OPINE: ORTHOGONAL PROJECTION IN INVERTIBLE NONLINEAR EMBEDDINGS FOR TURBULENT FLOWS

Anonymous authors
Paper under double-blind review

ABSTRACT

Proper orthogonal decomposition (POD) represents physical fields with orthogonal modes ordered by variance, so that one decomposition gives reduced coordinates in every dimension, but it needs many modes when the data lie near a curved manifold. Nonlinear autoencoders are more compact, but their information loss happens inside a nonlinear encoder, and a model trained with one latent dimension gives no model with a smaller one. We introduce Orthogonal Projection in Invertible Nonlinear Embeddings (OPINE), which learns an invertible nonlinear change of coordinates and performs POD in the transformed coordinates. Because the map is invertible, all information loss occurs in the POD projection, and ordering its basis after training gives reduced coordinates of every smaller dimension from a single model. Training alternates gradient updates of the map with a damped update that tracks the evolving POD subspace, and we prove that this update converges locally to the exact POD subspace. We evaluate OPINE on synthetic manifolds, chaotic Kuramoto–Sivashinsky dynamics, forced Burgers turbulence, two-dimensional Kolmogorov flow, and three-dimensional minimal channel flow. OPINE is more accurate than POD at every tested retained dimension above one, and on the Kuramoto–Sivashinsky and Burgers dynamics, it is more accurate than convolutional and fully connected autoencoders by more than two orders of magnitude at the largest retained dimension.

1 INTRODUCTION

A snapshot of a turbulent flow has thousands to millions of degrees of freedom, yet its dominant dynamics often lie near a much lower-dimensional set. Reduced coordinates that describe this set are the basis of data compression, flow analysis, and reduced-order models. The standard method for obtaining them is proper orthogonal decomposition (POD), known in statistics as principal component analysis. POD modes are orthogonal and ordered by the variance they capture, and for every r the leading r modes span the best r-dimensional linear subspace (Lumley, 1970; Berkooz et al., 1993; Taira et al., 2017). One decomposition therefore provides reduced models of every size, and each coordinate has a clear physical meaning.

This optimality, however, holds only among linear subspaces. Curved and transport-dominated solution sets, such as traveling waves, shocks, and the chaotic attractors of turbulent flows, can need many modes even when they have low-dimensional nonlinear parametrizations (Greif & Urban, 2019; Lee & Carlberg, 2020). Autoencoders overcome this limitation with nonlinear encoders and decoders, and they have improved on POD for fluid fields (Kramer, 1991; Milano & Koumoutsakos, 2002; Murata et al., 2020). However, a standard autoencoder has neither orthogonal modes nor an ordering, its encoder can map different states to the same latent vector, and a model trained with one latent dimension gives no model with a smaller one. Nested dropout, Matryoshka representations, and hierarchical autoencoders order the latent variables (Rippel et al., 2014; Kusupati et al., 2022; Fukami et al., 2020), but the truncation still happens inside a nonlinear encoder.

Our key idea is to keep POD as the *only* bottleneck, but to apply it in transformed coordinates. In *Orthogonal Projection in Invertible Nonlinear Embeddings* (OPINE), an invertible map, parametrized by a neural network, transforms the field without reducing its dimension, POD removes dimensions in the transformed coordinates, and the inverse map returns the result to physical space (Figure 1). Because the map is invertible and keeps the dimension of the field, no information is lost before

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.

the projection, and the whole model reduces to the identity when all coordinates are kept. OPINE can therefore be viewed as a search for an observable of the field, that is, a nonlinear change of variables in which the data lie close to a linear subspace, similar to the feature map of kernel PCA (Schölkopf et al., 1998) but invertible, so that no pre-image problem arises (Mika et al., 1999). Once this observable is found, the properties of POD, including orthogonal, ordered, and nested modes, come at no additional cost. Training is a coupled problem, because the POD basis changes whenever the map changes, and a sudden change of the basis would also change the loss suddenly. We therefore alternate gradient updates of the map with a damped, aligned step of orthogonal iteration. Our contributions are:

- **Method:** a factorization of nonlinear dimensionality reduction into an invertible change of coordinates and an explicit POD projection, which confines all information loss to the projection and gives reduced coordinates of every smaller dimension from one model (Section 3);
- **Training and theory:** a scheme that tracks the POD subspace without differentiating through an eigendecomposition, together with a proof that the damped update converges to the exact POD subspace and follows it closely while the map changes (Section 3.4; Appendix A);
- **Experiments:** from synthetic manifolds and one-dimensional chaotic dynamics to two- and three-dimensional turbulence, in which OPINE improves on POD at every retained dimension above one, and ablations that support the choice of the POD projection (Section 4).

2 Related Work

OPINE is related to three lines of work: autoencoders and nonlinear model reduction, ordered and invertible representations, and subspace tracking.

Autoencoders and nonlinear model reduction. Autoencoders learn a nonlinear encoder and decoder through a low-dimensional bottleneck and are the standard nonlinear generalization of POD (Kramer, 1991; Hinton & Salakhutdinov, 2006); variational autoencoders add a probabilistic latent space (Kingma & Welling, 2014). In fluid mechanics, autoencoders have compressed near-wall turbulence (Milano & Koumoutsakos, 2002), decomposed flows into nonlinear modes (Murata et al., 2020), built hierarchical or nearly orthogonal latent modes (Fukami et al., 2020; Eivazi et al., 2022), and learned coordinates for Koopman and inertial-manifold models (Lusch et al., 2018; Champion et al., 2019; Linot & Graham, 2020). In reduced-order modeling, convolutional autoencoders provide nonlinear trial manifolds for advection-dominated systems (Lee & Carlberg, 2020; Maulik et al., 2021), and POD precompression makes them cheaper to train (Fresca & Manzoni, 2022). Classical alternatives transport the data before a linear projection, as in shifted POD and registration (Reiss et al., 2018; Taddei, 2020), or add a polynomial correction to a POD subspace, as in quadratic manifolds (Geelen et al., 2023); these fix the form of the transformation. OPINE instead learns a general invertible map and applies POD *after* it, so that POD remains the only bottleneck.

Ordered and invertible representations. Invertible networks such as Real NVP, Glow, and invertible residual networks provide flexible bijective maps with tractable inverses (Dinh et al., 2017; Kingma & Dhariwal, 2018; Behrmann et al., 2019), and injective or manifold flows use such maps to learn low-dimensional manifolds (Brehmer & Cranmer, 2020; Caterini et al., 2021; Kothari et al., 2021; Cunningham et al., 2022). Closest to OPINE are nested-dropout flows, which order the coordinates of an invertible network and reconstruct by zeroing its trailing coordinates (Rippel et al., 2014; Bekasov & Murray, 2020), and Neural-PCA, which couples a flow to a PCA layer for likelihood-based representation learning (Li & Hooi, 2022). In principle, an unrestricted invertible map could absorb a global rotation, but stacks of local coupling layers cannot represent one cheaply. OPINE therefore keeps its bottleneck close to the POD subspace of the transformed data and updates it in closed form, which we show matters in practice. It is also trained for field reconstruction rather than likelihood.

Subspace tracking. Online PCA and subspace-tracking algorithms update dominant eigenspaces as the data evolve (Oja, 1982; Yang, 1995), whereas backpropagating through an eigendecomposition (Ionescu et al., 2015; Wang et al., 2019) is fragile near the repeated eigenvalues of paired Fourier modes. We use damped, aligned orthogonal iteration.

3 METHOD

Sections 3.1–3.3 present our first contribution, the factorization, and Section 3.4 the second, the training scheme. Let $x \in \mathbb{R}^{n_c \times N_1 \times \cdots \times N_d}$ be a discretized field with n_c components on a grid of $N_1 \times \cdots \times N_d$ points, and thus $N = n_c \prod_j N_j$ degrees of freedom. OPINE places an explicit projection inside an invertible change of coordinates:

$$x \xrightarrow{f_\theta} z \xrightarrow{\Pi_r} \hat{z}_r \xrightarrow{f_\theta^{-1}} \hat{x}_r. \tag{1}$$

Because f_θ is bijective, the projection Π_r is the only step that loses information.

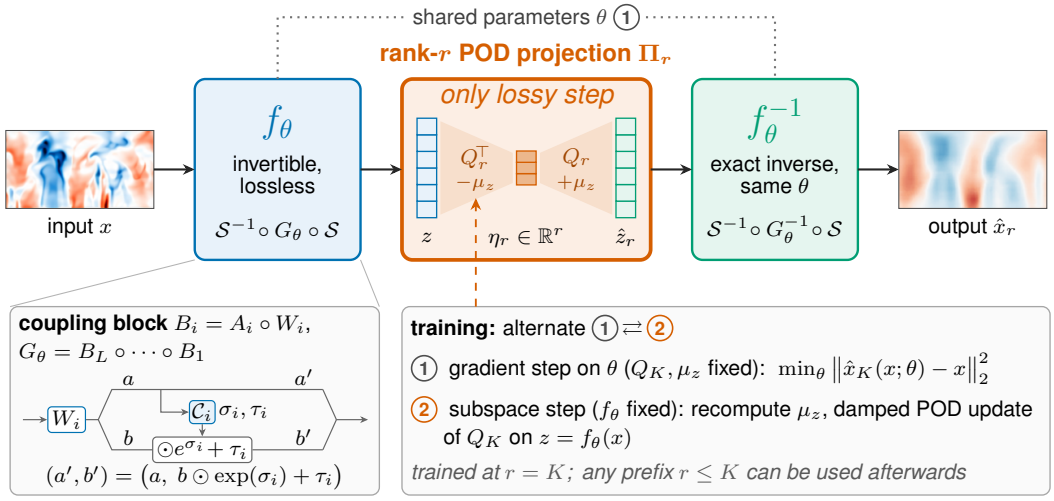

Figure 1: OPINE for a scalar field: only the rank-r POD projection of $z = \operatorname{vec} f_\theta(x)$ loses information, and f_θ and its exact inverse share θ. Training alternates the two steps at bottom right (Section 3.4). For multi-component fields, $\mathcal{S}$ is omitted and $f_\theta = G_\theta$.

3.1 REPRESENTATION

We first define the reduced coordinates. Let $z = \operatorname{vec} f_\theta(x) \in \mathbb{R}^N$, let μ_z be the mean of the transformed data, and let $Q_K = [q_1, \ldots, q_K]$ be an orthonormal basis of the learned K-dimensional subspace. For $r \le K$ and $Q_r = [q_1, \ldots, q_r]$, the reduced coordinates and the reconstruction are

$$\eta_r = Q_r^\top (z - \mu_z) \in \mathbb{R}^r, \qquad \hat{z}_r = \mu_z + Q_r \eta_r, \qquad \hat{x}_r = f_\theta^{-1}\big(\operatorname{unvec} \hat{z}_r\big). \tag{2}$$

Here $\Pi_r : z \mapsto \hat{z}_r$, and r counts real coefficients. The retained dimension D of the experiments equals r, and each model is trained with $K = D$. When all transformed coordinates are kept ($r = K = N$), the reconstruction is exact. Because Π_r is a projection and f_θ is bijective, $x \mapsto \hat{x}_r$ is a nonlinear projection onto an r-dimensional manifold, the kind of trial manifold used by manifold Galerkin and least-squares Petrov–Galerkin models (Lee & Carlberg, 2020).

3.2 INVERTIBLE MAP

The representation above holds for any invertible f_θ; we now describe the one we use. It is a stack $G_\theta = B_L \circ \cdots \circ B_1$ of blocks $B_i = A_i \circ W_i$, where W_i is an invertible channel mixing (Kingma & Dhariwal, 2018) and A_i is an affine coupling layer (Dinh et al., 2017) that splits the channels into (a, b) and maps

$$(a, b) \mapsto \big(a,\ b \odot \exp(\sigma_i) + \tau_i\big), \qquad (\sigma_i, \tau_i) = \mathcal{C}_i(a). \tag{3}$$

This layer can be inverted in closed form for any conditioner network $\mathcal{C}_i$. Multi-component fields use their physical components as channels, so that $f_\theta = G_\theta$. Scalar fields first pass through an invertible space-to-depth rearrangement $\mathcal{S}$, which folds small blocks of neighboring grid points into

channels, so that $f_\theta = \mathcal{S}^{-1} \circ G_\theta \circ \mathcal{S}$. We also tried invertible residual networks and monotone piecewise-quadratic maps, but affine coupling layers were the most accurate and the fastest to invert (Appendix C).

3.3 POD Bottleneck

The bottleneck is the POD of the transformed data. For fixed θ and transformed training states z_i with mean μ_z and covariance C_z, the rank-K POD basis

$$Q_K^\star \in \arg\min_{Q^\top Q = I_K} \sum_i \left\| (I - QQ^\top)(z_i - \mu_z) \right\|_2^2 \tag{4}$$

spans the leading K-dimensional eigenspace of C_z, whose eigenvalues we denote $\lambda_1 \geq \cdots \geq \lambda_N$.

3.4 Training

It remains to train the map and the basis together. The map f_θ minimizes the reconstruction error, while Q_K and μ_z must follow the POD of the transformed data, which changes as θ changes. We keep Q_K tied to that POD, rather than learning it as a free parameter, so that the retained subspace can be ordered by variance. Training alternates two updates, neither of which differentiates through an eigendecomposition. First, with Q_K and μ_z fixed, we take a round of gradient steps on

$$\mathcal{L}_\mathcal{B}(\theta) = \frac{1}{|\mathcal{B}|} \sum_{x \in \mathcal{B}} \left\| \hat{x}_K(x;\theta) - x \right\|_2^2 \tag{5}$$

over minibatches $\mathcal{B}$, with $\hat{x}_K$ from Eq. 2. Second, with f_θ fixed, we recompute μ_z and take one damped, aligned step of orthogonal iteration:

$$\widetilde{Q}_K = \mathrm{qf}(C_z Q_K), \qquad R = \arg\min_{R^\top R = I_K} \left\| \widetilde{Q}_K R - Q_K \right\|_F = U_R V_R^\top, \tag{6}$$

$$Q_K \leftarrow \mathrm{qf}\big[(1-\alpha) Q_K + \alpha \widetilde{Q}_K R\big], \tag{7}$$

where qf is the orthonormal factor of a thin QR factorization and $\widetilde{Q}_K^\top Q_K = U_R \Sigma V_R^\top$ is an SVD. Orthogonal iteration moves the subspace toward the dominant eigenspace but may rotate the basis within it, and the Procrustes rotation R undoes this rotation so that the old and new bases can be interpolated. Since $Q_K^\top \widetilde{Q}_K R = V_R \Sigma V_R^\top \succeq 0$, the update is well defined for every $\alpha \in [0, 1]$, and a smaller α gives stronger damping. Appendix A proves that this update converges to the exact POD subspace and follows it closely while f_θ changes.

After training, we diagonalize $Q_K^\top C_z Q_K = E \Lambda_Q E^\top$ and set $Q_K \leftarrow Q_K E$. This leaves the projection unchanged and orders the basis by variance, so each prefix Q_r spans the most energetic r-dimensional subspace of $\mathrm{span}(Q_K)$, and one model gives reduced coordinates for every $r \leq K$ (Section 4.6).

4 Experiments

Our third contribution is an evaluation of OPINE on datasets of increasing dimension, from synthetic one-dimensional profiles to three-dimensional turbulence. We ask four questions: whether OPINE finds transformed coordinates in which data of known low dimension have low rank (**Q1**; Section 4.2); whether it improves on POD and autoencoders with the same number of retained coefficients (**Q2**; Sections 4.3–4.5); whether its improvement over POD comes from the transformed coordinates or from the choice of projection (**Q3**; Section 4.2); and whether one trained model is accurate at every smaller D (**Q4**; Section 4.6).

4.1 Baselines and Evaluation

To keep the comparison fair, every method stores the same number D of real coefficients per snapshot: $r = D$ for POD and OPINE, and a D-dimensional latent vector for the autoencoders. POD is Eq. 2 with f_θ the identity. The CNN autoencoder compresses the field to its latent vector with

strided convolutions and boundary-aware padding, and reconstructs it with a mirrored decoder (Figure 7 in Appendix C). For the one-dimensional problems we also report a fully connected (MLP) autoencoder. OPINE also has far fewer parameters: for the turbulent flows, it has 6×10^4 to 10^5 parameters, whereas the CNN autoencoders have 10^6 to 2×10^7. All learned models are trained separately at each D and selected on validation data (Appendix C). We report the test normalized mean squared error (NMSE) $\sum_{i \in \mathcal{T}} \|x_i - \hat{x}_i\|_2^2 / \sum_{i \in \mathcal{T}} \|x_i\|_2^2$ on a held-out set $\mathcal{T}$ shared by all methods.

4.2 Synthetic Manifolds and Ablations

We begin with synthetic data whose low-dimensional parametrization is known exactly. The family $x_\phi(s) = \tanh[\sin(s+\phi)]$ on a periodic grid, with a random phase $\phi \sim \mathcal{U}(0, 2\pi)$, is a one-parameter set of traveling profiles. It is not low rank in the original coordinates, because the $\tanh$ generates all odd harmonics, but it is exactly rank 2 in the invertible coordinates $\operatorname{atanh} x_\phi = \sin s \cos\phi + \cos s \sin\phi$. We train at $D = 2$ with 32 samples and ten seeds. OPINE finds such coordinates (Q1): two POD modes of its transformed coordinates hold 99.99% of the transformed variance, and its test error is lower than that of POD on every seed and 28 times lower on average, whereas the CNN autoencoder overfits the small training set.

We then use four synthetic families of profiles on 128 points of $s \in [0, 2\pi)$ to identify the source of the improvement over POD (Q3). In *Pulse*, $x = \operatorname{sech}^2(d/0.3)$, where d is the periodic distance from a random center c; its states form a closed loop. *Pulse width* is the same pulse with a random width in $[0.2, 0.6]$ instead of 0.3. In *Front*, $x = \tanh[\kappa \sin(s - c)]$ with a random position c and steepness $\kappa \in [1, 5]$. In *Torus*, $x = \tanh[\sin(s + \phi_1) + \sigma(\cos(2s + \phi_2))]$ with two random phases and the logistic function σ; its states form a torus. POD needs many modes for all four, because the parameters move or deform the profiles. Table 1 retrains the same invertible map with different projections and otherwise identical training. Keeping the first D entries of z, as nested-dropout flows do at a fixed dimension, is usually no more accurate than plain POD. A fixed Fourier basis improves on POD on three of the four families, but OPINE is the most accurate on all four. In Figure 2, the pointwise error of OPINE is about two orders of magnitude below that of POD along the whole profile, whereas keeping the first D entries of z gives noisy reconstructions. For these translation-invariant families, POD of the raw data is itself a Fourier basis ordered by variance. The fixed Fourier basis differs from OPINE because the invertible map is not translation-equivariant, so the POD modes of z are not Fourier modes, and because the basis is ordered by wavenumber rather than by variance. Table 1 is an ablation of the projection with the architecture and training of the invertible map held fixed; OPINE is compared with the autoencoders on the dynamical systems below, where the benefit of the transformed coordinates is largest.

On the same families, damped tracking is as accurate as exact recomputation of the POD, consistent with Appendix A, whereas freezing the subspace at its initial POD roughly triples the error on Pulse. Tracking is also much cheaper than differentiating through an eigendecomposition (Appendix B).

Table 1: Bottleneck ablation at $D = 4$: median test NMSE. POD projects the raw data. The other three columns retrain the same invertible map with a different projection: keeping the first D entries of z and setting the others to their training mean, onto a fixed Fourier basis, or onto the POD subspace of the transformed field, which is OPINE. Best in bold. Figure 2 shows sample reconstructions.

		same invertible map, projection onto:		
	POD	first D entries	Fourier basis	OPINE
Torus	4.8×10^{-3}	5.3×10^{-1}	7.1×10^{-3}	$\mathbf{1.9\times10^{-5}}$
Front	1.0×10^{-2}	2.4×10^{-1}	3.9×10^{-5}	$\mathbf{6.0\times10^{-6}}$
Pulse	3.8×10^{-1}	3.4×10^{-1}	4.1×10^{-4}	$\mathbf{1.3\times10^{-4}}$
Pulse width	2.5×10^{-1}	5.6×10^{-1}	9.6×10^{-4}	$\mathbf{6.8\times10^{-4}}$

4.3 One-Dimensional Dynamics

Having confirmed that OPINE finds known low-dimensional coordinates, we turn to two one-dimensional systems whose dynamics concentrate near a low-dimensional set that is not known in advance. The Kuramoto–Sivashinsky (KS) equation $u_t + uu_s + u_{ss} + u_{ssss} = 0$ on a periodic

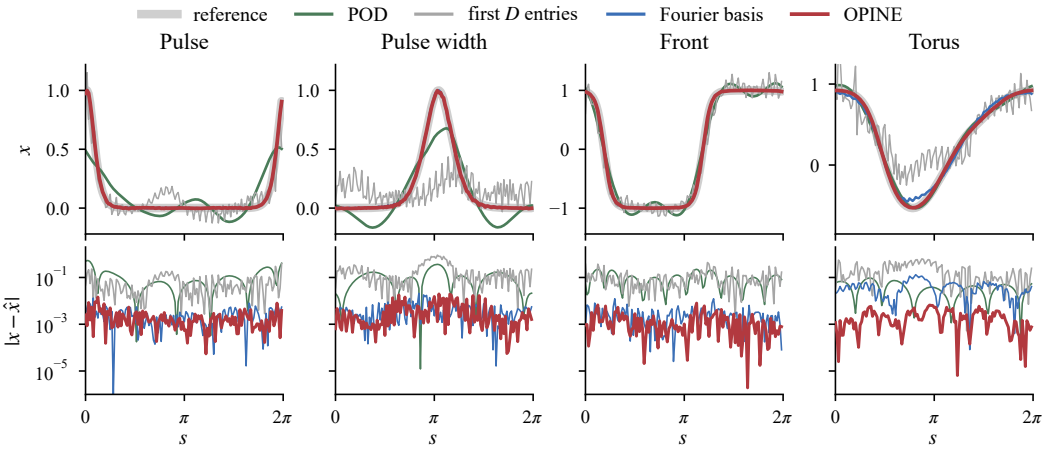

Figure 2: Sample reconstructions for the bottleneck ablation of Table 1 at $D = 4$. Top: a test profile (grey) and its reconstructions; the profile is the test sample with the median OPINE error. Bottom: pointwise error on a logarithmic scale. POD misses the shape of each profile, and keeping the first D entries of z gives noisy profiles. The error of OPINE is about two orders of magnitude below that of POD, and on Front and Torus it is also well below that of the fixed Fourier basis.

domain of length 22 models flame fronts and thin-film flows, and its chaotic attractor lies close to an inertial manifold of dimension about 8 (Linot & Graham, 2020). Forced Burgers turbulence, $u_t + uu_s = \nu u_{ss} + F$, is driven by random forcing of the largest scales that is uncorrelated in time. It produces a statistically stationary field of steep, moving shocks, a standard transport-dominated test case for linear reduction. Each system has eight independent trajectories, which are split by trajectory (Appendix B).

Figure 3 summarizes both systems. The left panels show a test trajectory in space and time: KS forms a slowly drifting cellular pattern, and Burgers a field of sharp fronts. The middle panels show where each method errs at a fixed D, with darker regions marking larger pointwise error. POD has large errors everywhere, the autoencoders have errors concentrated along the cells and shocks, and OPINE's error is small everywhere. The right panels show the test NMSE over D. From $D = 8$ onward, OPINE is at least as accurate as every other method, and its error keeps falling where the autoencoders level off, so that at the largest D it is more than two orders of magnitude below them. On KS, its error drops sharply near $D = 8$, the manifold dimension estimated by Linot & Graham (2020).

4.4 Two-Dimensional Kolmogorov Flow

We now move from one to two space dimensions. Kolmogorov flow is two-dimensional Navier–Stokes flow driven by a sinusoidal body force, $\partial_t \omega + \mathbf{u} \cdot \nabla \omega = \mathrm{Re}^{-1} \nabla^2 \omega - k_f \cos(k_f y)$, on a doubly periodic domain. At $\mathrm{Re} = 100$ with forcing wavenumber $k_f = 4$, the flow is chaotic, with coherent vortices that form, merge, and decay. We reduce vorticity snapshots on a 128×128 grid from 100 independent trajectories, split by trajectory so that test snapshots are never close in time to training snapshots (Appendix C).

Figure 4 shows that OPINE improves on POD at every D, by up to 75%, and that the margin grows with D (Q2). The CNN autoencoder, which has 20 to 300 times more parameters, is more accurate at small D, and OPINE is more accurate at the largest D. The reconstructions on the right show the source of the larger POD error: with the same number of coefficients, POD blurs the vortex cores and the thin filaments between them, which the nonlinear methods retain. Figure 5 decodes single reduced coordinates back to vorticity. The POD and OPINE coordinates come in phase-shifted pairs of the same spatial scale, as expected for a statistically homogeneous direction, and OPINE's are sharper; the CNN latent coordinates are not paired.

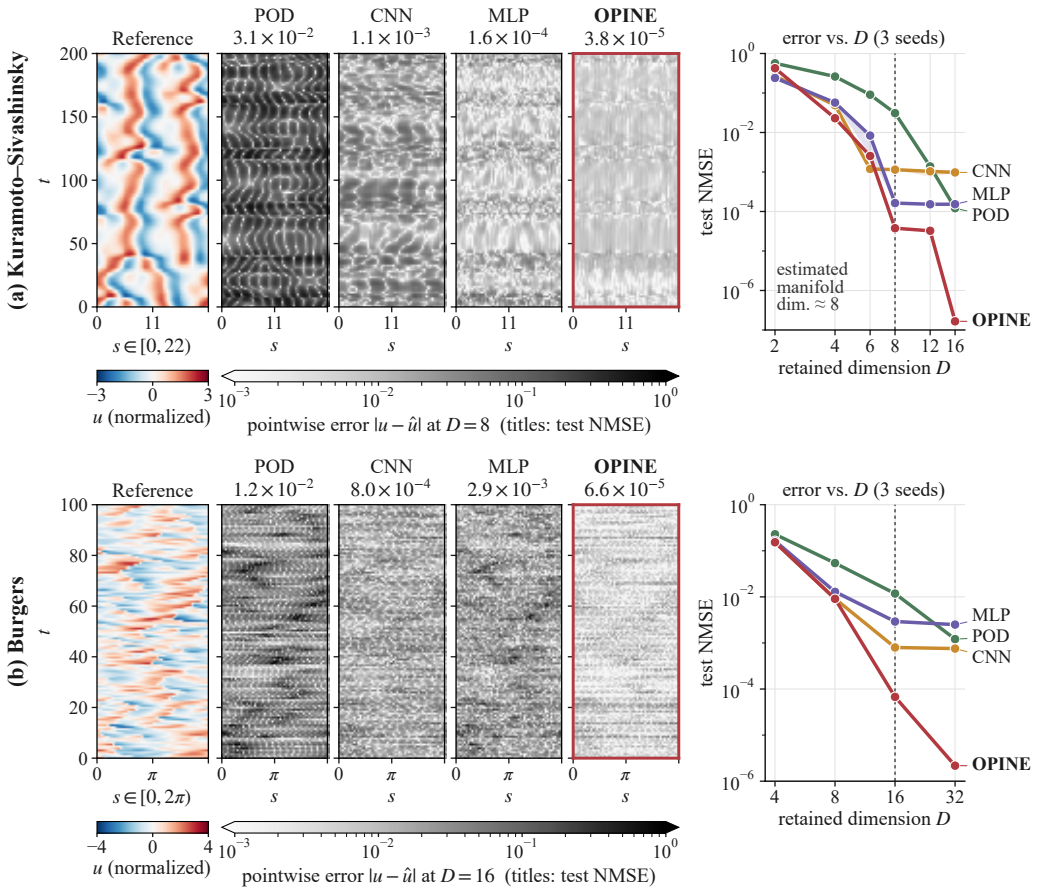

Figure 3: Kuramoto–Sivashinsky (a) and forced Burgers turbulence (b). Left: a test trajectory in space and time. Middle: pointwise error of each method at the D marked by the dashed line on the right, on a shared logarithmic scale (darker is worse); titles give the test NMSE of the run shown. Right: test NMSE versus D (median and range over three seeds).

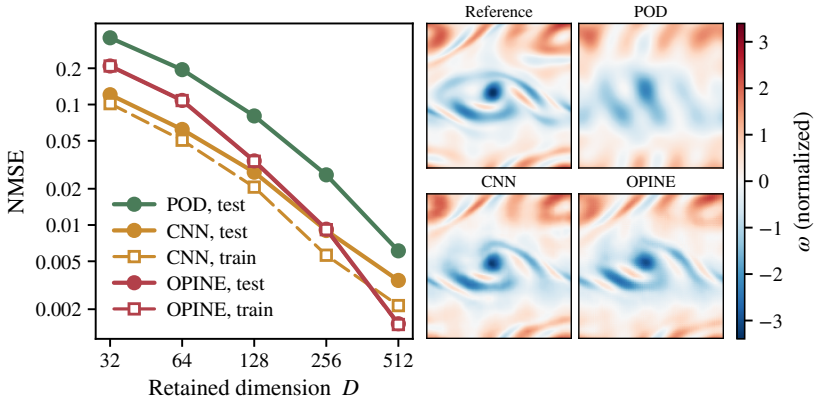

Figure 4: Kolmogorov flow. Left: test NMSE versus D (and train NMSE of the learned models). Right: vorticity of a test snapshot and its reconstructions at $D = 64$; POD blurs the vortex cores and filaments that the CNN and OPINE retain.

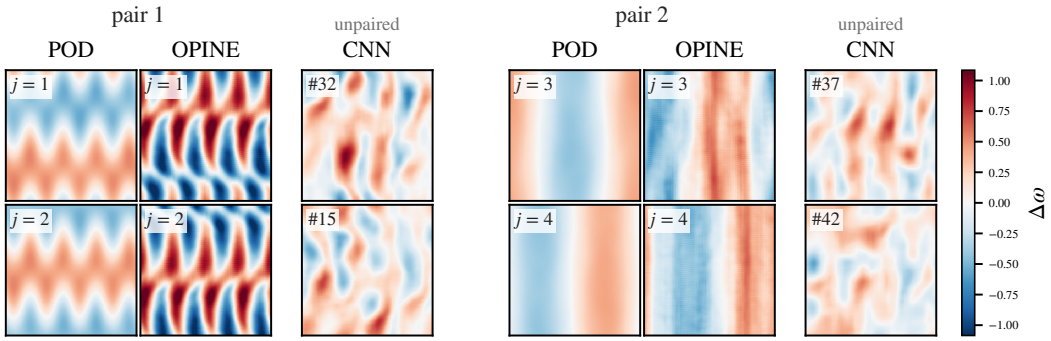

Figure 5: Decoded reduced coordinates for Kolmogorov flow at $D = 64$, each activated at one training standard deviation. POD and OPINE directions form phase-shifted pairs (j: index in variance order), and OPINE's are sharper. CNN latent coordinates have no ordering or pairing, so for each pair we show the two most correlated ones (#: latent index).

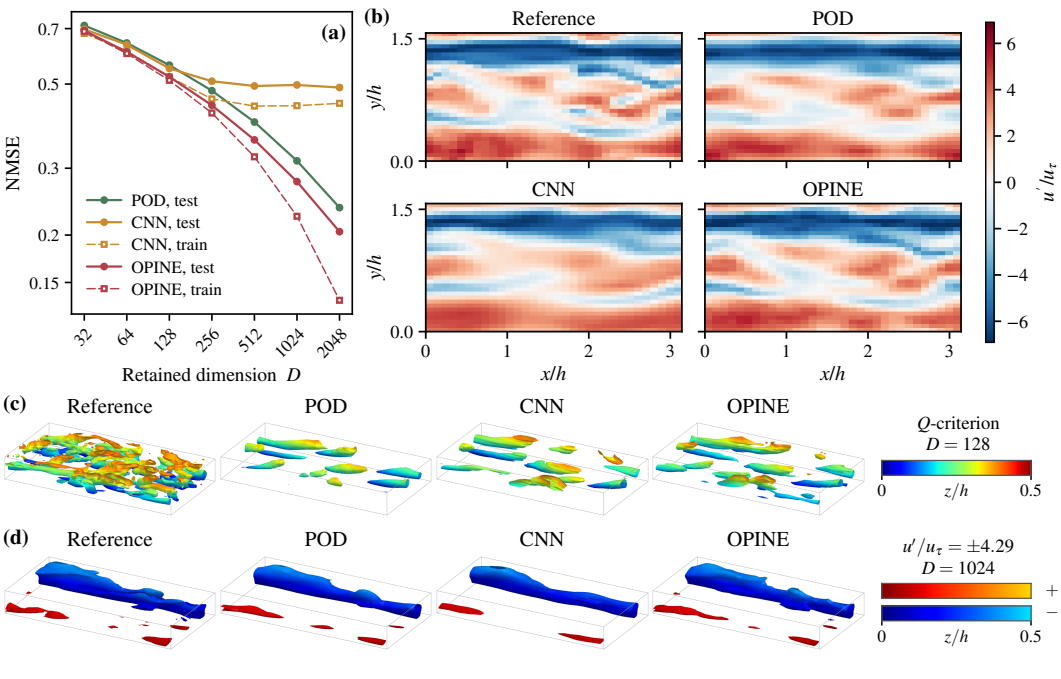

Figure 6: Three-dimensional minimal channel. (a) Test NMSE versus retained dimension D, with the train NMSE of the CNN and OPINE dashed. (b) Streamwise fluctuation in a wall-parallel plane of a test snapshot and its reconstructions at $D = 1024$. (c) Q-criterion isosurfaces at a common isovalue for $D = 128$ in another test snapshot, and (d) streamwise-fluctuation isosurfaces $u'/u_\tau = \pm 4.29$ (u_τ: friction velocity) at $D = 1024$ in the snapshot of (b); color indicates wall-normal position.

4.5 Three-Dimensional Minimal Channel Flow

Finally, we turn to wall-bounded turbulence, which is organized around elongated streaks of slow and fast fluid and the vortices that regenerate them near the wall. We reduce full three-dimensional velocity fields in a minimal channel at friction Reynolds number $Re_\tau = 180$, as in the direct numerical simulation of Kim et al. (1987). This small periodic box, in the spirit of the minimal flow unit of Jiménez & Moin (1991), holds only a few near-wall streaks and the vortices that sustain them. Each

snapshot has about 2×10^5 degrees of freedom, and the $20{,}000$ snapshots are split chronologically, so that the test period follows the training period (Appendix C).

Figure 6a shows the accuracy of each method from $D = 32$ to 2048. OPINE improves on POD at every D, by up to 14%, and its advantage grows with D. The CNN autoencoder follows POD at small D, but from $D = 256$ its training and test errors both level off, whereas OPINE's error keeps decreasing, with 20 to 200 times fewer parameters than the CNN. In a wall-parallel plane (Figure 6b), POD and the CNN smooth the near-wall streaks, whereas OPINE preserves more of their pattern and of the smaller-scale fluctuations.

Figure 6c,d shows the reconstructed vortices and streaks in three dimensions. Isosurfaces of the Q-criterion (Hunt et al., 1988), which mark vortex cores, show that POD recovers only a few smoothed vortices, while the CNN and OPINE recover larger sets. Isosurfaces of the streamwise fluctuation show the low-speed streak that dominates the box: all methods recover it, but OPINE reproduces more of its undulation and of the smaller high-speed regions near the wall.

4.6 Nested Reduced Coordinates

In the comparisons above, OPINE was trained separately at each D. Because its POD basis is ordered, a single model trained at the largest D also gives every smaller D by keeping only the leading modes (Section 3.4; Q4). The prefixes are more accurate than POD at every D on Kolmogorov flow and the minimal channel, and from $D = 3$ on the synthetic manifolds; on the minimal channel they come within 4% of separately trained models from $D = 512$ onward (Table 7; Appendix B). As with POD, one training run thus gives accurate models of every size. For the minimal channel, the prefixes of one model replace seven separately trained models, one for each D from 32 to 2048. As in POD, the accuracy of each prefix in the transformed coordinates is also known without any reconstruction: the mean squared error of the prefix of dimension r on the training data equals $\operatorname{tr} C_z - \sum_{j \le r} (\Lambda_Q)_{jj}$, with Λ_Q from Section 3.4 in decreasing order, so the eigenvalues show how much variance each smaller D retains.

5 Conclusion

This paper makes three contributions. First, OPINE factors nonlinear dimensionality reduction into an invertible change of coordinates and an explicit, ordered POD projection, so that all information loss is confined to the projection and one model gives reduced coordinates of every smaller dimension. Second, its training tracks the POD subspace with a damped, aligned step of orthogonal iteration, which avoids differentiating through an eigendecomposition, provably converges locally to the exact POD subspace, and needs only minibatch products with an $N \times K$ basis. Third, from synthetic manifolds to three-dimensional turbulence, OPINE improves on POD at every retained dimension above one, with far fewer parameters than the CNN autoencoders; on the KS and Burgers dynamics it is more than two orders of magnitude more accurate than autoencoders at the largest retained dimension, and the ablations support the choice of the POD projection. OPINE is not designed to outperform unconstrained autoencoders on every problem; its purpose is an interpretable reduction built only from an invertible change of variables and POD, and it is most beneficial when the states lie near a low-dimensional set that a linear subspace cannot capture. Its coordinates also remain interpretable: decoded one at a time, they form phase-shifted pairs like POD modes but sharper (Figure 5). The same construction, an invertible nonlinear map followed by an ordered POD projection, can be used wherever POD is used, for example for compression, modal analysis, and reduced-order modeling of other physical fields.

Limitations and future work. Training reduced dynamical models in OPINE coordinates is a natural next step beyond the reconstruction studied here; such models will need hyper-reduction because f_θ acts on the full field. The subspace could also be optimized directly on the Grassmann manifold (Edelman et al., 1998; Absil et al., 2008) instead of by damped orthogonal iteration. Projections constrained by symmetry, with POD performed within Fourier blocks of a homogeneous direction, are a further extension that we leave for future work. The analysis of the subspace update is local and does not cover the joint training of the map and the basis (Appendix A), and the flow datasets are simulations at moderate Reynolds numbers.

AI USE STATEMENT

Generative AI tools were used to assist with editing and polishing the manuscript, including improving clarity, grammar, and presentation, and to help draw parts of the schematics. The analysis of the subspace update in Appendix A, including the statements and proofs of Theorem 1 and Proposition 2, was drafted mainly with the help of generative AI; the authors checked every step of the derivations. All AI-assisted content was reviewed and verified by the authors, who take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

Section 3 specifies the model and the training algorithm, and Appendix A states the assumptions of the analysis. Appendices B and C give the architectures, optimizer settings, data generation, preprocessing, and data splits of every experiment. Code and data-generation scripts will be released upon publication.

REFERENCES

P.-A. Absil, Robert Mahony, and Rodolphe Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2008.

Jens Behrmann, Will Grathwohl, Ricky T. Q. Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pp. 573–582, 2019.

Artur Bekasov and Iain Murray. Ordering dimensions with nested dropout normalizing flows. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*, 2020.

Gal Berkooz, Philip Holmes, and John L. Lumley. The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Review of Fluid Mechanics*, 25:539–575, 1993. doi: 10.1146/annurev.fl.25.010193.002543.

Johann Brehmer and Kyle Cranmer. Flows for simultaneous manifold learning and density estimation. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Anthony L. Caterini, Gabriel Loaiza-Ganem, Geoff Pleiss, and John P. Cunningham. Rectangular flows for manifold learning. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Kathleen Champion, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116 (45):22445–22451, 2019.

Ricky T. Q. Chen, Jens Behrmann, David K. Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Edmond Cunningham, Adam D. Cobb, and Susmit Jha. Principal component flows. In *International Conference on Machine Learning*, 2022.

Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *International Conference on Learning Representations*, 2017.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Alan Edelman, Tomás A. Arias, and Steven T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM Journal on Matrix Analysis and Applications*, 20(2):303–353, 1998.

Hamidreza Eivazi, Soledad Le Clainche, Sergio Hoyas, and Ricardo Vinuesa. Towards extraction of orthogonal and parsimonious non-linear modes from turbulent flows. *Expert Systems with Applications*, 202:117038, 2022.

Ky Fan. On a theorem of Weyl concerning eigenvalues of linear transformations I. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.

Stefania Fresca and Andrea Manzoni. POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 388:114181, 2022. doi: 10.1016/j.cma.2021.114181.

Kai Fukami, Taichi Nakamura, and Koji Fukagata. Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data. *Physics of Fluids*, 32(9): 095110, 2020. doi: 10.1063/5.0020721.

Rudy Geelen, Stephen Wright, and Karen Willcox. Operator inference for non-intrusive model reduction with quadratic manifolds. *Computer Methods in Applied Mechanics and Engineering*, 403:115717, 2023.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 4th edition, 2013.

Constantin Greif and Karsten Urban. Decay of the Kolmogorov n-width for wave problems. *Applied Mathematics Letters*, 96:216–222, 2019. doi: 10.1016/j.aml.2019.05.013.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Philip Holmes, John L. Lumley, Gal Berkooz, and Clarence W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, 2nd edition, 2012.

Chin-Wei Huang, David Krueger, Alexandre Lacoste, and Aaron Courville. Neural autoregressive flows. In *International Conference on Machine Learning*, pp. 2078–2087, 2018.

Julian C. R. Hunt, Alan A. Wray, and Parviz Moin. Eddies, streams, and convergence zones in turbulent flows. In *Proceedings of the Summer Program, Center for Turbulence Research*, pp. 193–208, 1988. CTR-S88.

Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *IEEE International Conference on Computer Vision*, 2015.

Javier Jiménez and Parviz Moin. The minimal flow unit in near-wall turbulence. *Journal of Fluid Mechanics*, 225:213–240, 1991. doi: 10.1017/S0022112091002033.

Aly-Khan Kassam and Lloyd N. Trefethen. Fourth-order time-stepping for stiff PDEs. *SIAM Journal on Scientific Computing*, 26(4):1214–1233, 2005.

John Kim, Parviz Moin, and Robert Moser. Turbulence statistics in fully developed channel flow at low reynolds number. *Journal of Fluid Mechanics*, 177:133–166, 1987. doi: 10.1017/S0022112087000892.

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Dmitrii Kochkov, Jamie A. Smith, Ayya Alieva, Qing Wang, Michael P. Brenner, and Stephan Hoyer. Machine learning-accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021. doi: 10.1073/pnas.2101784118.

Konik Kothari, AmirEhsan Khorashadizadeh, Maarten de Hoop, and Ivan Dokmanić. Trumpets: Injective flows for inference and inverse problems. In *Conference on Uncertainty in Artificial Intelligence*, 2021.

Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243, 1991. doi: 10.1002/aic.690370209.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Kookjin Lee and Kevin T. Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020. doi: 10.1016/j.jcp.2019.108973.

Shen Li and Bryan Hooi. Neural PCA for flow-based representation learning. In *International Joint Conference on Artificial Intelligence*, pp. 3229–3235, 2022. doi: 10.24963/ijcai.2022/448.

Alec J. Linot and Michael D. Graham. Deep learning to discover and predict dynamics on an inertial manifold. *Physical Review E*, 101:062209, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

John L. Lumley. *Stochastic Tools in Turbulence*. Academic Press, 1970.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9:4950, 2018.

Romit Maulik, Bethany Lusch, and Prasanna Balaprakash. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Physics of Fluids*, 33(3):037106, 2021.

Sebastian Mika, Bernhard Schölkopf, Alex Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel PCA and de-noising in feature spaces. In *Advances in Neural Information Processing Systems*, volume 11, 1999.

Michele Milano and Petros Koumoutsakos. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, 182(1):1–26, 2002.

Thomas Müller, Brian McWilliams, Fabrice Rousselle, Markus Gross, and Jan Novák. Neural importance sampling. *ACM Transactions on Graphics*, 38(5):145:1–145:19, 2019.

Takaaki Murata, Kai Fukami, and Koji Fukagata. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *Journal of Fluid Mechanics*, 882:A13, 2020. doi: 10.1017/jfm.2019.822.

Erkki Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982. doi: 10.1007/BF00275687.

Julius Reiss, Philipp Schulze, Jörn Sesterhenn, and Volker Mehrmann. The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena. *SIAM Journal on Scientific Computing*, 40(3):A1322–A1344, 2018.

Oren Rippel, Michael A. Gelbart, and Ryan P. Adams. Learning ordered representations with nested dropout. In *International Conference on Machine Learning*, pp. 1746–1754, 2014.

Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

Tommaso Taddei. A registration method for model order reduction: Data compression and geometry reduction. *SIAM Journal on Scientific Computing*, 42(2):A997–A1027, 2020.

Kunihiko Taira, Steven L. Brunton, Scott T. M. Dawson, Clarence W. Rowley, Tim Colonius, Beverley J. McKeon, Oliver T. Schmidt, Stanislav Gordeyev, Vassilios Theofilis, and Lawrence S. Ukeiley. Modal analysis of fluid flows: An overview. *AIAA Journal*, 55(12):4013–4041, 2017.

Wei Wang, Zheng Dang, Yinlin Hu, Pascal Fua, and Mathieu Salzmann. Backpropagation-friendly eigendecomposition. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Bin Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43(1):95–107, 1995. doi: 10.1109/78.365290.