibility across seeds

The objective fixes a dictionary only up to an invertible map of its features (Appendix C.3), so reproducibility across training seeds is measured on spans, and both dictionaries' leading spans reproduce, the core-trained dictionary's more closely on average. Figure 8 reads the overlap of two seeds' leading-k spans against k, from the future-state functions at rank 60 and shrinkage 0.01 on the training block. At seven, ten and fourteen days the core-trained dictionary is above the data-trained dictionary at all but one to three values of k, and both are far above the random-subspace null. The basis is centered with the constant direction dropped, so k counts the modes after it; the raw basis gives the same ordering. Mode by mode, more of the core-trained dictionary's leading modes pass a fixed agreement threshold, at each of the three thresholds we read, 0.8, 0.9 and 0.95.

The separation is between the means, and it is not uniform. The bands of k are 1 to 5, 6 to 20, 21 to 40 and 41 to 59. In 8 of the 12 combinations of a lag and a band, the core-trained dictionary's worst seed pair clears the data-trained dictionary's best, failing only at k below 21 at seven and ten days.

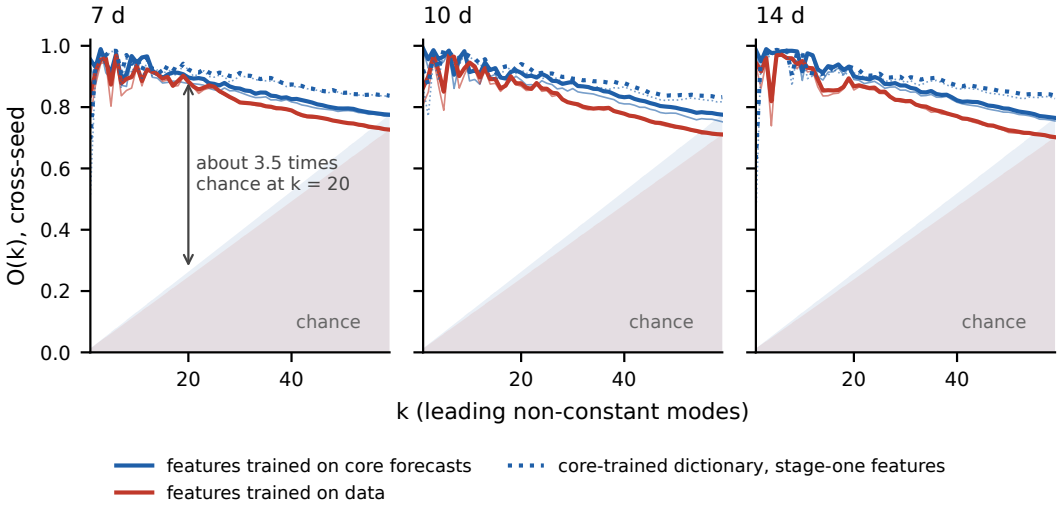

Figure 8: Both dictionaries' leading spans reproduce across training seeds, the core-trained dictionary's more closely on average, and both far above chance. $O(k)$ is the mean squared cosine of the principal angles between two seeds' leading-k spans: thick the seed-pair mean, thin their minimum, dotted the core-trained dictionary's stage-one features, and the filled band each curve's random-subspace null, which the measured overlap clears about three and a half times at $k = 20$.

F BENCHMARK RESULTS

This appendix gives the benchmark results behind Section 5 in full: the margins at one to seven days, the curves along lag and rank, and the training and held-out scores at two data sizes.

F.1 BENCHMARK MARGINS AT ONE TO SEVEN DAYS

Table 8 gives every benchmark configuration scored at one to seven days, behind the forcing axis of Section 5 and its margins at 1024 trajectories. At 256 trajectories the core-trained dictionary is level with the data-trained dictionary at one day and ahead at three to seven days, with every interval clearing zero. At 1024 the two separate at seven days, and at five days under the weakest forcing they are separated the other way under both protocols, as are, on the selected protocol's basic interval only, one day under the weakest forcing and three days under the middle one.

Table 8: Shallow water, seed 0: held-out VAMP-E at rank 60 and shrinkage 0.01, with the margin of the accurate core's dictionary over the data-trained dictionary and its basic 90% interval, for every scored configuration at one to seven days, with the accurate core at the truth's viscosity and the less viscous core at a tenth of it; the longer lags are in Figure 4 and Figure 11. At this protocol a basic interval can exclude its own point estimate, as it does in the three-day configuration at the strongest forcing; see Appendix B.2.

lag	r	N	accurate	less viscous	data	margin	basic 90%
1 d	0.02	1024	51.194	51.290	51.115	$+0.079$	$[-1.572, +0.291]$
1 d	0.02	256	49.027	49.152	48.996	$+0.030$	$[-1.376, +0.314]$
1 d	0.3	1024	50.701	50.964	49.764	$+0.938$	$[-2.407, +1.294]$
1 d	0.3	256	49.160	48.850	47.730	$+1.430$	$[-0.194, +1.996]$
3 d	0.02	1024	46.100	45.656	46.457	-0.357	$[-1.725, +1.411]$
3 d	0.02	256	40.197	40.771	38.409	$+1.788$	$[+0.413, +2.178]$
3 d	0.3	1024	42.986	43.585	43.184	-0.198	$[-2.414, +0.238]$
3 d	0.3	256	37.535	37.683	32.299	$+5.236$	$[+4.075, +5.603]$
3 d	1.0	256	36.253	35.910	28.259	$+7.994$	$[+6.272, +7.934]$
5 d	0.02	1024	39.085	39.587	40.478	-1.394	$[-2.103, -0.871]$
5 d	0.02	256	27.566	26.152	22.911	$+4.655$	$[+3.404, +5.599]$
5 d	0.3	1024	35.538	34.552	34.252	$+1.286$	$[-0.092, +1.621]$
5 d	0.3	256	23.496	23.934	8.783	$+14.712$	$[+12.101, +14.965]$
5 d	1.0	256	21.555	22.629	2.501	$+19.053$	$[+17.426, +19.356]$
7 d	0.02	1024	34.405	35.202	32.054	$+2.352$	$[+0.701, +2.435]$
7 d	0.02	256	12.062	14.585	2.710	$+9.352$	$[+7.144, +10.833]$
7 d	0.3	1024	28.842	28.875	24.053	$+4.789$	$[+2.135, +4.932]$
7 d	0.3	256	14.311	13.953	-5.507	$+19.819$	$[+18.187, +20.910]$
7 d	1.0	256	12.327	14.614	-13.670	$+25.998$	$[+24.314, +27.418]$

F.2 BENCHMARK CURVES ALONG LAG AND RANK

Figure 4 draws the benchmark's margins over lag and forcing; Figure 9 draws the held-out score of every dictionary and basis along the lag at each forcing, and Figure 10 the training and held-out scores along the rank at seven days under the two stronger forcings. The configurations at ten and fourteen days come from a separate set of thirty-day trajectories while the shorter lags come from twenty-day ones, so the lag axis joins two sets between seven and ten days. Both dictionaries score higher at ten days than at seven at every forcing, which we attribute to the change of trajectory set rather than to the lag.

At seven to fourteen days the POD basis read at the accurate core's forecast, which inherits that forecast's accuracy, scores above the core-trained dictionary by less as the forcing rises, in held-out VAMP-2 (Figure 4(c)) and in held-out score (Figure 9). Under the strongest forcing it is below the core-trained dictionary at ten and fourteen days, so the two cross where the trajectory sets meet. Its excess over the core-trained dictionary is not monotone in the lag at any forcing, and fourfold more data does not move it at ten days.

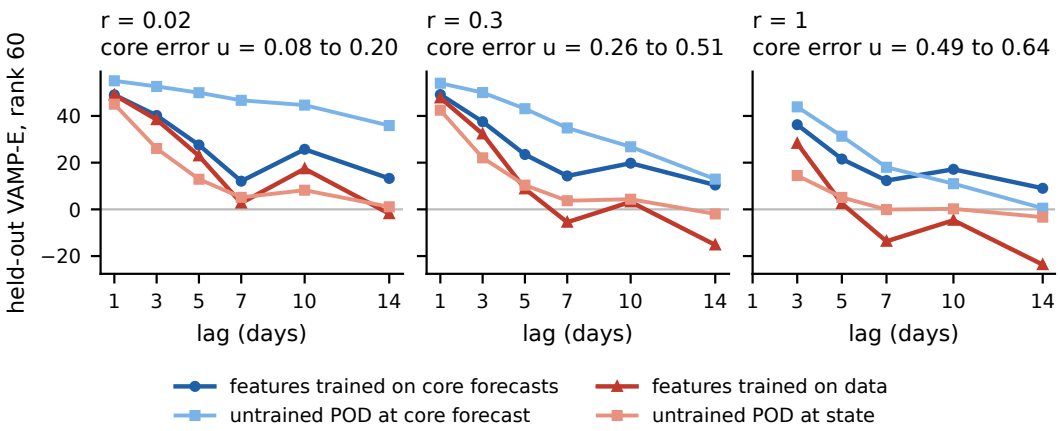

Figure 9: Every dictionary's and basis's held-out score falls as the lag grows, and the untrained POD basis read at the core's forecast is above the core-trained dictionary at the two weaker forcings. Held-out VAMP-E against lag in the benchmark, one panel per backscatter ratio, at rank 60, shrinkage 0.01, 256 trajectories and one seed; each panel is titled with its ratio r and with the relative forecast error u at the first and last lag it plots.

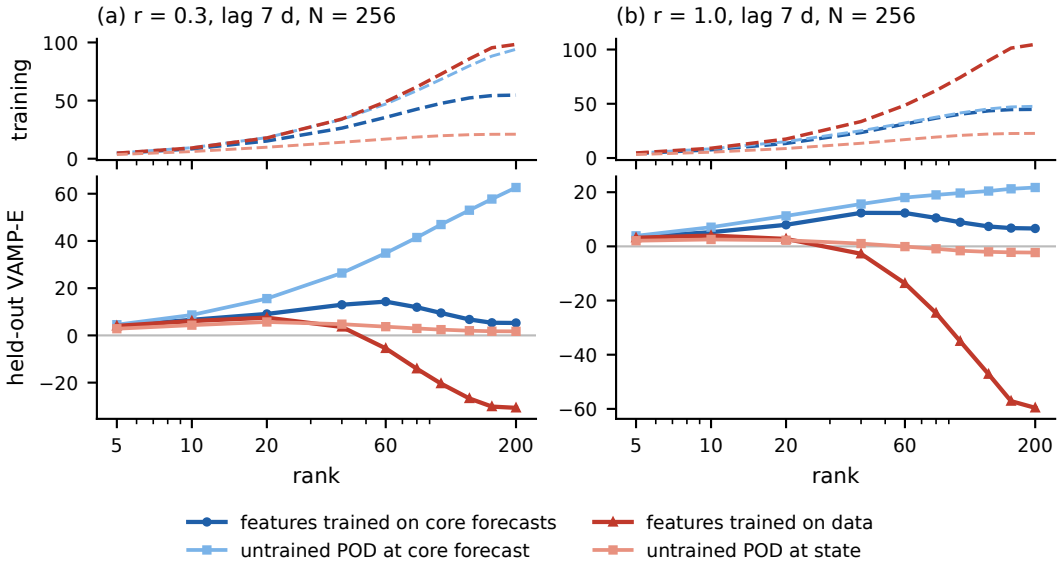

Figure 10: The same shape appears where the forcing is set rather than found: the data-trained dictionary's held-out score falls far past its peak, the core-trained dictionary's declines gently, and the margin widens with the forcing. VAMP-E against rank at seven days and shrinkage 0.01, training dashed and held out solid, at backscatter 0.3 and 1.0, in the colors of Figure 1; both POD bases need the explicit state ensemble, which exists at this resolution and not at an operational one.

F.3 Generalization gaps by data size

The number of training trajectories gives a finite-sample check of the noise argument. The data-trained dictionary's generalization gap is expected to grow more than the core-trained dictionary's as the training pairs grow fewer. The margin itself is not predicted, since the core-trained dictionary is learned from the states of those pairs too. Table 9 gives the training and held-out scores of the three dictionaries at 256 and 1024 trajectories, and Table 8 the margins at 1024. More trajectories narrow every dictionary's generalization gap, the data-trained dictionary's the most in all but one

configuration scored at both sizes. The atmosphere varies the same axis by the length of the training window, in Appendix E.2.2.

Table 9: Every dictionary's generalization gap shrinks as the data grow, and the data-trained dictionary's shrinks most in all but one configuration. Training and held-out VAMP-E of the three dictionaries at two data sizes, each configuration training over held out, at rank 60 and shrinkage 0.01 on the whole held-out block, one training seed; configurations scored at only one data size are left out.

lag	r	N	accurate	less viscous	data
1 d	0.02	256	56.39 / 49.03	56.35 / 49.15	56.68 / 49.00
1 d	0.02	1024	56.12 / 51.19	56.37 / 51.29	56.41 / 51.12
1 d	0.3	256	55.69 / 49.16	55.76 / 48.85	55.99 / 47.73
1 d	0.3	1024	55.31 / 50.70	55.90 / 50.96	55.78 / 49.76
3 d	0.02	256	52.06 / 40.20	52.21 / 40.77	52.94 / 38.41
3 d	0.02	1024	52.02 / 46.10	51.73 / 45.66	52.42 / 46.46
3 d	0.3	256	49.28 / 37.53	49.09 / 37.68	51.80 / 32.30
3 d	0.3	1024	48.35 / 42.99	48.30 / 43.59	49.36 / 43.18
5 d	0.02	256	46.54 / 27.57	46.34 / 26.15	49.70 / 22.91
5 d	0.02	1024	46.59 / 39.09	46.91 / 39.59	47.95 / 40.48
5 d	0.3	256	41.62 / 23.50	42.57 / 23.93	49.78 / 8.78
5 d	0.3	1024	40.93 / 35.54	41.05 / 34.55	42.95 / 34.25
7 d	0.02	256	42.81 / 12.06	41.37 / 14.59	49.63 / 2.71
7 d	0.02	1024	42.10 / 34.41	41.21 / 35.20	42.96 / 32.05
7 d	0.3	256	35.57 / 14.31	37.16 / 13.95	49.01 / -5.51
7 d	0.3	1024	34.65 / 28.84	34.77 / 28.88	38.59 / 24.05
10 d	1.0	256	26.31 / 17.16	29.86 / 18.12	43.42 / -4.67
10 d	1.0	1024	25.86 / 22.00	26.59 / 23.30	31.47 / 21.90
14 d	1.0	256	20.23 / 9.03	23.20 / 10.90	44.46 / -23.67
14 d	1.0	1024	18.38 / 15.35	20.19 / 16.91	26.17 / 8.49

G Where the margin comes from

The subsections here take the candidate sources of the margin in turn: the accuracy of the core's forecast, a target free of transition noise, the span against the fit, and the reuse of the transitions the span was trained on.

G.1 What the core supplies

The three subsections below take what the core offers a dictionary: the accuracy of its forecast, the difference between its two versions, and a target with neither noise nor dynamics.

G.1.1 The core's forecast error

The core's forecast is further from the future in the data than the present state is at three to fourteen days. Its relative forecast error u, over the whole held-out block in the unweighted spectral coefficients, is 0.778 at one day and then 1.113, 1.344, 1.435, 1.473 and 1.476 at three, five, seven, ten and fourteen days.

G.1.2 The two cores

Section 5.3 reads what the core-error hypothesis of Section 3.4 allows, that a core whose forecast has the larger relative error u need not give a worse dictionary, on the two cores' held-out VAMP-2 scores, which no shrinkage enters. Table 10 gives each core's relative forecast error at ten and fourteen days under the middle forcing, beside the score of the POD basis read at its forecast.

The separation between the two cores' VAMP-2 scores depends on the forcing. Table 11 gives the difference in every configuration: the less viscous core's VAMP-2 minus the other's, so a positive

Table 10: The two cores at ten and fourteen days, at backscatter 0.3 with 256 training trajectories: each forecast's relative error u, and the untrained POD basis read at that forecast, held out at rank 60 and shrinkage 0.01. Below, the less viscous core's dictionary minus the accurate one's, in both seeds, with the basic 90% interval of that difference and not of the rows above.

	10 d		14 d	
core	u	POD at forecast	u	POD at forecast
accurate, the truth's viscosity	0.504	26.804	0.507	12.938
less viscous, a tenth of it	0.624	21.290	0.637	7.042
seed 0: $+1.682$ $[+1.467, +2.751]$ at 10 d, $+0.581$ $[+0.318, +1.554]$ at 14 d				
seed 1: $+0.593$ $[+0.341, +1.612]$ at 10 d, $+1.171$ $[+0.892, +2.089]$ at 14 d				

value means the core with the larger u spans more. The separation grows with the forcing at ten days, and at fourteen days it is largest at the middle forcing rather than at the strongest, so the forcing is not a knob that widens the gap without limit. At the weakest forcing the fixed-protocol score difference between the two cores reads $+2.09$ and -1.55 at ten days and $+1.81$ and $+1.99$ at fourteen, in the two seeds and without an interval. At the strongest forcing and fourteen days the score difference is -0.667 in seed 1, the one configuration under the two stronger forcings where the less viscous core's dictionary scores lower, while the VAMP-2 scores tie.

Table 11: The two cores' held-out VAMP-2 scores separate more as the forcing rises at ten days, and at fourteen days most at the middle forcing rather than at the strongest. The less viscous core minus the accurate one at 200 features in every configuration where the two are compared, so a positive value means the core with the larger u spans more. The two seeds are separate trainings rather than a resampling, and no interval is attached to a VAMP-2.

forcing r	lag	seed 0	seed 1
0.02	10 d	-0.304	-1.145
	14 d	-0.840	$+1.280$
0.3	10 d	$+2.959$	$+2.230$
	14 d	$+4.000$	$+2.039$
1.0	10 d	$+6.762$	$+3.421$
	14 d	$+0.223$	-0.047

G.1.3 IDENTITY TARGET

To separate what the core's transitions contain from their being free of transition noise, a third dictionary is trained with the same objective on the identity target, which pairs each state with itself and so has neither transition noise nor dynamics. It is trained over nineteen configurations at seed 0 and 120,000 steps, and scored at the fixed and the two-fold protocols. The configurations are one and three days at r of 0.02 and 0.3 with 256 and 1024 trajectories, three days at $r = 1.0$ with 256, five and seven days at all three forcings with 256, and ten and fourteen days at r of 0.3 and 1.0 with 256. It solves its own task on held-out states in all nineteen: the pair it learns reproduces the state it is given.

Scored on transitions in the data it is far below the dictionaries the body reports, at 0.77 to 1.06 at ten and fourteen days against the accurate core's 9.03 to 19.77, and below the untrained POD basis read at the state in all fifteen settings where both are read. At ten and fourteen days it scores about one, what the constant function alone scores. At the fixed protocol the data-trained dictionary falls below it in five of the nineteen configurations, all of them configurations where the data-trained dictionary's own held-out score is negative; at the two-fold protocol it is above it in all nineteen.

G.2 Span and fit parts of the margin

G.2.1 VAMP-2 each span loses held out

This subsection takes two measurements: the margin's division between the held-out VAMP-2 and the estimation error, and the VAMP-2 each dictionary's span loses from the training block to the held-out block, which Section 6 reports.

The decomposition equation 4 separates what a dictionary's two spans reach from what the fit over them estimates, so a margin divides into a difference of held-out VAMP-2 scores, which belongs to the spans, and a difference of estimation errors, which belongs to the weights. Figure 11 reads that division at each lag in the benchmark. It groups the two approximation terms, because a VAMP-2 difference is minus their sum and equation 10 does not separate the input side from the output side.

A VAMP-2 from finitely many rows exceeds its population value, and the rows here are not independent. Figure 11 draws a correction for that bias, a delete-one-unit jackknife, with a lag whose seeds disagree marked mixed. The unit is a trajectory here. In the benchmark at shrinkage 0.01 the jackknife puts most of the margin in the estimation error from five to ten days at both forcings and at fourteen days at $r = 1.0$. The margin itself does not turn on any of this, since the same bias enters a VAMP-2 and the estimation error beneath it and cancels in their difference.

The data-trained dictionary's span does not reproduce its training VAMP-2 on the held-out block; the core-trained dictionary's largely does. A VAMP-2 is a property of a span, so the same pair can be read on the training rows and on the held-out rows. In the atmosphere, from five to fourteen days the data-trained dictionary's span loses between 92.7 and 104.4 of VAMP-2, in every lag and every seed, while the core-trained dictionary's loses at most 3.9; bias-corrected, 104.8 to 125.8 against at most 15.0. In the benchmark the data-trained dictionary's span loses more of its training VAMP-2 than the core-trained dictionary's in all 14 configurations under the two stronger forcings from three to fourteen days at 256 trajectories, bias-corrected. By equation 3 that is where a difference should fall: dictionary learning chooses the data-trained dictionary's spans on one draw of the data's futures, and the core-trained dictionary's are not trained on them.

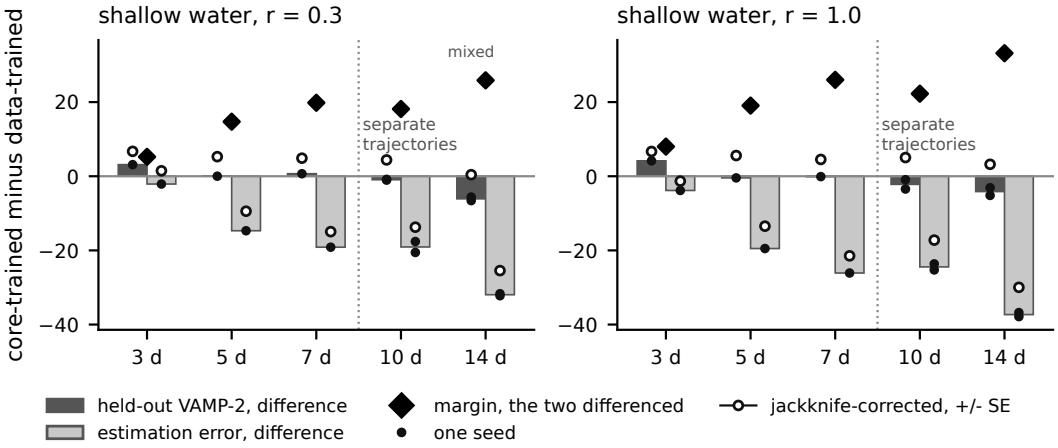

Figure 11: The core minus the data-trained dictionary in held-out VAMP-2 and in fit's estimation error at each lag, with their difference, the margin, as a diamond, and a bias correction. Bars are seed means and dots the seeds, at rank 60 and shrinkage 0.01 under the two stronger forcings; the jackknife correction is the white circles, and the dotted line is where the benchmark changes trajectory set.

G.2.2 Independent refitting

To measure the share of the margin that comes from fitting the operator on the transitions the span was trained on, the data-trained dictionary's coefficients are refitted on transitions disjoint from those, with its span and encoders untouched. Each configuration's encoders, trained on its 256 trajectories, are applied unchanged to 256 further trajectories of the same pool, disjoint from those

and from the 64 held out; the operator is fitted there at the fixed protocol and scored on the same held-out trajectories. The recovery is the data-trained dictionary's change under that refit minus the core-trained dictionary's change under the same refit, so a plain change of the fitting set is subtracted off. Table 12 gives the configurations. The core-trained dictionary's own change is between -0.227 and $+1.078$, which is why the recovery is read as the data-trained dictionary's.

One limit belongs with these numbers. The intervals resample the held-out rows only, in the moving blocks of Table 2. They are therefore conditional on this one draw of the independent trajectories and on the dictionaries as trained, and do not include the variability of drawing another set, which is the limit Appendix B.2 states for training.

Table 12: Refitting the data-trained dictionary's coefficients on trajectories its span was not trained on recovers most of its margin against the core-trained dictionary, which barely moves. Rank 60 and shrinkage 0.01 with 256 trajectories in each fitting set; the recovery is net of the core-trained dictionary's change under the same refit, and the share is of the core-minus-data margin at the original fit that it closes. The last column is the refitted core-trained dictionary's held-out VAMP-E minus the refitted data-trained dictionary's, over the same held-out trajectories, with its own basic interval.

r	lag	seed	recovery, basic 90%	share	margin after refit, basic 90%
0.3	7 d	0	$+15.818$ $[+14.134, +16.498]$	0.798	$+4.001$ $[+3.535, +4.906]$
	10 d	0	$+13.115$ $[+11.774, +13.608]$	0.792	$+3.443$ $[+3.154, +4.010]$
	14 d	0	$+18.763$ $[+17.136, +19.514]$	0.731	$+6.895$ $[+6.172, +7.534]$
	10 d	1	$+14.268$ $[+12.902, +15.109]$	0.724	$+5.451$ $[+3.367, +6.274]$
	14 d	1	$+20.115$ $[+18.441, +20.804]$	0.771	$+5.958$ $[+4.958, +6.471]$
1.0	7 d	0	$+18.066$ $[+16.051, +19.261]$	0.695	$+7.932$ $[+7.387, +8.793]$
	10 d	0	$+16.052$ $[+14.757, +16.529]$	0.735	$+5.775$ $[+5.336, +6.322]$
	14 d	0	$+24.489$ $[+22.291, +26.177]$	0.749	$+8.203$ $[+7.778, +8.828]$
	10 d	1	$+15.883$ $[+14.595, +16.492]$	0.700	$+6.816$ $[+6.368, +7.563]$
	14 d	1	$+24.398$ $[+22.834, +25.621]$	0.724	$+9.278$ $[+8.727, +10.031]$

H ADDITIONS TO THE CORE-TRAINED DICTIONARY

No fit over the core-trained dictionary reaches the part of $\mathcal{K}$ outside its span U, which lies in the two approximation terms of equation 10, and this appendix adds learned functions to that dictionary in two ways to reach it. Both follow the same three steps, and they differ only in the second.

1. Learn the core-trained dictionary $(\mathbf{f}_{\text{core}}, \mathbf{g}_{\text{core}})$ from the core, as in Section 3, and keep it fixed.
2. Learn a second pair of feature maps, whose number of features is stated with each result.
3. Concatenate the two and fit the operator on reanalysis over the concatenation, as in Section 3.1.

Table 13 sets the two side by side. In words, the combined dictionary learns where the core's forecast goes wrong, and the remainder dictionary learns what the core's functions cannot express at all. Forecast containment is not specific to the core's own dictionary: a dictionary trained on the core's error reproduces $\mathbf{g}_{\text{core}}(\Phi_{\text{core}}\mathbf{x})$ nearly as well.

H.1 COMBINED DICTIONARY

Construction. The combined dictionary adds a correction dictionary $(\mathbf{f}_{\text{corr}}, \mathbf{g}_{\text{corr}})$, trained on the core's error $\mathcal{K}^{\text{corr}} = \mathcal{K} - \mathcal{K}^{\text{core}}$, to learn where the core's forecast goes wrong. For each pair $(\mathbf{x}, \mathbf{y})$ in the data the core forecasts $\Phi_{\text{core}}\mathbf{x}$, and the training signal is the miss $\mathbf{g}(\mathbf{y}) - \mathbf{g}(\Phi_{\text{core}}\mathbf{x})$. The loss equation 2 then uses the cross-moment $\mathbb{E}[\mathbf{f}(\mathbf{x})\,(\mathbf{g}(\mathbf{y}) - \mathbf{g}(\Phi_{\text{core}}\mathbf{x}))^{\top}]$ in place of the one between present and future. The future-state features are normalized in the pullback metric, the second moment of that miss.

Table 13: The two additions to the core-trained dictionary.

	combined dictionary	remainder dictionary
learned on	the core's error $\mathcal{K}^{\mathrm{corr}} = \mathcal{K} - \mathcal{K}^{\mathrm{core}}$	the remainder $(\mathcal{I} - \mathcal{P}_{U_0})\mathcal{K}$
signal	the data's future minus the core's forecast, $\mathbf{g}(\mathbf{y}) - \mathbf{g}(\Phi_{\mathrm{core}}\mathbf{x})$	the data's future $\mathbf{g}(\mathbf{y})$
constraint	future-state features normalized in the pullback metric	present-state features kept orthogonal to U_0
reach beyond the block	the part of $\mathcal{K}^{\mathrm{corr}}$ outside U	all of the remainder, by construction
result	less than a second core-trained dictionary of the same number of features	no held-out gain at seven days
reason	$\mathcal{K}^{\mathrm{corr}}$ lies mostly inside U	

Result. At every core rank and lag, adding correction features to the first k_c core features scores below adding the same number of further core features, in every configuration but two. Both add forty features at a small core rank and beat the equal number of features by little: seven days, seed 0, $k_c = 5$, $+0.56$, and fourteen days, seed 1, $k_c = 10$, $+0.29$.

Absorption by the fit. The fit over the core-trained dictionary already recovers $\mathcal{P}_U\mathcal{K}^{\mathrm{corr}}$ by Proposition 1. So the correction dictionary helps only through the part of $\mathcal{K}^{\mathrm{corr}}$ outside U, and that part is small. A projection onto the core's span, fitted on the training block, explains 0.845, 0.907 and 0.928 of the correction dictionary's second moment there at one, three and seven days, and applied held out 0.816, 0.879 and 0.906. At three and seven days most of the small part left over also does not reproduce. Its summed squared canonical correlation is 46.3, 35.3 and 42.0 on the training block against 31.6, 9.0 and 4.4 held out. At seven days nine tenths of it is gone held out. The correction dictionary thus relearns mostly what U already holds, and what it adds beyond U is largely fitted sampling noise.

The principal angles between the spans show why, and Table 14 gives them. On the present-state side the correction dictionary lies almost inside the core-trained dictionary's span, and at small core rank it repeats the features already in use, where the core's own next features do not. It repeats less on the future-state side, yet the features it does not repeat still score below core features of the same number, which points to its target, the core's error, rather than what the core's span leaves of $\mathcal{K}$.

Table 14: Directions the correction dictionary shares with the core-trained dictionary, counted as principal angles with a cosine above 0.9 on the training block, at three, seven and fourteen days in three seeds. The last row is the control: the core-trained dictionary's own next features against the span they are added to.

	of the first 10	of the first 40
present-state side, against all 200 core features	9	36 to 37
future-state side, against all 200 core features	6 to 8	15 to 22
present-state side, against the first 10 core features	7 to 9	
the core's own next features, against the first k_c, to $k_c = 60$	at most 2	

H.2 Remainder dictionary

A dictionary trained on the remainder is ahead of an equal number of further core-trained features in every seed only at three days, and behind at fourteen and, with forty core features in place, from seven (Figure 12, Table 15).

I Extended related work

Transfer-operator estimators. Transfer-operator estimators fit the operator over a dictionary, which extended dynamic mode decomposition takes as given (Williams et al., 2015), kernel methods and the linear inverse model fix (Kostic et al., 2022; Penland & Sardeshmukh, 1995), and deep

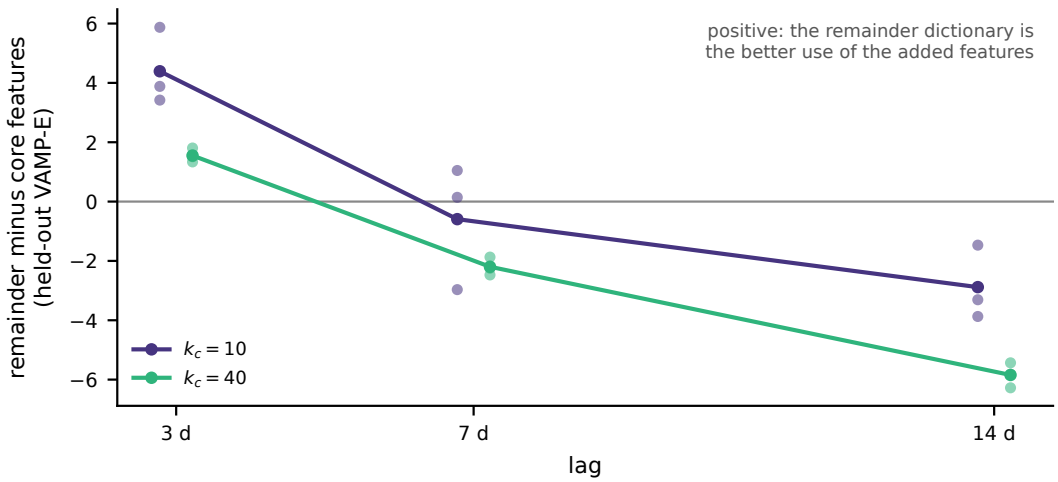

Figure 12: A second training aimed at the remainder repays only where the core-trained dictionary has not already spanned what the data support. It is the better use of the added features at three days with ten core features in place, and the worse one at fourteen days and, with forty in place, from seven. Held-out VAMP-E of the remainder dictionary minus the one extended by more core features, both built on the first k_c core features, at rank $\min(60, w)$ for w features and each one's own two-fold shrinkage over the whole held-out block; three seeds, configurations in Appendix H.

Table 15: The paired differences behind Figure 12: two dictionaries of the same number of features, differing only in what is added to the first k_c core features, a remainder dictionary or the core's own next features, the entry being the held-out score of the first minus the second. A positive value means the remainder features are the better use of the added features, and the last column counts the seeds whose basic 90% interval excludes zero.

first k_c core features	lag	remainder minus more core, by seed	seeds separated
10	3 d	$+3.42$ / $+5.87$ / $+3.88$	3 of 3
10	7 d	-2.97 / $+0.14$ / $+1.05$	1 of 3
10	14 d	-3.31 / -1.47 / -3.87	3 of 3
40	3 d	$+1.51$ / $+1.81$ / $+1.34$	0 of 3
40	7 d	-2.25 / -2.47 / -1.87	3 of 3
40	14 d	-5.82 / -5.43 / -6.28	3 of 3

Koopman learning trains (Takeishi et al., 2017; Lusch et al., 2018). VAMPnets (Mardt et al., 2018; Wu & Noé, 2020), DPNets (Kostic et al., 2024b) and related objectives (Pfau et al., 2019; Turri et al., 2026) train it by variational scores. The low-rank objective used here has the leading singular subspace as its population minimizer (Ryu et al., 2024; Jeong et al., 2025; Kostic et al., 2024a), which dictionary learning gives up by fitting against the core's operator (Section 3.3).

Data reuse. Fitting the operator to the pairs its features were selected on makes the training score overstate the held-out one, which McGibbon & Pande (2015) detect by cross-validation over independent trajectories. Analyzing a selection on the data it was made on, called double dipping, distorts the result unless its statistic is independent of the selection criterion or is computed on independent data (Kriegeskorte et al., 2009). Sample splitting supplies such data by spending pairs and needs independent trajectories (Section 1), while the objective that trains the core-trained dictionary does not depend on which future follows which state (Section 3.1).

Models of the dynamics. Operator estimation has used the known equations of motion, without data pairs (Valva & Giannakis, 2025), and constraints from known physics on the operator (Baddoo et al., 2023). Learned forecasters need no numerical model at inference (Lam et al., 2023; Bi et al., 2023; Price et al., 2025; Bodnar et al., 2025), with spherical architectures (Bonev et al., 2023) and self-supervised pretraining (Lessig et al., 2023). Hybrid models keep a core at inference (Kochkov

et al., 2024), with learned parameterizations (Rasp et al., 2018; Brenowitz & Bretherton, 2019; Yuval & O'Gorman, 2020) or a learned correction (Farchi et al., 2021; Watt-Meyer et al., 2021). Transfer learning pretrains a forecaster on climate-model simulations and fine-tunes it on reanalysis (Ham et al., 2019; Rasp & Thuerey, 2021), which Rasp & Thuerey (2021) use to avoid overfitting. Here the core, run from the states in the data, supplies only the dictionary's training target, never appears at inference and need not forecast accurately (Proposition 1); Appendix H scores a dictionary trained on its error.