

# MODEL REDUCTION FOR NONLINEAR SYSTEMS BY BALANCED TRUNCATION OF STATE AND GRADIENT COVARIANCE\*

SAMUEL E. OTTO<sup>†</sup>, ALBERTO PADOVAN<sup>†</sup>, AND CLARENCE W. ROWLEY<sup>†</sup>

**Abstract.** Data-driven reduced-order models often fail to make accurate forecasts of high-dimensional nonlinear dynamical systems that are sensitive along coordinates with low-variance because such coordinates are often truncated, e.g., by proper orthogonal decomposition, kernel principal component analysis, and autoencoders. Such systems are encountered frequently in shear-dominated fluid flows where nonnormality plays a significant role in the growth of disturbances. In order to address these issues, we employ ideas from active subspaces to find low-dimensional systems of coordinates for model reduction that balance adjoint-based information about the system’s sensitivity with the variance of states along trajectories. The resulting method, which we refer to as covariance balancing reduction using adjoint snapshots (CoBRAS), is analogous to balanced truncation with state and adjoint-based gradient covariance matrices replacing the system Gramians and obeying the same key transformation laws. Here, the extracted coordinates are associated with an oblique projection that can be used to construct Petrov–Galerkin reduced-order models. We provide an efficient snapshot-based computational method analogous to balanced proper orthogonal decomposition. This also leads to the observation that the reduced coordinates can be computed relying on inner products of state and gradient samples alone, allowing us to find rich nonlinear coordinates by replacing the inner product with a kernel function. In these coordinates, reduced-order models can be learned using regression. We demonstrate these techniques and compare to a variety of other methods on a simple, yet challenging three-dimensional system and a nonlinear axisymmetric jet flow simulation with  $10^5$  state variables.

**Key words.** data-driven modeling, active subspaces, balanced truncation, kernel method, adjoint method, method of snapshots, nonnormal systems, oblique projection, Grassmann manifold

**MSC codes.** 14M15, 15A03, 15A24, 15A42, 34A45, 34C20, 47B32, 57R35, 57R40, 76D55, 90C06, 93A15, 93C10

**DOI.** 10.1137/22M1513228

**1. Introduction.** Models that describe the time-evolution of physical systems such as fluid flows are key to forecasting, tracking, and controlling their behavior. While the governing equations for these systems, e.g., the Navier–Stokes equations, can often be simulated, these simulations are too costly to be used in real-time applications. Instead, a simplified reduced-order model (ROM) can be used to describe the most important aspects of the original system’s behavior. Methods for constructing ROMs from the original full-order model (FOM) entail finding a suitable projection that reduces the dimension of the state space without neglecting important information. Many different approaches are possible depending on what state information is deemed important. For reviews of modern methods, see [20, 51, 8].

The simplest, and by far the most common, approach is to say that a “feature” or state-space coordinate contains important information about the system if it explains

\*Submitted to the journal’s Methods and Algorithms for Scientific Computing section August 2, 2022; accepted for publication (in revised form) May 16, 2023; published electronically September 21, 2023.

<https://doi.org/10.1137/22M1513228>

**Funding:** This research was supported by the Army Research Office under grant W911NF-17-1-0512 and the Air Force Office of Scientific Research under grant FA9550-19-1-0005. The first author was supported by the National Science Foundation Graduate Research Fellowship Program under grant DGE-2039656.

<sup>†</sup>Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544 USA (sotto@princeton.edu, apadovan@princeton.edu, cwwrowley@princeton.edu).

a large amount of variance among the states generated along typical trajectories. The optimal linear features in this respect are given by principal component analysis (PCA) also known in some communities as proper orthogonal decomposition (POD). The first application of POD to model reduction of fluid flows was by Lumley [37], with later work by Sirovich [59] and Holmes et al. [23]. As noted by Sirovich [59], the PCA/POD basis vectors or “modes” can be computed using the “method of snapshots” without assembling the state covariance matrix. The resulting coordinates of a state vector projected onto the leading modes can be found using only inner products between the given state vector and the original data. Schölkopf [56] used this observation to develop kernel PCA (KPCA), which extracts rich nonlinear features by applying PCA/POD after lifting data into a higher-dimensional (possibly infinite-dimensional) reproducing kernel Hilbert space (RKHS). Nonlinear features that are optimal for reconstructing the system’s states can also be found using autoencoder neural networks and employed for model reduction as described by Lee and Carlberg [36].

Nonnormality in shear-dominated fluid flows causes the evolution of these systems to be heavily influenced by certain low-energy (low-variance) disturbances [61, 55]. Therefore, the features (coordinates) we choose to construct a ROM must account for both the variance of the system’s states as well as the sensitivity of the system’s dynamics. In the context of a linear time-invariant input-output system these quantities are captured by the controllability and observability Gramians. Thanks to their transformation properties, these Gramians may be simultaneously diagonalized by a change of coordinates, leading to the balanced truncation (BT) algorithm introduced by Moore [42]. Rowley [50] discovered that the oblique projection and features extracted by BT can be approximated using a method of snapshots called balanced POD (BPOD) involving data obtained by impulse response simulations of the system and its adjoint. This enabled BT to be successfully applied to enormous systems including linearized fluid simulations where it would be computationally impractical to compute the Gramians [4, 2, 26, 27]. Analogous approaches can be applied to linear time-periodic input-output systems, where reduced-order models are obtained by balancing the time-periodic Gramians [53, 39, 38, 45].

Moving away from linear systems, we can find several generalizations of balanced truncation for quadratic bilinear (QB) systems [6], as well as to nonlinear systems [54, 62, 33]. However, there is room for improvement on these methods. As we demonstrate in [44], QB balancing (QB BT) as well as the QB iterative rational Krylov algorithm (QB-IRKA) [7] lose accuracy away from the stable equilibrium about which they are based in addition to being extremely difficult to compute for large-scale systems, e.g., with  $10^5$  or more states. The nonlinear balancing methods introduced in [54, 62] are computationally challenging owing to the need to solve PDEs over the state space. Progress has been made on this issue using Taylor series approximation [19, 31, 32], but the computations remain daunting for systems with more than  $10^3$  states. The method introduced in [33] relies on empirical Gramians obtained via nonlinear impulse response simulations. However, the choices for the impulses are ambiguous [26, 25], and the number of impulses required scales linearly with the state dimension [28]. This scaling with the state dimension is unavoidable since the range of the empirical observability Gramian, measuring the system’s sensitivity, lies in the span of the sampled initial conditions. A generalization of the empirical Gramians framework is developed in [12, 13], where state and output trajectory data are lifted into an RKHS, yielding a nonlinear dimension reduction map that can be computed using the kernel function. This approach still relies on sampled trajectories for each state dimension of the FOM.

Sampling a function's gradient and applying POD to these samples reveals an "active subspace" [16] in which most of the function's variation is captured. Because the number of samples can be much smaller than the dimension of the function's domain, active subspaces have become a valuable tool for parameter studies and design optimization. The generalization introduced by Zahm et al. [66] identifies a projection that balances the sensitivity of an arbitrary function against the variance of an underlying probability distribution of points in the domain. In the present work, we apply this approach to reduce the dimension of dynamical systems by considering the function that maps initial states to sequences of outputs. We identify a projection that balances the sensitivity of this function with the variance of state space data collected from the system along typical trajectories. The gradient samples are computed using an adjoint sensitivity method applied to random projections of output sequences from the system. The resulting method, which we call covariance balancing reduction using adjoint snapshots (CoBRAS), is demonstrated on several challenging problems.

Our key technical observation is that the method introduced by Zahm et al. [66] is identical to BT with the Gramians replaced by state and gradient covariance matrices sharing the same transformation properties. Immediately, this yields a new method of snapshots resembling BPOD that allows the projection to be computed for high-dimensional systems where assembling the covariance matrices is prohibitive. This projection is optimal with respect to reconstruction of the state and gradient data in the sense described by Singler [57, 58]. Moreover, the features extracted by the projection depend only on inner products among state and gradient vectors. This allows us to replace the inner product with a kernel function in order to identify rich nonlinear features in a higher-dimensional RKHS, as in [56]. While related work by [49] introduces the idea of lifting into a high-dimensional space using a nonlinear feature map, the underlying reliance on inner products enabling implicit computations via a kernel is not recognized. Another related approach [10] optimizes nonlinear features based on gradient alignment. In contrast, our approach does not require optimization and relies only on the singular value decomposition (SVD) of a kernel matrix whose size is independent of the state dimension. Our approach also differs from the kernel-based generalization of the empirical Gramians framework described in [12, 13] since we rely on gradient samples obtained from the system's linearized adjoint and an appropriate lifting of these gradients into the RKHS in order to avoid the costly sampling of a trajectory for each state variable of the FOM.

We demonstrate the proposed method on a challenging three-dimensional nonlinear toy model, and on an incompressible axisymmetric jet flow simulation at Reynolds number  $Re = 2000$ . We will see that, not only does CoBRAS perform well as a stand-alone model reduction technique, but it can also be used to provide a good initial guess for the recently introduced trajectory-based optimization for oblique projections (TrOOP) framework [44].

**2. Balancing variance and sensitivity.** Consider a discrete-time dynamical system referred to as the FOM

$$(2.1) \quad \begin{aligned} x(t+1) &= f(x(t), u(t)), & x(0) &= x_0 \\ y(t) &= g(x(t), u(t)), \end{aligned}$$

with state vector  $x \in \mathbb{R}^n$ , input  $u \in \mathbb{R}^{q_0}$ , and output  $y \in \mathbb{R}^{m_0}$ . We seek to approximate the output of this system over a range of initial conditions and input signals using a ROM

$$(2.2) \quad \begin{aligned} z(t+1) &= \tilde{f}(z(t), u(t)), & z(0) &= h(x_0) \\ \hat{y}(t) &= \tilde{g}(z(t), u(t)), \end{aligned}$$

whose state  $z$  evolves in a lower-dimensional space  $\mathbb{R}^r$  with  $r < n$ . The ROM may be constructed by first selecting a map  $h: \mathbb{R}^n \rightarrow \mathbb{R}^r$  and then employing one of several possible methods to find the approximate dynamics  $\tilde{f}$  and observation map  $\tilde{g}$  that govern the encoded state variables  $z = h(x)$ . These methods include (nonlinear) Galerkin methods as well as regression-based approaches and recurrent neural networks. Regardless of the approximation technique used for  $\tilde{f}$  and  $\tilde{g}$ , the quality of the ROM depends heavily on the features selected by  $h$ . Our goal will be to choose the map  $h$  using simple data-driven methods that enable construction of accurate ROMs. The map we identify will be associated with a projection operator that can be used to construct Petrov–Galerkin ROMs.

The features extracted by the encoding map  $h$  must contain the information we need to forecast the future outputs of the system. In this paper, we formalize this notion as follows. Evolving the system over a time-horizon  $L$  with a sequence of inputs  $u(0:L-1) = (u(0), \dots, u(L-1)) \in \mathbb{R}^q$  produces outputs  $y(0:L) = (y(0), y(1), \dots, y(L)) \in \mathbb{R}^m$ , giving rise to a map

$$(2.3) \quad F = (F_0, \dots, F_L) : (x(0), u(0:L-1)) \mapsto y(0:L)$$

with components  $F_\tau : (x(0), u(0:\tau-1)) \mapsto y(\tau)$ . In the present context, we seek an  $h$  that allows for close approximation of  $F(x(t), u(t:t+L-1))$  by a function  $\tilde{F}(h(x(t)), u(t:t+L-1))$  along trajectories of the system.

Specifically, we consider approximating  $F$  over a probability distribution of states  $x$  and input sequences  $\bar{u}$  constructed as follows. We begin with a distribution over initial conditions  $x_0$  and input sequences  $u(0:N+L-1)$  determined by the problem of interest. From the resulting trajectories of (2.1), we draw  $t$  uniformly at random from  $\{0, \dots, N\}$  to construct  $x = x(t)$  and  $\bar{u} = u(t:t+L-1)$ .

**2.1. Extracting linear features.** We first consider the case where  $h(x) = \Psi^T x$  is a linear map providing coordinates for the range of a linear projection  $P = \Phi \Psi^T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ . The projection uniquely decomposes any  $x \in \mathbb{R}^n$  into  $x = x_1 + x_2$ , where  $x_1 \in \text{Range}(P)$  and  $x_2 \in \text{Null}(P) = \text{Null}(h)$ . Intuitively speaking, a good projection for approximating  $F$  will be one where the variance of  $x_2$  is small and the sensitivity of  $F$  to variation of  $x_2$  is also small. The problem of selecting a suitable projection based on these criteria for a general vector valued function  $F$  is studied by Zahm et al. [66]. In this approach, one starts with a probability distribution over states and quantifies the variation of the state  $x$  using the covariance matrix

$$(2.4) \quad W_x = \mathbb{E}[xx^T].$$

Here, we apply this idea to a dynamical system (2.1) by considering the probability distribution constructed above for states and input sequences along trajectories.

*Remark 2.1.* We center (2.4) about the origin rather than the mean as one may wish to center about an arbitrary point in state space via a coordinate shift; for example, it is common to center about an equilibrium of the system (2.1).

Following [66], the sensitivity of  $F$  may be quantified using the gradient covariance matrix

$$(2.5) \quad W_g = \mathbb{E}[\nabla_x F(x, \bar{u}) \nabla_x F(x, \bar{u})^T],$$

where  $\nabla_x F(x, \bar{u}) = D_x F(x, \bar{u})^T$ . The optimal approximating function in the mean-square sense is given by the conditional expectation

$$(2.6) \quad \hat{F}(Px, \bar{u}) = \mathbb{E} [F(x, \bar{u}) \mid Px],$$

which averages  $F(x_1 + x_2, \bar{u})$  over  $x_2$ , where  $x = x_1 + x_2$  is the decomposition induced by  $P$ . To understand the relationship between the covariance matrices and the approximation accuracy of (2.6), we consider the case where the conditional distribution of  $x$  given  $\bar{u}$  is Gaussian. In this setting, Theorem 2.2 provides an explicit bound.

**THEOREM 2.2.** *Let  $(x, \bar{u})$  have distribution  $\mu$  on  $\mathbb{R}^n \times \mathbb{R}^q$  such that the conditional distribution of  $x$  given  $\bar{u}$  is almost surely Gaussian with positive-definite covariance  $\Sigma_{x|\bar{u}} = \mathbb{E}[(x - \mathbb{E}[x|\bar{u}])(x - \mathbb{E}[x|\bar{u}])^T \mid \bar{u}]$ . Let  $\Sigma_x = \mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^T]$  denote the marginal covariance. Suppose that there is a constant  $C \geq 0$  so that*

$$(2.7) \quad C\Sigma_x - \Sigma_{x|\bar{u}} \succeq 0$$

*is positive semidefinite almost surely. This holds with  $C = 1$  when  $(x, \bar{u})$  are jointly Gaussian. Let  $F \in L^2(\mathbb{R}^n \times \mathbb{R}^q, \mu; \mathbb{R}^m)$  have continuous partial derivatives, and let  $P: \mathbb{R}^n \rightarrow \mathbb{R}^n$  be a rank- $r$  linear projection. Then the mean square approximation error of (2.6) is bounded by*

$$(2.8) \quad \mathbb{E} [\|F(x, \bar{u}) - \hat{F}(Px, \bar{u})\|^2] \leq C \operatorname{Tr} [W_g(I - P)W_x(I - P)^T],$$

where  $W_x$  and  $W_g$  are given by (2.4) and (2.5).

*Proof.* The core of this result is Proposition 2.5 in Zahm et al. [66], which, in turn, is derived from the Gaussian Poincaré inequality in Chen [15]. Since the rest of the proof is not especially instructive, we give it in Appendix A.  $\square$

Similar bounds likely hold for much larger classes of non-Gaussian distributions due to the results in [46, 67]. We also note that the upper bound on the right-hand side of (2.8) can be re-written as

$$(2.9) \quad \operatorname{Tr} [W_g(I - P)W_x(I - P)^T] = \mathbb{E} [\|W_g^{1/2}(x - Px)\|^2].$$

This can be interpreted as the mean square gradient-weighted difference between states and their projections.

In [66], a projection operator that minimizes the upper bound (2.8) is found by solving a generalized eigenvalue problem  $W_g V = W_x^{-1} V \Lambda$  and constructing  $P = V_r V_r^T W_x^{-1}$  where the columns of  $V_r$  are the  $r$  eigenvectors with largest eigenvalues (see Proposition 2.6 in [66]). The covariance matrices are approximated via Monte-Carlo sampling. However, as the dimension  $n$  becomes very large, it becomes computationally impractical to assemble the covariance matrices and solve the eigenvalue problem. Moreover, as the output dimension  $m$  becomes large, it becomes impractical to compute  $\nabla_x F$ .

We address these issues using the observation that the projection found in [66] is identical to the one found by applying BT [42] to the state and gradient covariance matrices instead of the controllability and observability Gramians of a linear system. The BT algorithm has been extensively studied and admits computationally efficient snapshot-based approximations in the case of large  $n$  and  $m$  using the BPOD algorithm introduced by Rowley [50]. The key observation is Theorem 2.3, which yields a factorized optimal projection using SVD and factors of the covariance matrices. This result is closely related to the optimality of BPOD for data reconstruction shown by Singler in [57].

THEOREM 2.3 (factorized covariance balancing). *Let  $W_x = XX^T$  and  $W_g = YY^T$  and form the SVD*

$$(2.10) \quad Y^T X = U \Sigma V^T, \quad \Sigma = \text{diag}(\sigma_1, \dots, \sigma_n),$$

*with its rank- $r$  truncation denoted by  $U_r \Sigma_r V_r^T$ . If  $\sigma_r > 0$ , then the minimum*

$$(2.11) \quad \min_{\substack{P \in \mathbb{R}^{n \times n}, \\ P^2 = P, \text{rank}(P) = r}} \text{Tr} \left[ W_g (I - P) W_x (I - P)^T \right] = \sigma_{r+1}^2 + \dots + \sigma_n^2$$

*is achieved by*

$$(2.12) \quad \boxed{P = \Phi \Psi^T, \quad \text{where} \quad \Phi = X V_r \Sigma_r^{-1/2} \quad \text{and} \quad \Psi = Y U_r \Sigma_r^{-1/2}.}$$

*Proof.* Thanks to the permutation identity for the trace and the definition of the Frobenius norm, we have

$$(2.13) \quad \text{Tr} \left[ W_g (I - P) W_x (I - P)^T \right] = \|Y^T X - Y^T P X\|_F^2.$$

The matrix  $Y^T P X$  has rank at most  $r$ , and so the Eckart–Young–Mirsky theorem [18, 41] gives

$$(2.14) \quad \|Y^T X - Y^T P X\|_F^2 \geq \|Y^T X - U_r \Sigma_r V_r^T\|_F^2 = \sigma_{r+1}^2 + \dots + \sigma_n^2.$$

Direct substitution shows that equality is obtained with  $P$  defined by (2.12).  $\square$

To clarify the connection with balanced truncation, we observe that the state and gradient covariance matrices obey the same transformation laws as the controllability and observability Gramians under linear changes of coordinates  $x = T\tilde{x}$ . The function to be approximated becomes  $F(x) = F(T\tilde{x}) =: \tilde{F}(\tilde{x})$  and its derivative transforms according to  $DF(x) = D\tilde{F}(\tilde{x})T^{-1}$ . Thus, the state and gradient covariance matrices transform according to

$$(2.15) \quad W_x = T \tilde{W}_x T^T \quad \text{and} \quad W_g = T^{-T} \tilde{W}_g T^{-1},$$

where  $W_g$  is computed using  $F$  and  $\tilde{W}_g$  is computed using  $\tilde{F}$ . Thanks to this transformation law, when  $W_x$  and  $W_g$  are positive definite it is possible to find a “balancing” transformation  $T = X V \Sigma^{-1/2}$ ,  $T^{-1} = \Sigma^{-1/2} U^T Y^T$  so that  $\tilde{W}_x = \tilde{W}_g = \Sigma$  are equal and diagonal [34, 42]. It is easy to see that the projector described by (2.12) corresponds to the truncation operator

$$(2.16) \quad T^{-1} P T = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}$$

in the coordinate system where the covariance matrices are balanced. While a balancing transformation requires the covariance matrices to be positive definite, the optimal projection described by Theorem 2.3 does not. It may be computed using low-rank factors  $X$  and  $Y$ .

Following a similar construction to BPOD [50], factors  $X$  and  $Y$  of the sample-based covariance matrices can be formed using the Monte-Carlo samples of the state and gradient. Letting  $\{x_1, \dots, x_{s_x}\}$  be samples of the state, then the sample-based state covariance can be factored using

$$(2.17) \quad X = \frac{1}{\sqrt{s_x}} \begin{bmatrix} x_1 & \dots & x_{s_x} \end{bmatrix}.$$

In order to reduce the computational burden associated with sampling the gradient, we use an independent, zero-mean, isotropic random vector  $\xi \in \mathbb{R}^m$ , i.e.,  $\mathbb{E}[\xi\xi^T] = I$  to define the univariate gradient

$$(2.18) \quad g = \nabla_x(\xi^T F)(x, \bar{u})$$

satisfying  $\mathbb{E}[gg^T] = W_g$ . We can then factor the sample-based gradient covariance matrix using samples of the univariate gradient arranged into the columns of

$$(2.19) \quad Y = \frac{1}{\sqrt{s_g}} \begin{bmatrix} g_1 & \cdots & g_{s_g} \end{bmatrix},$$

where the columns  $g_i$  are drawn using the joint distribution of  $\xi$  and  $x$  (which are independent). Using these factors and Theorem 2.3, we can compute the desired linear features using

$$(2.20) \quad \boxed{z = h(x) = \Psi^T x = \Sigma_r^{-1/2} U_r^T Y^T x.}$$

This computation does not use the  $n \times n$  covariance matrices, and may be performed using inner products between sampled state and gradient vectors in  $\mathbb{R}^n$ . In particular, assembling  $Y^T X$  and computing its SVD has time complexity  $\mathcal{O}(s_x s_g n + s_x s_g \min\{s_x, s_g\})$ . Once this is done, evaluating (2.20) has time complexity  $\mathcal{O}(s_g n + s_g r)$ .

**2.2. The need for gradient sampling.** In this section we discuss why it is important to sample the gradient of  $F$  rather than relying solely on samples of its values  $F(x_i)$  at points  $x_i \in \mathbb{R}^n$ ,  $i = 1, \dots, s$ . Consider the simplest case when  $F$  is a linear map with rank  $r$ . The essential issue is that if one relies solely on the sampled values  $F(x_i)$ , then generically, one cannot learn about the  $r$ -dimensional subspace on which  $F$  is the most sensitive (i.e., the range of  $F^T$ ) without sampling in all  $n$  directions. However, the subspace  $\text{Null}(F)^\perp = \text{Range}(F^T) = \text{Range}(W_g)$  is quickly spanned using  $r$  samples of the gradient  $\nabla(\xi_i^T F)(x_i) = F^T \xi_i$  for almost every  $\xi_1, \dots, \xi_r$  (with respect to Lebesgue measure). This well-known result is an immediate consequence of the following lemma.

**LEMMA 2.4.** *Let  $M \in \mathbb{R}^{m \times n}$  be a matrix with  $\text{rank}(M) = r$ . If  $r \leq s \leq n$ , then almost every matrix  $X \in \mathbb{R}^{n \times s}$  (with respect to Lebesgue measure) has linearly independent columns and satisfies  $\text{rank}(MX) = r$ .*

*Proof.* See Appendix B for the proof.  $\square$

Thus, if columns of  $X$  are the vectors  $\xi_1, \dots, \xi_r$ , the  $r$ -dimensional range of  $M = F^T$  is spanned by  $F^T \xi_1, \dots, F^T \xi_r$ . Related approximation results based on random sampling for matrices with decaying singular values, but possibly full rank, can be found in [21].

On the other hand,  $\text{Range}(F^T)$  cannot be uniquely determined from the pairs  $(x_i, F(x_i))$  until the number of samples  $s$  is at least as large as  $n - r$ . When  $s < n - r$  there is always a map  $\tilde{F}$  with  $\tilde{F}x_i = Fx_i$  for every  $i = 1, \dots, s$ , but with  $\text{Range}(\tilde{F}^T) \neq \text{Range}(F^T)$ . To see this, when  $s < n - r$  we cannot have  $\text{Null}(F) \subset \text{Range}(X)$  since their dimensions are incompatible. Considering the orthogonal complements, we cannot have  $\text{Range}(X)^\perp \subset \text{Range}(F^T)$ . Given a (reduced) SVD  $F = U\Sigma V^T$ , adding a nonzero vector  $v \in \text{Range}(X)^\perp$  with  $v \notin \text{Range}(F^T)$  to the first row of  $V^T$  produces  $\tilde{F} = U\Sigma(V^T + e_1 v^T)$  with the stated properties. To quantify the degree

of “nonuniqueness” for  $\text{Range}(F^T)$  given the samples, we recall that  $r$ -dimensional subspaces of  $\mathbb{R}^n$  belong to the Grassmann manifold  $\mathcal{G}_{n,r}$  [65, 1, 5]. For generic samples forming the columns of an  $n \times s$  matrix  $X$ , the following theorem characterizes the possible subspaces  $\text{Range}(\tilde{F}^T)$  as a submanifold of  $\mathcal{G}_{n,r}$  with dimension  $r(n-s)$ .

**THEOREM 2.5.** *Let  $F : \mathbb{R}^{m \times n}$  be a matrix with  $\text{rank}(F) = r$ , and let  $X \in \mathbb{R}^{n \times s}$  be a matrix with linearly independent columns satisfying  $\text{rank}(FX) = r$ . Then*

$$(2.21) \quad \mathcal{V}_{F,X} = \left\{ \text{Range}(\tilde{F}^T) : \tilde{F} \in \mathbb{R}^{m \times n}, \text{rank}(\tilde{F}) = r, \text{ and } \tilde{F}X = FX \right\}$$

*is an  $r(n-s)$ -dimensional submanifold of  $\mathcal{G}_{n,r}$  diffeomorphic to  $\mathbb{R}^{(n-s) \times r}$ . The diameter of  $\mathcal{V}_{F,X}$  in  $\mathcal{G}_{n,r}$  is bounded by*

$$(2.22) \quad \frac{\pi}{2} \sqrt{\min\{r, n-s\}} \leq \sup_{V,W \in \mathcal{V}_{F,X}} d(V,W) \leq \sup_{V,W \in \mathcal{G}_{n,r}} d(V,W) \leq \frac{\pi}{2} \sqrt{r},$$

*where  $d$  denotes the geodesic distance on  $\mathcal{G}_{n,r}$  (see [65, 5]).*

*Proof.* See Appendix B for the proof. □

Thus, if  $s < n$  (so that the dimension of  $\mathcal{V}_{F,X}$  is positive), then there are infinitely many possibilities for the range of  $F^T$  that are consistent with the sampled data. When  $s \leq n - r$ , these possibilities are not close together, in fact, they can differ by the maximum amount possible between subspaces.

In the general setting, the covariance matrices  $W_x$  and  $W_g$  may not have low rank. However, in many cases they have quickly decaying eigenvalues giving them low “effective rank”

$$(2.23) \quad r(W) = \text{Tr}(W) / \|W\|,$$

where  $\|W\|$  is the operator norm (see Remark 5.53 in Vershynin [63], as well as sections 5.6, 7.6.1, and 9.2.3 in [64]). For linear time-invariant systems this is known to hold for the observability and controllability Gramians when there are few inputs and observations [48, 3]. Results on the nonasymptotic theory of random matrices including [52, 43, 63, 30, 24] show that covariance matrices with low effective rank can be estimated accurately in the operator norm with high probability using a number of samples that is independent of the ambient dimension  $n$ . This allows for accurate computation of projectors (2.12) using state and gradient samples even when the state dimension far exceeds the number of samples we can reasonably obtain.

**2.3. Extracting nonlinear features using a kernel method.** The fact that the features (2.20) can be computed relying only on inner products in  $\mathbb{R}^n$  suggests a natural reformulation as a kernel method yielding nonlinear features  $z = h(x)$ . Here, we apply the same approach as above after lifting the problem into a high-dimensional (possibly infinite-dimensional) RKHS and computing the required inner products implicitly via the reproducing kernel. A similar approach is described in [49], where feature maps into finite-dimensional spaces were computed explicitly.

We begin by summarizing some basic results and definitions that can be found in [9, 22]. Recall that an RKHS  $\mathcal{H}$  over a subset  $\mathcal{X} \subset \mathbb{R}^n$  is a Hilbert space of functions where pointwise evaluation of  $f \in \mathcal{H}$  at  $x \in \mathcal{X}$  is a bounded linear functional

$$(2.24) \quad f(x) = \langle K_x, f \rangle_{\mathcal{H}}.$$

The function  $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  defined by  $K(x, y) := K_y(x) = \langle K_x, K_y \rangle_{\mathcal{H}}$  is called the “reproducing kernel” of the RKHS. The reproducing kernel is symmetric, namely



$K(x, y) = K(y, x)$ , and positive-definite in the sense that for any  $x_1, \dots, x_N \in \mathcal{X}$  and constants  $c_1, \dots, c_N \in \mathbb{R}$  we have

$$(2.25) \quad \sum_{i,j=1}^N c_i K(x_i, x_j) c_j \geq 0.$$

Moreover, any function  $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$  with these properties uniquely defines an RKHS whose reproducing kernel is  $K$ . In particular, the RKHS defined by  $K$  is the completion of the span of  $\{K_x\}_{x \in \mathcal{X}}$ . The “feature map”  $\Phi_K : \mathcal{X} \rightarrow \mathcal{H}$  defined by  $x \mapsto K_x$  provides lifted representatives of states in the RKHS. When the kernel is continuous, then so is the feature map. The so-called “kernel trick” in machine learning refers to working with such lifted representatives implicitly by means of the kernel  $K$  without explicit calculations in  $\mathcal{H}$ , which may be infinite-dimensional.

Observe that the empirical covariance matrix  $W_x$  in (2.4) may be defined by its action on a vector  $v \in \mathbb{R}^n$  by

$$W_x v = \frac{1}{s_x} \sum_{i=1}^{s_x} x_i \langle x_i, v \rangle.$$

Analogously, the lifted empirical covariance operator  $W'_x : \mathcal{H} \rightarrow \mathcal{H}$  for a collection of states  $\{x_i\}_{i=1}^{s_x} \subset \mathcal{X}$  is defined by its action on a function  $f \in \mathcal{H}$  by

$$(2.26) \quad W'_x f = \frac{1}{s_x} \sum_{i=1}^{s_x} (K_{x_i} - K_0) \langle K_{x_i} - K_0, f \rangle_{\mathcal{H}},$$

where we are centering about the lift of the origin  $K_0$ . Analogously to the finite-dimensional case, this operator can be factorized as  $W'_x = X X^*$  with  $X^*$  denoting the adjoint of the operator  $X : \mathbb{R}^{s_x} \rightarrow \mathcal{H}$  whose action on  $w = (w_1, \dots, w_{s_x}) \in \mathbb{R}^{s_x}$  is defined by

$$(2.27) \quad X w = \frac{1}{\sqrt{s_x}} \sum_{i=1}^{s_x} (K_{x_i} - K_0) w_i.$$

We must define a gradient covariance operator for balancing by lifting the gradients  $\nabla(\xi^T F)(x)$  into the RKHS. To do this, we ensure that there is a differentiable function  $F'$  defined on the image  $\mathcal{X}' := \Phi_K(\mathcal{X}) \subset \mathcal{H}$  so that the diagram

$$(2.28) \quad \begin{array}{ccc} \mathcal{X} & \xrightarrow{F} & \mathbb{R}^m \\ \Phi_K \downarrow & \nearrow F' & \\ \mathcal{X}' & & \end{array}$$

commutes. We lift the gradient  $\nabla(\xi^T F)(x)$  into  $\mathcal{H}$  by computing  $\nabla(\xi^T F')(K_x)$ . Since this will involve differentiating the feature map, we briefly summarize a remarkable result by Zhou [68] showing that the derivatives of a smooth kernel also have reproducing properties. Let  $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}_0^n$  be a multi-index and denote  $\partial^\alpha = \frac{\partial^{|\alpha|}}{\partial x_1^{\alpha_1} \dots \partial x_n^{\alpha_n}}$ , where  $|\alpha| = \alpha_1 + \dots + \alpha_n$ . If  $\mathcal{X}$  is a bounded open set and  $K \in C^{2s}(\bar{\mathcal{X}} \times \bar{\mathcal{X}})$ ,  $s \geq 1$ , is the reproducing kernel for  $\mathcal{H}$ , then Theorem 1 in [68] shows that for  $|\alpha| \leq s$  the function  $(\partial^\alpha K)_x$  defined by  $(\partial^\alpha K)_x(y) := \partial^{(\alpha, 0)} K(x, y) = \partial^\alpha K_y(x)$  is an element of  $\mathcal{H}$  and

$$(2.29) \quad \partial^\alpha f(x) = \langle (\partial^\alpha K)_x, f \rangle_{\mathcal{H}}.$$

In the following lemma, we use this result to show that the feature map is continuously Fréchet differentiable.

LEMMA 2.6. Let  $\mathcal{X} \subset \mathbb{R}^n$  be a bounded open set, and let  $\mathcal{H}$  be an RKHS with reproducing kernel  $K \in C^2(\bar{\mathcal{X}} \times \bar{\mathcal{X}})$ . Then the feature map  $\Phi_K : x \mapsto K_x$  has a Fréchet derivative  $D\Phi_K(x) : \mathbb{R}^n \rightarrow \mathcal{H}$  at each  $x \in \mathcal{X}$  and satisfies

$$(2.30) \quad D\Phi_K(x)v = \sum_{j=1}^n (\partial^{e_j} K)_x v_j, \quad D\Phi_K(x)^* f = \nabla f(x),$$

for every  $v = (v_1, \dots, v_n) \in \mathbb{R}^n$  and  $f \in \mathcal{H}$ . Moreover, the map  $x \mapsto D\Phi_K(x)$  with  $x \in \mathcal{X}$  is continuous with respect to the operator norm, i.e.,  $\Phi_K$  is a  $C^1$  function on  $\mathcal{X}$  (see Definition 1.1.2 in Kesavan [29]).

*Proof.* See Appendix C for the proof.  $\square$

Given this setup, our main result is the following.

THEOREM 2.7. Let  $\mathcal{X} \subset \mathbb{R}^n$  be a bounded open set, and let  $\mathcal{H}$  be an RKHS with reproducing kernel  $K \in C^2(\bar{\mathcal{X}} \times \bar{\mathcal{X}})$  so that the feature map  $\Phi_K$  is injective on the closure  $\bar{\mathcal{X}}$  and the derivative Gram matrix,

$$(2.31) \quad G(x) := D\Phi_K(x)^* D\Phi_K(x) = \begin{bmatrix} \partial^{(e_1, e_1)} K(x, x) & \cdots & \partial^{(e_1, e_n)} K(x, x) \\ \vdots & \ddots & \vdots \\ \partial^{(e_n, e_1)} K(x, x) & \cdots & \partial^{(e_n, e_n)} K(x, x) \end{bmatrix},$$

is positive-definite for every  $x \in \mathcal{X}$ . Then the feature map  $\Phi_K$  is a  $C^1$  embedding of  $\mathcal{X}$  into  $\mathcal{H}$  (see Definition 3.3.1 in Margalef-Roig and Domínguez [40]). Hence, there is a  $C^1$  function  $F' = F \circ \Phi_K^{-1} : \Phi_K(\mathcal{X}) \rightarrow \mathbb{R}^m$  so that (2.28) commutes and for every  $x \in \mathcal{X}$  we have

$$(2.32) \quad \nabla(\xi^T F')(K_x) = D\Phi_K(x)G(x)^{-1} \nabla(\xi^T F)(x) \in \mathcal{H}.$$

*Proof.* See Appendix C for the proof.  $\square$

Note that the theorem requires that the feature map  $\Phi_K : x \mapsto K_x$  be injective. An easy test for injectivity is that, for every distinct  $x_1, x_2 \in \bar{\mathcal{X}}$ ,

$$(2.33) \quad \|K_{x_1} - K_{x_2}\|_{\mathcal{H}}^2 = K(x_1, x_1) - 2K(x_1, x_2) + K(x_2, x_2) > 0.$$

To form the empirical gradient covariance operator, we first produce gradient samples  $\{g_i\}_{i=1}^{s_g}$  from randomly chosen directions  $\xi_i$  and states  $\tilde{x}_i$ , as in (2.18) (here, we use  $\tilde{x}_i$  for the sampled states, because these may be distinct from the states  $x_i$  used to form the state covariance operator in (2.26)). We then lift the gradient samples using Theorem 2.7, to define the empirical gradient covariance operator

$$(2.34) \quad W'_g = \frac{1}{s_g} \sum_{i=1}^{s_g} D\Phi_K(\tilde{x}_i)G(\tilde{x}_i)^{-1}g_i g_i^T G(\tilde{x}_i)^{-1} D\Phi_K(\tilde{x}_i)^*.$$

Analogously to the finite-dimensional case, this operator may be factorized as  $W'_g = YY^*$  with  $Y : \mathbb{R}^{s_g} \rightarrow \mathcal{H}$  defined by its action on  $w = (w_1, \dots, w_{s_g}) \in \mathbb{R}^{s_g}$  according to

$$(2.35) \quad Yw = \frac{1}{\sqrt{s_g}} \sum_{i=1}^{s_g} w_i D\Phi_K(\tilde{x}_i)G(\tilde{x}_i)^{-1}g_i.$$

The adjoint of this operator acting on a lifted state  $K_x$  for  $x \in \mathcal{X}$  is the vector in  $\mathbb{R}^{s_g}$  whose  $i$ th component is given by

$$(2.36) \quad [Y^* K_x]_i = \frac{1}{\sqrt{s_g}} g_i^T G(\tilde{x}_i)^{-1} \nabla K_x(\tilde{x}_i).$$

Here, the  $n$ -dimensional vector

$$(2.37) \quad \nabla K_y(x) = \left( \partial^{(e_1, 0)} K(x, y), \dots, \partial^{(e_n, 0)} K(x, y) \right)$$

is found by differentiating the kernel with respect to the coordinates of the first entry. Thus, we make the crucial observation that (2.36) can be computed using the kernel without doing explicit calculations in the possibly infinite-dimensional RKHS.

To find a truncated balancing transformation for  $W'_x$  and  $W'_g$ , we compute an SVD of the  $s_g \times s_x$  matrix  $Y^* X = U \Sigma V^T$ , whose elements are given by

$$(2.38) \quad [Y^* X]_{i,j} = \frac{1}{\sqrt{s_g s_x}} g_i^T G(\tilde{x}_i)^{-1} (\nabla K_{x_j}(\tilde{x}_i) - \nabla K_0(\tilde{x}_i)),$$

thanks to (2.36). As in the finite-dimensional case, an oblique projection  $P: \mathcal{H} \rightarrow \mathcal{H}$  onto an  $r$ -dimensional subspace may be defined by (2.12). The information extracted by  $P$  acting on a lifted state  $K_x$  is encoded in the  $r$ -dimensional feature vector

$$(2.39) \quad z = h(x) = \Psi^*(K_x - K_0) = \Sigma_r^{-1/2} U_r^T (Y^* K_x - Y^* K_0).$$

These features can be computed explicitly using the kernel and (2.36). The feature vectors associated with the original data are given by the columns of

$$(2.40) \quad [h(x_1) \quad \dots \quad h(x_{s_x})] = \sqrt{s_x} \Sigma_r^{1/2} V_r^T.$$

Although we do not pursue nonlinear Galerkin modeling in this paper, one could use the derivative of the feature vector (2.39) for this purpose. The action of the derivative on a vector  $v \in \mathbb{R}^n$  is given by

$$(2.41) \quad D h(x) v = \Sigma_r^{-1/2} U_r^T D(Y^* \Psi_K(x)) v,$$

where the elements of  $D(Y^* \Psi_K(x)) v = \frac{1}{\sqrt{s_g}} [g_i^T G(\tilde{x}_i)^{-1} H(\tilde{x}_i, x) v]_{i=1}^{s_g}$  are computed using the matrix  $H(x, y) = D \Phi_K(x)^* D \Phi_K(y) = [\partial^{(e_i, e_j)} K(x, y)]_{i,j=1}^n$ .

Some examples of common kernels and the corresponding functions  $\nabla K_y(x)$  and  $G(x)^{-1}$  are given in Table 1. Fortunately, we see that  $\nabla K_y(x)$  can be computed with time complexity  $\mathcal{O}(n)$ . Moreover, in each case  $G(x)^{-1}$  has structure that allows us to act with it on a vector with time complexity  $\mathcal{O}(n)$ . These properties allow the method to be implemented on problems with very large  $n$ , where matrix inversion and even dense matrix-vector products would be computationally prohibitive. Finally, we note that when  $K(x, y) = x^T y$ , the kernel method becomes identical to the technique described in subsection 2.1.

**3. Randomized adjoint sampling methods.** In order to obtain the gradient samples described in subsection 2.1, we must be able to compute gradients of the output sequence map  $F$ . Perhaps the most natural way to compute such gradients is by finite difference approximation; however, this method would scale poorly with

TABLE 1

*Some common smooth kernels and the functions required to implement the kernel method.*

$K(x, y)$	$\nabla K_y(x)$	$G(x)^{-1}$
$\alpha + x^T y, \quad \alpha \geq 0$	$y$	$I$
$(\alpha + x^T y)^p, \quad \begin{cases} p > 1 \\ \alpha > 0 \end{cases}$	$p(\alpha + x^T y)^{p-1} y$	$\frac{1}{p(\alpha + \ x\ ^2)^{p-1}} \left[ I - \left( \frac{p-1}{\alpha + p\ x\ ^2} \right) x x^T \right]$
$\exp\left(-\frac{\ x-y\ ^2}{2\sigma^2}\right)$	$-\frac{1}{\sigma^2} K(x, y)(x - y)$	$\sigma^2 I$

the state dimension. Instead, in this section we describe a more computationally efficient way of obtaining these gradient samples, using linearized adjoint equations derived from (2.1). Our method entails computing gradients of output sequences with respect to the initial conditions  $x_0$  of “mini-trajectories” with length  $L + 1$ . In computing these gradients, the adjoint method also produces gradients with respect to intermediate states along these trajectories, which, in general, must be discarded. However, when we sample from the probability distribution constructed from longer trajectories of the system in the paragraph preceding subsection 2.1 the intermediate gradient information can be retained, yielding computational benefits. We discuss a gradient sampling method for an empirical version of this distribution in subsection 3.1. In subsection 3.2 we discuss a simplified method that can be applied when the distribution over pairs  $(x, \bar{u})$  is “stationary” in a sense that will be described later.

The random vectors  $\xi \in \mathbb{R}^{m_0(L+1)}$  that we use to sample the gradient are constructed as follows. Let  $\eta \in \mathbb{R}^{m_0}$  be a zero mean random vector with  $\mathbb{E}[\eta\eta^T] = (L+1)I$ , and let  $\tau \in \{0, \dots, L\}$  be chosen uniformly at random. Then the random vector  $\xi \in \mathbb{R}^{m_0(L+1)}$  formed by placing  $\eta$  into the  $\tau$ th slot of

$$(3.1) \quad \xi = (0 \quad \cdots \quad 0 \quad \eta \quad 0 \quad \cdots \quad 0) = e_\tau \otimes \eta,$$

also has zero mean and covariance  $\mathbb{E}[\xi\xi^T] = I$ .

Along a trajectory of (2.1), the gradients

$$(3.2) \quad g_\eta(t, k) = \nabla_x(\eta^T F_k)(x(t), u(t:t+k-1))$$

satisfy the adjoint equation

$$(3.3) \quad \begin{aligned} g_\eta(t_f - k, k) &= D_x f(x(t_f - k), u(t_f - k))^T g_\eta(t_f - (k-1), k-1) \\ g_\eta(t_f, 0) &= D_x g(x(t_f))^T \eta. \end{aligned}$$

With our choice of  $\xi$  given by (3.1), we generate

$$(3.4) \quad \nabla_x(\xi^T F)(x(0), u(0:L-1)) = \nabla_x(\eta^T F_\tau)(x(0), u(0:\tau-1)) = g_\eta(0, \tau)$$

by computing  $g_\eta(\tau - k, k)$  for  $k = 0, 1, \dots, \tau$  recursively using (3.3). In what follows, we discuss cases where these intermediate gradient samples can be incorporated in the empirical gradient covariance.

**3.1. Sampling from long trajectories.** In many cases of practical interest we generate one or more long trajectories of snapshots  $x(0:N+L)$  via simulation with different initial conditions and input sequences  $u(0:N+L-1)$ . For simplicity, we consider a single long trajectory where the first  $N+1$  snapshots serve as initial conditions for the time- $L$  input-output map (2.3). Hence, the state covariance is based on a uniform distribution over the first  $N+1$  snapshots. To draw samples of the gradient with respect to this distribution, we provide Algorithm 3.1.

---

**Algorithm 3.1** Sample gradients from long trajectories.
 

---

- 1: **input:** the time-horizon  $L$ , the number of samples  $s_g$ , a long trajectory  $\{x(0), x(1), \dots, x(N+L)\}$ , and a distribution for  $\eta \in \mathbb{R}^{m_0}$  with zero mean and  $\mathbb{E}[\eta\eta^T] = (L+1)I$
- 2: **for**  $i = 1, 2, \dots, s_g$  **do**
- 3:   Draw  $t'$  uniformly from  $\{0, \dots, N\}$ , draw  $\tau'$  uniformly from  $\{0, \dots, L\}$ , and draw  $\eta$  from its distribution.
- 4:   Letting  $t_f = t' + \tau'$ , solve the adjoint equation (3.3) to generate  $g_\eta(t_f - k, k)$  for each  $k = 0, \dots, \min\{L, t_f\}$ .
- 5:   With  $\tau_{\min} = \max\{0, t_f - N\}$  and  $\tau_{\max} = \min\{L, t_f\}$ , arrange the samples with  $\tau_{\min} \leq k \leq \tau_{\max}$  into the matrix

$$Y_i = \frac{1}{\sqrt{1 + \tau_{\max} - \tau_{\min}}} [g_\eta(t_f - \tau_{\min}, \tau_{\min}) \quad \cdots \quad g_\eta(t_f - \tau_{\max}, \tau_{\max})].$$

6: **end for**

7: **return** the gradient sample matrix  $Y = \frac{1}{\sqrt{s_g}} [Y_1 \quad \cdots \quad Y_{s_g}]$ .

---

During the  $i$ th stage of this procedure we obtain samples by solving the adjoint equation (3.3) over a time horizon of length  $\min\{L, t_f\}$ . In most applications, the computational cost to act with  $D_x f(x, u)^T$  on a vector is comparable to evaluating  $f(x, u)$ , meaning that the cost of each stage is comparable to simulating the FOM (2.1) over the same time horizon  $\min\{L, t_f\}$ .

In what follows, we show that Algorithm 3.1 corresponds to a Monte-Carlo approximation for the gradient covariance. First, we observe that the gradient covariance matrix can be written in terms of the sequences generated by solving (3.3) from final times  $t_f = t' + \tau'$  with  $(t', \tau')$  drawn uniformly from  $\{0, \dots, N\} \times \{0, \dots, L\}$ . In particular, given a final time  $t_f$  between 0 and  $N+L$ , the number of initial times  $t$  and prediction horizons  $\tau$  that sum to  $t_f$  is given by

$$(3.5) \quad \begin{aligned} \nu(t_f) &= \sum_{t=0}^N \sum_{\tau=0}^L \delta_{t+\tau, t_f} = 1 + \min\{t_f, N, L, N+L-t_f\} \\ &= 1 + \min\{L, t_f\} - \max\{0, t_f - N\}. \end{aligned}$$

Using this counting function we can express the gradient covariance as

$$(3.6) \quad \begin{aligned} W_g &= \frac{1}{(N+1)(L+1)} \sum_{t=0}^N \sum_{\tau=0}^L \mathbb{E}_\eta [g_\eta(t, \tau) g_\eta(t, \tau)^T] \\ &= \frac{1}{(N+1)(L+1)} \sum_{t'=0}^N \sum_{\tau'=0}^L \mathbb{E}_\eta \left[ \frac{1}{\nu(t' + \tau')} \sum_{t=0}^N \sum_{\tau=0}^L \delta_{t+\tau, t'+\tau'} g_\eta(t, \tau) g_\eta(t, \tau)^T \right] \\ &= \frac{1}{(N+1)(L+1)} \sum_{t'=0}^N \sum_{t_f=t'}^{t'+L} \mathbb{E}_\eta \left[ \frac{1}{\nu(t_f)} \sum_{k=\max\{0, t_f-N\}}^{\min\{L, t_f\}} g_\eta(t_f - k, k) g_\eta(t_f - k, k)^T \right]. \end{aligned}$$

The outer two summations in the last expression compute an average over final times  $t_f$  and the expectation is computed with respect to the random vector  $\eta$ . The inner

summation in the last expression computes an average over the horizon lengths  $k$  corresponding to initial times  $t = t_f - k$  falling between 0 and  $N$ . The empirical covariance with factor  $Y$  produced by Algorithm 3.1 corresponds to using a Monte-Carlo method to approximate the outer two summations and the expectation over  $\eta$  in the last expression of (3.6).

**3.2. Sampling from a stationary distribution.** We consider a statistically stationary case where the distributions of states and inputs are independent of time. More precisely, the distributions of  $(x(t), u(t:t+L-1))$  and  $(x_0, u(0:L-1))$  are assumed to be identical for each  $t = 1, \dots, N$ . Hence, the distribution for  $(x, \bar{u})$  described in the paragraph before subsection 2.1 is the same as the distribution for  $(x_0, u(0:L-1))$ . For example, this situation occurs when the initial conditions  $x_0$  are sampled from the invariant distribution on an attractor with zero input, or input provided by state feedback plus independent noise. Another example is when  $x_0$  and  $u$  are drawn by choosing an initial time uniformly along a periodic orbit of (2.1). We observe that the stationarity assumption implies that  $g_\eta(L-k, k)$  and  $g_\eta(0, k)$  are identically distributed for  $k = 0, \dots, L$ , yielding

$$(3.7) \quad W_g = \mathbb{E} \left[ \frac{1}{L+1} \sum_{\tau=0}^L g_\eta(0, \tau) g_\eta(0, \tau)^T \right] = \mathbb{E} \left[ \frac{1}{L+1} \sum_{k=0}^L g_\eta(L-k, k) g_\eta(L-k, k)^T \right].$$

Here the expectation is taken over the initial condition  $x_0$ , the input sequence  $u(0:L-1)$ , and the random vector  $\eta$ .

To construct a Monte-Carlo approximation of the gradient covariance, we first sample “mini-trajectories”  $x_i(0:L)$ ,  $i = 1, \dots, s_g$ , of the system. For each, we choose  $\eta_i$  independently at random and solve (3.3), yielding the columns of a matrix

$$(3.8) \quad Y_i = \frac{1}{\sqrt{L+1}} \begin{bmatrix} g_{\eta_i}(L, 0) & g_{\eta_i}(L-1, 1) & \cdots & g_{\eta_i}(0, L) \end{bmatrix}.$$

The empirical gradient covariance for the collection of mini-trajectories can then be factored using

$$(3.9) \quad Y = \frac{1}{\sqrt{s_g}} \begin{bmatrix} Y_1 & \cdots & Y_{s_g} \end{bmatrix},$$

where no intermediate gradient samples have been wasted.

## 4. Results.

**4.1. A challenging model problem.** We consider the same problem proposed in [44], where we seek two-dimensional Petrov–Galerkin ROMs for the system

$$(4.1) \quad \begin{aligned} \dot{x}_1 &= -x_1 + 20x_1x_3 + u, \\ \dot{x}_2 &= -2x_2 + 20x_2x_3 + u, \\ \dot{x}_3 &= -5x_3 + u, \\ y &= x_1 + x_2 + x_3. \end{aligned}$$

As discussed in [44], model reduction for this system is challenging due to its strong nonlinear interactions involving the state  $x_3$ , which has small variance compared to  $x_1$  and  $x_2$ . In [44] the projections found by various methods mentioned in section 1 (namely, POD, BT, QB BT, and QB-IRKA) were compared to those found by a newly

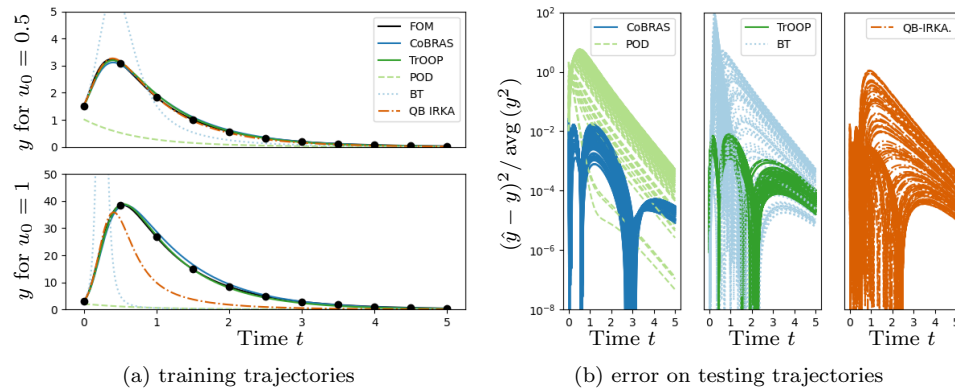


FIG. 1. In (a) we show the output trajectories of the FOM (4.1) and various ROMs along the training trajectories. The samples are indicated by black dots. In (b) we show the normalized square prediction errors for each ROM along 100 trajectories with  $u_0$  chosen uniformly at random from the interval  $[0, 1]$ .

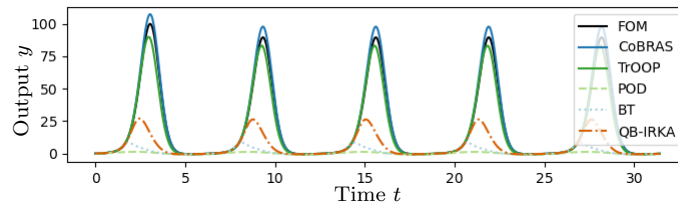


FIG. 2. We show the output trajectory of the FOM (4.1) and the predicted outputs of each ROM with zero initial condition and input  $u(t) = \sin(t)$ .

proposed method: trajectory-based optimization for oblique projections (TrOOP). TrOOP is an iterative method that uses gradient descent to find a Petrov–Galerkin projection that minimizes error along a collection of training trajectories. Here, we add CoBRAS to this list using precisely the same setup. The training data consisted of the two trajectories shown in Figure 1a sampled every  $\Delta t = 0.5$ . These are nonlinear impulse-responses generated by simulating (4.1) with zero input and initial conditions  $x(0) = (u_0, u_0, u_0)$  with magnitudes  $u_0 = 0.5$  and  $u_0 = 1.0$ . The covariance matrices for CoBRAS were defined using  $L = 5$  of these intervals as the horizon length and initial conditions uniformly distributed over the 22 sample points.

The prediction accuracy of the resulting ROMs on 100 impulse responses with magnitudes  $u_0$  drawn uniformly from the interval  $[0, 1]$  is shown in Figure 1b. Performance on a trajectory with sinusoidal input  $u(t) = \sin(t)$  is also shown in Figure 2. We observe that CoBRAS achieve prediction accuracy comparable to the optimized projection found by TrOOP. The performance is also not very sensitive to the choice of horizon length, with comparable prediction accuracy observed when  $L \geq 4$ .

**4.2. Nonlinear axisymmetric jet flow.** We consider the same nonlinear jet flow problem discussed in [44] at a higher Reynolds number  $Re = 2000$ . This system is governed by a discretization of the axisymmetric incompressible Navier–Stokes equations with  $10^5$  states. The flow is sensitive to disturbances introduced upstream in the shear layer about a stable equilibrium. We observe the full state of the system given by the equilibrium-subtracted velocity field scaled by a radially-dependent

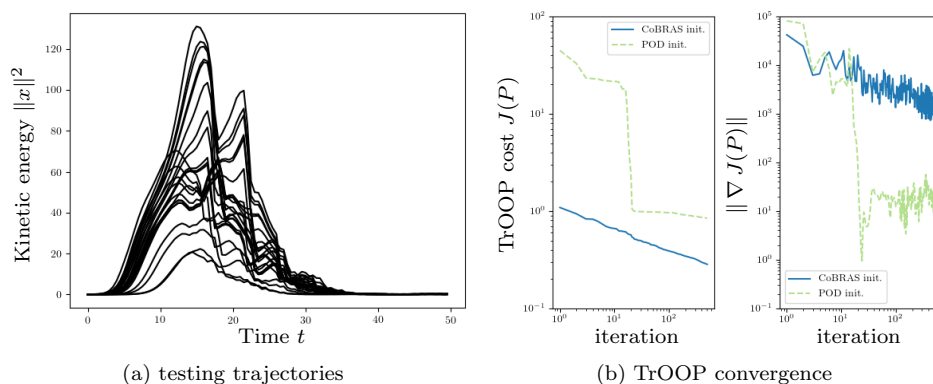


FIG. 3. In (a) we show the energy of the jet flow along the testing impulse-response trajectories. In (b) we compare the convergence of TrOOP for the jet flow using POD and CoBRAS for initialization and the cost function (4.4).

weight. The weight is chosen so that the inner product on the state space is Euclidean and the squared norm of a state equals the kinetic energy of the corresponding equilibrium-subtracted flowfield. Input is provided to the system via a source term in the radial momentum equation concentrated in the shear layer near a radius 0.5 and down-stream distance from the nozzle 1.0, as indicated in Figure 5a. The resulting FOM can be written as

$$(4.2) \quad \dot{x} = f_0(x) + bu, \quad x(0) = x_0,$$

where  $f_0$  is a quadratic function derived from our spatial discretization of the Navier–Stokes equations and  $b$  is a vector corresponding to the source term described above. For more details, see [44].

The training data consists of 12 nonlinear impulse-response trajectories in which we set  $u = 0$  and  $x_0 = bu_0$ , with magnitudes  $u_0 = \pm 0.005, \pm 0.02, \pm 0.04, \pm 0.06, \pm 0.08$ , and  $\pm 0.10$ . The state along each trajectory was sampled 100 times at intervals  $\Delta t = 0.5$ . The testing data consisted of 25 such trajectories with impulse magnitudes drawn uniformly at random from the interval  $[-0.1, 0.1]$ . The energy along these trajectories, plotted in Figure 3a, indicates that the system undergoes rapid and nonlinear transient growth before the disturbances leave the computational domain through the outflow boundary.

**4.2.1. Petrov–Galerkin models.** We construct Petrov–Galerkin ROMs for the jet flow (4.2) using  $r = 40$ -dimensional projection operators  $P = \Phi\Psi^T$  determined by POD, BPOD about the equilibrium, CoBRAS, and TrOOP. Specifically, the state forecasts  $\hat{x}(t) = \Phi z(t)$  were obtained by simulating the 40-dimensional system

$$(4.3) \quad \dot{z} = \tilde{f}(z, u) := \Psi^T f_0(\Phi z) + \Psi^T bu, \quad z(0) = \Psi^T x_0,$$

where  $\tilde{f}$  was evaluated using precomputed tensors assembled from the quadratic function  $f_0$  and the columns of  $\Phi$  and  $\Psi$  as in section 4.2 of [23]. While wall-clock timing is implementation dependent, our simulations of (4.3) took  $\sim 0.5$  seconds per trajectory of length  $100\Delta t$  on a laptop computer, whereas simulating (4.2) took  $\sim 180$  seconds per trajectory.



The POD projection was computed using the snapshots from the training trajectories. We computed the BPOD projection using a time horizon equal to the length of each training trajectory and 20-dimensional output projection. For CoBRAS, the gradient was sampled along the training trajectories using the method described in subsection 3.1 with a time horizon consisting of  $L = 40$  intervals of length  $\Delta t$  and  $N = 100$  (we simulated an extra  $40\Delta t$  at the end of each training trajectory). We solved the adjoint equation 10 times per trajectory, yielding  $s_g = 120$ . Per the discussion at the end of subsection 3.1, the computational cost to obtain the gradient samples was comparable to simulating the FOM (4.2) over a time horizon equal to  $s_g \cdot L \cdot \Delta t = 4800\Delta t$ .

The CoBRAS projection was used to initialize the gradient descent in TrOOP with the same cost function for the jet flow described in [44]. This cost function is given by

$$(4.4) \quad J(P) = \frac{1}{12} \sum_{k=1}^{12} \frac{\sum_{l=0}^{99} \|x^{(k)}(l\Delta t) - \hat{x}^{(k)}(l\Delta t; P)\|^2}{\sum_{l=0}^{99} \|x^{(k)}(l\Delta t)\|^2} + \gamma\rho(P),$$

where  $\hat{x}^{(k)}$  denotes the projection-dependent state forecast of (4.3) corresponding to the  $k$ th training trajectory  $x^{(k)}$ . The regularization function  $\rho$  is defined in [44] and we use the same regularization strength  $\gamma = 10^{-3}$ . At each step of the geometric conjugate gradient method (Algorithm 4.2 in [44]), line search was carried out in order to satisfy the weak Wolfe conditions with  $c_1 = 0.01$  and  $c_2 = 0.1$ . The gradient was computed using the adjoint sensitivity method described by Algorithm 4.1 in [44] with  $q = 3$  Gauss–Legendre quadrature points per sampling interval. The convergence of TrOOP with initialization provided by CoBRAS and POD are compared in Figure 3b. In both cases, we performed 500 iterations. We observe that after roughly 20 iterations with POD initialization, the gradient rapidly decreases by three orders of magnitude and the cost remains approximately constant, indicating that TrOOP has become stuck near a local minimum of the cost function. A much better initialization is provided by CoBRAS, which allows TrOOP to reach a lower value of the cost.

The error introduced by each projection in reconstructing the states along the testing trajectories is plotted in Figure 4a. Unsurprisingly, the POD projection has the lowest square error at nearly all times. On the other hand, BPOD cannot accurately reconstruct states that depart from the responses of the linearized system. Both CoBRAS and TrOOP have larger reconstruction error than POD, except at very early times. Until roughly  $t = 5$ , the reconstruction error for CoBRAS and TrOOP is lower than POD.

We compare the performance of the ROMs in terms of forecasting error normalized by the mean kinetic energy along each testing trajectory in Figure 4b. Predicted flowfield snapshots along the most energetic testing trajectory are shown in Figure 5. We observe that the POD-based model does not capture the growth of the disturbance, likely because the most energetic POD modes were primarily supported downstream. On the other hand the BPOD-based model captures the initial growth, but rapidly blows up as nonlinearities become significant. Between these extremes is CoBRAS, which captures the initial growth and has physically plausible, yet quantitatively inaccurate behavior at later times. Though CoBRAS provided quantitatively accurate forecasts for only a short time, it produced a suitable initial projection for TrOOP, which was able to significantly reduce the error at later times.

Similar and slightly improved results were also obtained using CoBRAS with longer gradient sampling horizons  $L$  and more gradient samples. The performance of

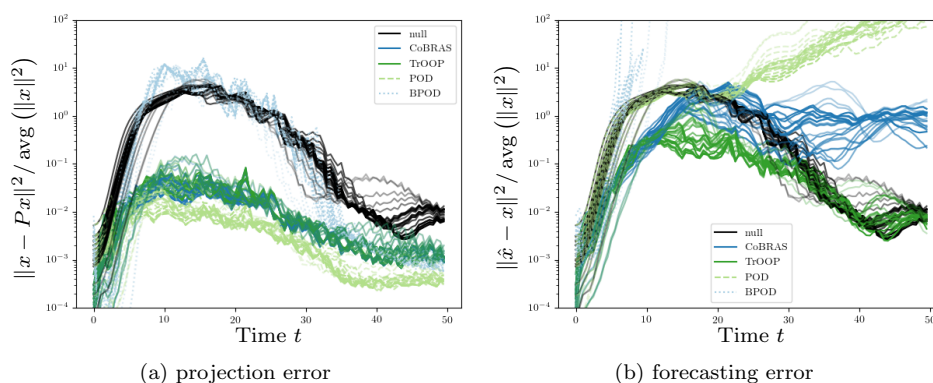


FIG. 4. In (a) we show the reconstruction error using the different linear projections along each testing trajectory. The “null” projection means  $Px = 0$ . In (b) we show the error of the predictions made by the Petrov–Galerkin ROMs. The “null” forecast means  $\hat{x} = 0$ . In (a) and (b) we normalize the errors by the mean kinetic energy along each trajectory. The opacity of the trajectories increases with  $\text{avg}(\|x\|^2)$ .

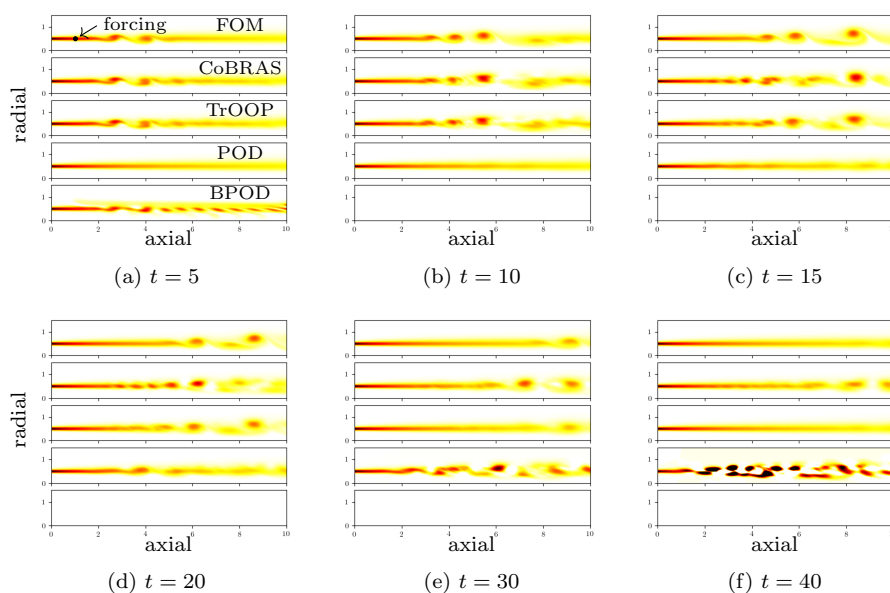


FIG. 5. Snapshots of predicted vorticity in the jet flow using 40-dimensional Petrov–Galerkin ROMs along the most energetic testing trajectory, which had impulse magnitude  $u_0 = 0.0938$ .

POD and BPOD was largely independent of the model dimension, while the dimension had a significant effect on the performance of CoBRAS. Certain model dimensions consistently produced accurate predictions regardless of the gradient sampling strategy, while others produced models that blew up on some of the larger trajectories after correctly predicting the initial growth of the disturbance up to  $t \approx 15$ .

**4.2.2. Learned models in nonlinear feature space.** In this section we compare 15- and 30-dimensional ROMs for the jet flow in the spaces of nonlinear coordinates found by KPCA and the kernel CoBRAS (K-CoBRAS) method described in

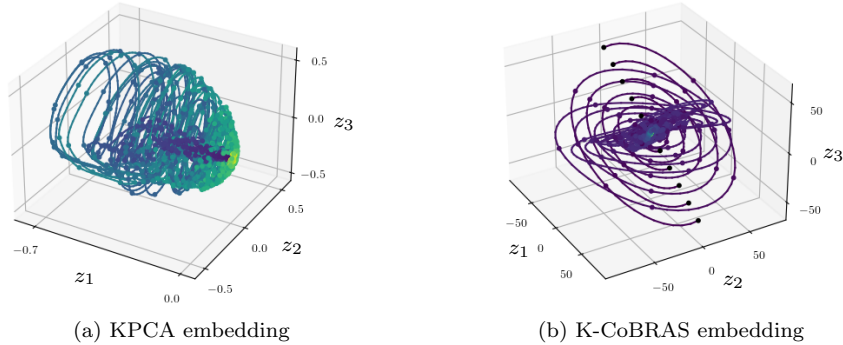


FIG. 6. In (a) and (b) we show the training trajectories of the jet flow in the leading three nonlinear coordinates found by KPCA and K-CoBRAS, respectively. The coloring is from blue to yellow according to time. (Figure in color online.)

subsection 2.3. For both methods we use the Gaussian kernel with width  $\sigma = 8.0$  (see the third row in Table 1). The same gradient samples described above in subsection 4.2.1 were used to construct the K-CoBRAS embedding. To preserve the location of the equilibrium, the KPCA features were centered about the origin by projecting onto the leading eigenvectors of the empirical state covariance operator (2.26).

The training trajectories described at the beginning of subsection 4.2 are plotted in the resulting nonlinear feature spaces  $z = h(x)$  using the leading three KPCA coordinates in Figure 6a and the leading three K-CoBRAS coordinates in Figure 6b. To make these plots, we interpolated the embedded data using piecewise cubic polynomials and derivative information  $\dot{z} = Dh(x)\dot{x}$  obtained at each sample point using the kernel tangent map (2.41) and the FOM. While the KPCA coordinates reflect the energetic growth and decay along trajectories, the points at early and late times are mapped very closely to the equilibrium point at the origin. On the other hand, K-CoBRAS rely on sensitivity information and consequently maps the initial conditions to points far away from the origin. The resulting trajectories spiral inward towards the stable equilibrium point. The leading K-CoBRAS coordinates primarily capture the initial growth of the disturbance in an upstream region, with at least  $r = 10$  coordinates being necessary to represent the behavior of downstream vortices.

To construct ROMs using the nonlinear coordinates  $z = h(x)$  extracted by KPCA and K-CoBRAS, we employed kernel ridge regression (KRR) to fit the discrete-time dynamics  $\tilde{f}$  and reconstruction map  $\tilde{g}$  to the training data described at the beginning of subsection 4.2. This yields a model in the form of (2.2) where  $\tilde{f}$  approximates the map  $z(t) \mapsto z(t + \Delta t)$  and  $\tilde{g}$  approximates the reconstruction map  $z \mapsto x$ . Since it would be impractical to fit a reconstruction function for each of the  $10^5$  state variables in the jet flow, we used the (nonlinear) KPCA and K-CoBRAS coordinates to reconstruct the leading 100 (linear) POD and CoBRAS coordinates, respectively. The full state was then linearly reconstructed in the 100-dimensional POD and CoBRAS bases. Gaussian radial basis function (RBF) kernels were used for regression with the parameters listed in Table 2 selected by 5-fold cross-validation over parameter grids. The fitting process was carried out using the “KernelRidge” and “GridSearchCV” tools in Scikit-Learn [47] after normalizing the variance of each coordinate. Though wall-clock timing is implementation dependent, our forecasts took  $\sim 2$  seconds per trajectory using the 15- and 30-dimensional K-CoBRAS models on a laptop computer. Forecasts using the 15- and 30-dimensional KPCA models took  $\sim 5$  and  $\sim 13$  seconds per trajectory, respectively.

TABLE 2

Kernel ridge regression parameters used to fit the dynamics and reconstruction maps for the jet flow in the nonlinear feature spaces identified by K-CoBRAS and KPCA.

Coordinates	Dimension $r$	Mapping	KRR regularization $\alpha$	RBF width $\gamma$
K-CoBRAS	15	$\tilde{g}$	0.1000	0.1000
K-CoBRAS	15	$\tilde{f}$	$7.943 \times 10^{-6}$	$5.012 \times 10^{-4}$
K-CoBRAS	30	$\tilde{g}$	0.005012	0.01585
K-CoBRAS	30	$\tilde{f}$	$3.981 \times 10^{-5}$	0.001
KPCA	15	$\tilde{g}$	0.02512	0.03981
KPCA	15	$\tilde{f}$	$1.585 \times 10^{-4}$	$2.512 \times 10^{-3}$
KPCA	30	$\tilde{g}$	0.01259	0.01585
KPCA	30	$\tilde{f}$	$6.310 \times 10^{-5}$	$5.012 \times 10^{-4}$

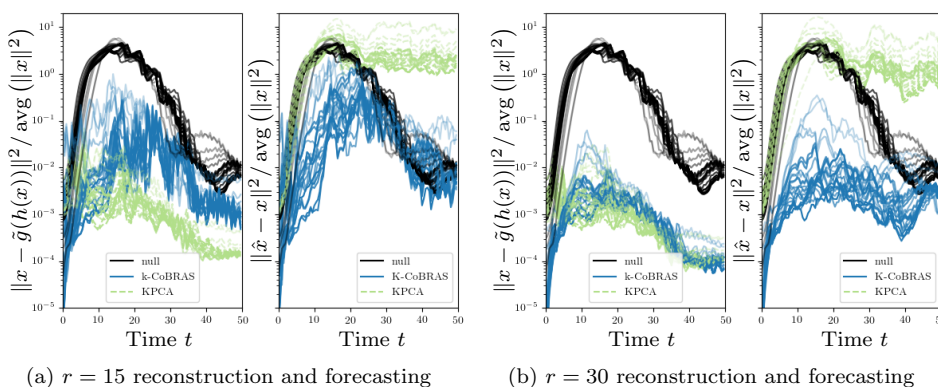


FIG. 7. We show the reconstruction and forecasting error along testing trajectories using the 15- and 30-dimensional models of the jet flow constructed in KPCA and K-CoBRAS coordinates. For the “null” predictions, we reconstruct/forecast zero. Opacity of the curves increases with  $\text{avg}(\|x\|^2)$ .

The prediction accuracy of the resulting data-driven ROMs on the testing trajectories is shown in Figure 7. The models built in K-CoBRAS coordinates have superior prediction accuracy at nearly all times, with accuracy increasing with the number of coordinates in the model. On the other hand, the models in KPCA coordinates are no better than the null forecast, though their reconstruction error is lower than the K-CoBRAS models after roughly  $t = 10$ . The K-CoBRAS reconstructions are more accurate at early times, allowing these models to capture the initial growth of disturbances in the upstream region of the flow. Examining the predicted snapshots in Figure 8 using the 15-dimensional models along the most energetic testing trajectory confirms that the model in KPCA coordinates fails to capture the upstream disturbances, yielding inaccurate forecasts. On the other hand, the fitted models in K-CoBRAS coordinates accurately predict the flow’s response even at late times.

**5. Conclusion.** We have introduced covariance balancing reduction using adjoint snapshots (CoBRAS) as a method for finding low-dimensional coordinates for model reduction of nonlinear dynamical systems. These coordinates capture information about future outputs of the system in the sense of [66] by balancing the sensitivity of future outputs against the variance of the distribution of states generated along trajectories, as measured by empirical state and gradient covariance matrices. The resulting method is analogous to balanced truncation (BT) [42] with covariance ma-

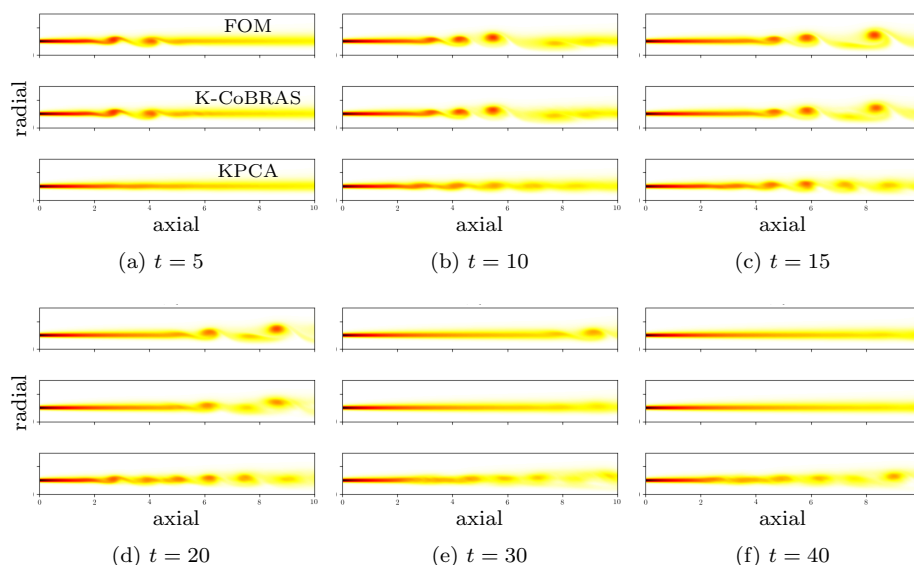


FIG. 8. Snapshots of predicted vorticity in the jet flow using learned 15-dimensional models in kernel feature space along the most energetic testing trajectory, which had impulse magnitude  $u_0 = 0.0938$ .

trices replacing the system Gramians and obeying the same transformation laws. An efficient snapshot-based computational procedure is provided by analogy to balanced proper orthogonal decomposition (BPOD) [50]. The features extracted by CoBRAS are associated with an oblique projection operator that can be used directly for constructing Petrov–Galerkin models. We demonstrate the performance of CoBRAS–Galerkin models on a simple, yet challenging three-dimensional system as well as on a jet flow simulation with  $10^5$  state variables. Not only did the CoBRAS–Galerkin models perform well, but they also provided superior initializations compared to POD for the recently proposed method of trajectory-based optimization for oblique projections (TrOOP) [44].

We observe that the CoBRAS features (coordinates) depend only on inner products between state and gradient vectors. This allows us to replace the inner product with a kernel function in order to identify nonlinear features in a higher-dimensional reproducing kernel Hilbert space (RKHS) using a method we refer to as kernel-CoBRAS (K-CoBRAS). While the associated projection is now defined in the RKHS and cannot be used for Petrov–Galerkin projection of the full-order model, it is possible to construct ROMs governing the evolution of the extracted nonlinear coordinates by employing regression-based methods. We applied this approach to the jet flow and compared the performance of the K-CoBRAS coordinates to kernel principal component analysis (KPCA) coordinates by fitting the discrete-time dynamics and reconstruction maps via KRR. The learned K-CoBRAS model accurately captured the dynamics, while the KPCA model failed to capture the growth of disturbances. We argue that this is likely because KPCA truncated coordinates associated with the low-variance upstream growth, while K-CoBRAS retained these features due to its reliance on sensitivity information.

Our code was written in Python and is available at <https://github.com/samotto1/CoBRAS>.

**Appendix A. Proof of Theorem 2.2.** By the tower rule for conditional expectation, we have

$$(A.1) \quad \mathbb{E} \left[ \|F(x, \bar{u}) - \hat{F}(Px, \bar{u})\|^2 \right] = \mathbb{E} \left[ \mathbb{E} \left[ \|F(x, \bar{u}) - \hat{F}(Px, \bar{u})\|^2 \mid \bar{u} \right] \right].$$

Letting  $W_{g|\bar{u}} = \mathbb{E} [\nabla_x F(x, \bar{u}) \nabla_x F(x, \bar{u})^T \mid \bar{u}]$  and applying Proposition 2.5 in Zahm et al. [66] to the conditional expectation yields

$$(A.2) \quad \mathbb{E} \left[ \|F(x, \bar{u}) - \hat{F}(Px, \bar{u})\|^2 \right] \leq \mathbb{E} [\text{Tr} [W_{g|\bar{u}}(I - P)\Sigma_{x|\bar{u}}(I - P)^T]].$$

Using our assumption about  $C$ , it follows that

$$(A.3) \quad CW_x - \Sigma_{x|\bar{u}} = C \underbrace{(W_x - \Sigma_x)}_{\mathbb{E}[x]\mathbb{E}[x]^T} + C\Sigma_x - \Sigma_{x|\bar{u}} \succeq 0$$

is positive semidefinite almost surely. Applying this result, linearity of the expectation, and the tower rule, we obtain

$$(A.4) \quad \mathbb{E} \left[ \|F(x, \bar{u}) - \hat{F}(Px, \bar{u})\|^2 \right] \leq \mathbb{E} [\text{Tr} [W_{g|\bar{u}}(I - P)CW_x(I - P)^T]]$$

$$(A.5) \quad = C \text{Tr} [\mathbb{E} [W_{g|\bar{u}}] (I - P)W_x(I - P)^T]$$

$$(A.6) \quad = C \text{Tr} [W_g(I - P)W_x(I - P)^T].$$

It remains to verify that we can take  $C = 1$  in the jointly Gaussian case. Thanks to Proposition 3.13 in Eaton [17], when  $(x, \bar{u})$  has Gaussian distribution, the conditional covariance matrix is given explicitly by

$$(A.7) \quad \Sigma_{x|\bar{u}} = \Sigma_x - \Sigma_{x,\bar{u}}\Sigma_{\bar{u}}^+\Sigma_{x,\bar{u}}^T,$$

where  $\Sigma_{x,\bar{u}} = \mathbb{E}[(x - \mathbb{E}[x])(\bar{u} - \mathbb{E}[\bar{u}])^T]$  and  $\Sigma_{\bar{u}}^+$  denotes the Moore–Penrose pseudoinverse of  $\Sigma_{\bar{u}} = \mathbb{E}[(\bar{u} - \mathbb{E}[\bar{u}])(\bar{u} - \mathbb{E}[\bar{u}])^T]$ . Since  $\Sigma_x - \Sigma_{x|\bar{u}} = \Sigma_{x,\bar{u}}\Sigma_{\bar{u}}^+\Sigma_{x,\bar{u}} \succeq 0$  is positive semidefinite, we can take  $C = 1$ . This completes the proof of the theorem.

**Appendix B. Proof of Theorem 2.5.** We begin by proving Lemma 2.4, which ensures that almost every matrix  $X$  with respect to Lebesgue measure satisfies the hypotheses of the theorem.

*Proof of Lemma 2.4.* Let  $\Phi \in \mathbb{R}^{n \times r}$  have linearly independent columns spanning  $\text{Range}(M^T)$ , and let  $A \in \mathbb{R}^{s \times r}$  be a matrix with linearly independent columns. The functions  $\phi : X \mapsto \det(X^T X)$  and  $\psi : X \mapsto \det(\Phi^T X A)$  are nonconstant polynomials on the Euclidean space  $\mathbb{R}^{n \times s}$ . It follows from the main result in [14] that the level sets of  $\phi$  and  $\psi$  have zero Lebesgue measure. The level set  $\phi^{-1}(0)$  corresponds to matrices  $X \in \mathbb{R}^{n \times s}$  whose columns are linearly dependent. The level set  $\psi^{-1}(0)$  contains every matrix  $X \in \mathbb{R}^{n \times s}$  for which  $\text{rank}(MX) < r$ . Thus, the union  $\phi^{-1}(0) \cup \psi^{-1}(0)$  has Lebesgue measure zero and  $\phi^{-1}(0) \cup \psi^{-1}(0)$  contains every  $X \in \mathbb{R}^{n \times s}$  failing to have linearly independent columns or failing to satisfy  $\text{rank}(MX) = r$ .  $\square$

We introduce some notation and recall some basic concepts that we use throughout our proof of Theorem 2.5. If  $M$  is a matrix, let  $\text{row}_{i_0:i_1}(M)$  denote the submatrix of  $M$  formed from rows  $i_0$  through  $i_1$  with  $i_1 \geq i_0$ . Similarly,  $\text{col}_{j_0:j_1}$  denotes the submatrix formed from columns  $j_0$  through  $j_1$ . The smooth manifold of  $q \times p$  matrices with linearly independent columns is denoted  $\mathbb{R}_*^{q \times p}$ . We recall that the canonical projection map  $\pi_{q,p} : \mathbb{R}_*^{q \times p} \rightarrow \mathcal{G}_{q,p}$  given by  $\pi_{q,p}(\Phi) = \text{Range}(\Phi)$  is a surjective submersion. This follows from the quotient manifold theorem (Theorem 21.10 in Lee

[35]) and the fact that  $\mathcal{G}_{q,p}$  is the quotient of  $\mathbb{R}_*^{q \times p}$  under a free and proper action of the general linear group of invertible  $p \times p$  matrices. Thanks to Theorem 4.29 in [35], a function  $f$  on  $\mathcal{G}_{q,p}$  is smooth if and only if  $f \circ \pi_{q,p}$  is smooth. We also make use of the following method for constructing smooth functions on the Grassmannian. Suppose that  $\bar{f}$  is a smooth function on  $\mathbb{R}_*^{q \times p}$  that is constant on the fibers of  $\pi_{q,p}$ , i.e.,  $\bar{f}$  satisfies  $\bar{f}(\Psi) = \bar{f}(\Psi A)$  for every  $\Psi \in \mathbb{R}_*^{q \times p}$  and invertible  $p \times p$  matrix  $A$ . Thanks to Theorem 4.30 in [35] there is a unique smooth function  $f$  on  $\mathcal{G}_{q,p}$  satisfying  $\bar{f} = f \circ \pi_{q,p}$ , that is,  $\bar{f}(\Psi) = f(\text{Range}(\Psi))$  for every  $\Psi \in \mathbb{R}_*^{q \times p}$ .

LEMMA B.1 (Grassmannian cross-section; see Absil, Mahony, and Sepulchre [1]). *Let  $1 \leq p \leq q$  be integers. For each  $M \in \mathbb{R}_*^{q \times p}$ , the affine “cross-section”*

$$(B.1) \quad S_M = \{Y \in \mathbb{R}_*^{q \times p} : M^T(Y - M) = 0\}$$

*is mapped diffeomorphically onto the open subset of the Grassmannian*

$$(B.2) \quad \mathcal{U}_M = \{\text{Range}(Y) \in \mathcal{G}_{q,p} : \det(M^T Y) \neq 0\}$$

*by the canonical projection  $\pi_{q,p} : Y \mapsto \text{Range}(Y)$ . The smooth inverse is the “cross-section mapping”  $\sigma_M : \mathcal{U}_M \rightarrow S_M$  defined by*

$$(B.3) \quad \sigma_M : \text{Range}(Y) \mapsto Y(M^T Y)^{-1} M^T M.$$

*Proof.* See Absil, Mahony, and Sepulchre [1]. □

Setting  $M = [I_p \ 0]^T$  in the above lemma yields the following useful corollary.

COROLLARY B.2. *Let  $1 \leq p \leq q$  be integers. The subset*

$$(B.4) \quad \mathcal{W}_{q,p} = \left\{ \text{Range}(Y) \in \mathcal{G}_{q,p} : \det(\text{row}_{1:p}(Y)) \neq 0 \right\}$$

*is open in the Grassmannian and diffeomorphic to  $\mathbb{R}^{(q-p) \times p}$ . In particular, a diffeomorphism  $\phi_{q,p} : \mathbb{R}^{(q-p) \times p} \rightarrow \mathcal{W}_{q,p}$  is given by*

$$(B.5) \quad \phi_{q,p} : A \mapsto \text{Range} \left( \begin{bmatrix} I_p \\ A \end{bmatrix} \right).$$

*Proof.* Taking  $M = [I_p \ 0]^T$  in Lemma B.1, we observe that the canonical projection  $\pi_{q,p}$  maps the cross-section

$$(B.6) \quad S_M = \left\{ Y \in \mathbb{R}_*^{q \times p} : \text{row}_{1:p}(Y) = I_p \right\}$$

diffeomorphically onto  $\mathcal{W}_{q,p}$ . The proof is completed by observing that the cross-section is parametrized by the entries in the submatrix  $\text{row}_{(p+1):q}(Y)$ . □

To prove Theorem 2.5, we begin by reducing to the case where the columns of  $X$  are given by the first  $s$  columns of the  $n \times n$  identity matrix, denoted  $E_{n,s} = [I_s \ 0]^T$ . Let  $X = Q_X R$  be a reduced QR factorization where  $Q_X$  is an  $n \times s$  matrix whose columns are an orthonormal basis for the range of  $X$  and  $R$  is an invertible  $s \times s$  matrix. It follows that  $\tilde{F}X = FX$  if and only if  $\tilde{F}Q_X = FQ_X$ . Letting  $Q$  be an  $n \times n$  orthonormal matrix whose first  $s$  columns are given by  $Q_X$ , we have  $Q^T Q_X = E_{n,s}$ .

It follows that  $\tilde{F}Q_X = FQ_X$  if and only if  $\tilde{F}QE_{n,s} = FQE_{n,s}$ . Since  $\text{Range}(\tilde{F}^T) = Q \text{Range}((\tilde{F}Q)^T)$ , we have

$$(B.7) \quad \mathcal{V}_{F,X} = Q\mathcal{V}_{FQ,E_{n,s}},$$

and so it suffices to study  $\mathcal{V}_{FQ,E_{n,s}}$ .

We make use of the following characterization of sets  $\mathcal{V}_{F,X}$ .

LEMMA B.3. *Let  $F$  and  $X$  satisfy the hypotheses of Theorem 2.5. Then*

$$(B.8) \quad \mathcal{V}_{F,X} = \{ \text{Range}(\Phi) : \Phi \in \mathbb{R}^{n \times r} \text{ and } \text{Range}(X^T \Phi) = \text{Range}(X^T F^T) \}.$$

*Proof.* Suppose that there is a matrix  $\tilde{F} \in \mathbb{R}^{m \times n}$  with rank  $r$  satisfying  $\tilde{F}X = FX$ . Letting  $\tilde{F} = U\Sigma\Phi^T$  be a reduced SVD, we have  $\text{Range}(\Phi) = \text{Range}(\tilde{F}^T)$ . Since  $X^T\Phi\Sigma U^T = X^T F^T$ , we obtain  $\text{Range}(X^T F^T) \subset \text{Range}(X^T \Phi)$ . Since  $\Sigma U^T$  is surjective, we conclude that  $\text{Range}(X^T \Phi) = \text{Range}(X^T F^T)$ .

Conversely, suppose that  $\Phi \in \mathbb{R}^{n \times r}$  satisfies  $\text{Range}(X^T \Phi) = \text{Range}(X^T F^T)$ . Then there is an  $r \times m$  matrix  $A$  so that

$$(B.9) \quad X^T \Phi A = X^T F^T.$$

Since  $\text{rank}(FX) = r$ , we must also have  $\text{rank}(\Phi A) = r$ . Setting  $\tilde{F} = A^T \Phi^T$  we conclude that  $\text{Range}(\Phi) = \text{Range}(\tilde{F}^T)$  and  $\tilde{F}X = FX$ .  $\square$

Let  $S_0$  be an  $s \times s$  permutation matrix so that the first  $r$  rows of  $S_0^T E_{n,s}^T Q^T F^T$  are linearly independent. Applying the Lemma B.3 to  $\mathcal{V}_{FQ,E_{n,s}}$ , we obtain

$$(B.10) \quad \mathcal{V}_{FQ,E_{n,s}} = \left\{ \text{Range}(\Phi) : \Phi \in \mathbb{R}^{n \times r} \text{ and } \right. \\ \left. \text{Range}(S_0^T E_{n,s}^T \Phi) = \text{Range}(S_0^T E_{n,s}^T Q^T F^T) \right\}.$$

We observe that the  $n \times n$  permutation matrix

$$(B.11) \quad S = \begin{bmatrix} S_0 & 0 \\ 0 & I_{n-s} \end{bmatrix},$$

satisfies  $SE_{n,s} = E_{n,s}S_0$ , and so

$$(B.12) \quad \mathcal{V}_{FQ,E_{n,s}} = \left\{ S \text{Range}(S^T \Phi) : \Phi \in \mathbb{R}^{n \times r} \text{ and } \right. \\ \left. \text{Range}(E_{n,s}^T S^T \Phi) = \text{Range}(E_{n,s}^T S^T Q^T F^T) \right\}.$$

From this, we obtain  $\mathcal{V}_{FQ,E_{n,s}} = S\mathcal{V}_{FQS,E_{n,s}}$ , further reducing our problem to studying the set  $\mathcal{V}_{FQS,E_{n,s}}$ . To summarize the results so far, we have

$$(B.13) \quad \mathcal{V}_{F,X} = Q\mathcal{V}_{FQ,E_{n,s}} = QS\mathcal{V}_{FQS,E_{n,s}},$$

where  $QS$  is a unitary transformation.

Let  $M \in \mathbb{R}^{s \times r}$  be a matrix with linearly independent columns spanning the  $r$ -dimensional subspace  $\text{Range}(E_{n,s}^T S^T Q^T F^T) \subset \mathbb{R}^s$ . The set we aim to study can be expressed as

$$(B.14) \quad \mathcal{V}_{FQS,E_{n,s}} = \left\{ \text{Range}(\Phi) : \Phi \in \mathbb{R}^{n \times r} \text{ and } \text{Range}(\text{row}_{1:s}(\Phi)) = \text{Range}(M) \right\}.$$



By construction of  $S_0$ , the submatrix  $\text{row}_{1:r}(M)$  is invertible. This implies that  $\text{row}_{1:r}(\Phi)$  is invertible for every  $\Phi \in \mathbb{R}^{n \times r}$  with  $\text{Range}(\text{row}_{1:s}(\Phi)) = \text{Range}(M)$ . Hence, we have

$$(B.15) \quad \mathcal{V}_{FQS, E_{n,s}} \subset \mathcal{W}_{n,r},$$

where  $\mathcal{W}_{n,r}$  is the open submanifold of  $\mathcal{G}_{n,r}$  described in Corollary B.2.

We construct a diffeomorphism  $\mathcal{W}_{n-s+r,r} \rightarrow \mathcal{V}_{FQS, E_{n,s}}$  by first introducing the map  $\bar{f}: \mathbb{R}_*^{(n-s+r) \times r} \rightarrow \mathbb{R}_*^{n \times r}$  defined by

$$(B.16) \quad \bar{f}: \Psi \mapsto \begin{bmatrix} \text{row}_{1:r}(\Psi) \\ \text{row}_{r+1:s}(M) \text{row}_{1:r}(M)^{-1} \text{row}_{1:r}(\Psi) \\ \text{row}_{r+1:n-s+r}(\Psi) \end{bmatrix}.$$

This map is well defined because the matrix appearing on the right has  $r$  columns and contains all of the rows of  $\Psi$ , of which  $r$  are linearly independent. Since this is a linear map of matrices, it is obviously smooth and gives rise to a smooth map  $\pi_{n,r} \circ \bar{f}: \mathbb{R}_*^{(n-s+r) \times r} \rightarrow \mathcal{G}_{n,r}$ . We observe that  $\text{Range}(\bar{f}(\Psi))$  only depends on the range of  $\Psi$  for if  $A$  is an invertible  $r \times r$  matrix, then we have

$$(B.17) \quad \text{Range}(\bar{f}(\Psi A)) = \text{Range}(\bar{f}(\Psi) A) = \text{Range}(\bar{f}(\Psi)).$$

It follows from Theorem 4.30 in [35] that there is a unique smooth map  $f: \mathcal{G}_{n-s+r,r} \rightarrow \mathcal{G}_{n,r}$  satisfying  $\pi_{n,r} \circ \bar{f} = f \circ \pi_{n-s+r,r}$ , that is,

$$(B.18) \quad f(\text{Range}(\Psi)) = \text{Range}(\bar{f}(\Psi))$$

for every  $\Psi \in \mathbb{R}_*^{(n-s+r) \times r}$ . Restricting  $f$  to the subset  $\mathcal{W}_{n-s+r,r} \subset \mathcal{G}_{n-s+r,r}$  will provide the desired diffeomorphism onto  $\mathcal{V}_{FQS, E_{n,s}}$  as we now show.

First, we show that  $f$  is injective on  $\mathcal{G}_{n-s+r,r}$ . If  $f(\text{Range}(\Psi_0)) = f(\text{Range}(\Psi_1))$ , then we have  $\text{Range}(\bar{f}(\Psi_0)) = \text{Range}(\bar{f}(\Psi_1))$ . This implies that there is an invertible  $r \times r$  matrix  $A$  so that  $\bar{f}(\Psi_0) = \bar{f}(\Psi_1)A$ . Taking a subset of rows in (B.16) yields  $\Psi_0 = \Psi_1 A$ . Therefore,  $\text{Range}(\Psi_0) = \text{Range}(\Psi_1)$ , proving that  $f$  is injective.

Next, we show that  $f(\mathcal{W}_{n-s+r,r}) \subset \mathcal{V}_{FQS, E_{n,s}}$ . Consider an element  $\text{Range}(\Psi) \in \mathcal{W}_{n-s+r,r}$  with  $\Psi \in \mathbb{R}_*^{n-s+r \times r}$ . Since  $\text{row}_{1:r}(\Psi)$  is invertible, we have

$$(B.19) \quad \text{Range}(\Psi) = \text{Range}(\Psi \text{row}_{1:r}(\Psi)^{-1} \text{row}_{1:r}(M)).$$

Using this, we obtain

$$(B.20) \quad f(\text{Range}(\Psi)) = \text{Range} \left( \begin{bmatrix} \text{row}_{1:r}(M) \\ \text{row}_{r+1:s}(M) \\ \text{row}_{r+1:n-s+r}(\Psi) \text{row}_{1:r}(\Psi)^{-1} \text{row}_{1:r}(M) \end{bmatrix} \right),$$

which is obviously an element of  $\mathcal{V}_{FQS, E_{n,s}}$  thanks to (B.14).

Finally, to show that  $f$  is a diffeomorphism of  $\mathcal{W}_{n-s+r,r}$  onto  $\mathcal{V}_{FQS, E_{n,s}}$ , we construct a smooth inverse map. The inverse is the restriction to  $\mathcal{V}_{FQS, E_{n,s}} \subset \mathcal{W}_{n,r}$  (recall (B.15)) of the smooth map  $g: \mathcal{W}_{n,r} \rightarrow \mathcal{W}_{n-s+r,r}$  defined by

$$(B.21) \quad g: \underbrace{\text{Range} \left( \begin{bmatrix} I_r \\ \text{row}_{1:s-r}(A) \\ \text{row}_{s-r+1:n-r}(A) \end{bmatrix} \right)}_{\phi_{n,r}(A)} \mapsto \underbrace{\text{Range} \left( \begin{bmatrix} I_r \\ \text{row}_{s-r+1:n-r}(A) \end{bmatrix} \right)}_{\phi_{n-s+r,r}(\text{row}_{s-r+1:n-r}(A))}.$$

This map is well defined and smooth thanks to Corollary B.2.

To show that  $f \circ g$  is the identity on  $\mathcal{V}_{FQS, E_{n,s}}$ , let  $\text{Range}(\Phi) \in \mathcal{V}_{FQS, E_{n,s}}$  for an  $n \times r$  matrix  $\Phi$ . Since  $\text{Range}(\text{row}_{1:s}(\Phi)) = \text{Range}(M)$  and  $\text{row}_{1:r}(M)$  is invertible, it follows that  $\text{row}_{1:r}(\Phi)$  is invertible. By replacing  $\Phi$  with  $\Phi \text{row}_{1:r}(\Phi)^{-1}$ , we may assume without loss of generality that  $\text{row}_{1:r}(\Phi) = I_r$ . With this choice for  $\Phi$ , we observe that  $\text{row}_{1:s}(M) \text{row}_{1:r}(M)^{-1} = \text{row}_{1:s}(\Phi)$ . Using this observation and the definitions of  $f$  and  $g$ , we calculate

$$\begin{aligned}
 (B.22) \quad f \circ g(\text{Range}(\Phi)) &= f \circ g \left( \text{Range} \left( \begin{bmatrix} I_r \\ \text{row}_{r+1:s}(\Phi) \\ \text{row}_{s+1:n}(\Phi) \end{bmatrix} \right) \right) \\
 &= f \left( \text{Range} \left( \begin{bmatrix} I_r \\ \text{row}_{s+1:n}(\Phi) \end{bmatrix} \right) \right) \\
 &= \text{Range} \left( \tilde{f} \left( \begin{bmatrix} I_r \\ \text{row}_{s+1:n}(\Phi) \end{bmatrix} \right) \right) \\
 &= \text{Range} \left( \begin{bmatrix} I_r \\ \text{row}_{r+1:s}(M) \text{row}_{1:r}(M)^{-1} \\ \text{row}_{s+1:n}(\Phi) \end{bmatrix} \right) \\
 &= \text{Range}(\Phi).
 \end{aligned}$$

This proves that  $f \circ g$  is the identity on  $\mathcal{V}_{FQS, E_{n,s}}$ .

To prove that  $g \circ f$  is the identity on  $\mathcal{W}_{n-s+r}$ , we recall that  $f(\mathcal{W}_{n-s+r,r}) \subset \mathcal{V}_{FQS, E_{n,s}}$ . Combining this with the result that  $f \circ g$  is the identity on  $\mathcal{V}_{FQS, E_{n,s}}$  yields

$$(B.23) \quad f \circ g \circ f = f.$$

Since  $f$  is injective, it follows that  $g \circ f$  is the identity on  $\mathcal{W}_{n-s+r,r}$ , proving that  $\mathcal{V}_{FQS, E_{n,s}}$  is diffeomorphic to  $\mathcal{W}_{n-s+r,r}$ , and hence to  $\mathbb{R}^{(n-s) \times r}$ .

To estimate the diameter of  $\mathcal{V}_{F,X}$  in the Grassmannian  $\mathcal{G}_{n,r}$ , it suffices to study  $\mathcal{V}_{FQS, E_{n,s}}$  since  $\mathcal{V}_{F,X}$  is related to  $\mathcal{V}_{FQS, E_{n,s}}$  by a unitary transformation. The diffeomorphism  $h = f \circ \phi_{n-s+r,r} : \mathbb{R}^{(n-s) \times r} \rightarrow \mathcal{V}_{FQS, E_{n,s}}$  constructed above can be written explicitly as

$$(B.24) \quad h : A \mapsto \text{Range} \left( \begin{bmatrix} I_r \\ \text{row}_{r+1:s}(M) \text{row}_{1:r}(M)^{-1} \\ A \end{bmatrix} \right).$$

Letting the columns of  $U \in \mathbb{R}^{s \times r}$  be an orthonormal basis for

$$\text{Range} \left( \begin{bmatrix} I_r \\ \text{row}_{r+1:s}(M) \text{row}_{1:r}(M)^{-1} \end{bmatrix} \right),$$

a change of coordinates on  $\mathbb{R}^{(n-s) \times r}$  yields a new diffeomorphism  $\tilde{h} =: \mathbb{R}^{(n-s) \times r} \rightarrow \mathcal{V}_{FQS, E_{n,s}}$  given by

$$(B.25) \quad \tilde{h} : A \mapsto \text{Range} \left( \begin{bmatrix} U \\ A \end{bmatrix} \right).$$

Let  $p = \min\{n-s, r\}$ , and let  $U_0$  denote the first  $p$  columns of  $U$  and let  $U_1$  denote the remaining  $r-p$  columns of  $U$ . For each  $\varepsilon \in (0, 1]$ , we define a subspace

$$(B.26) \quad V_\varepsilon = \tilde{h} \left( \begin{bmatrix} \frac{\sqrt{1-\varepsilon^2}}{\varepsilon} I_p & 0 \\ 0 & 0 \end{bmatrix} \right) = \text{Range} \left( \begin{bmatrix} \varepsilon U_0 & U_1 \\ \sqrt{1-\varepsilon^2} I_p & 0_{p \times (r-p)} \\ 0_{(n-s-p) \times p} & 0_{(n-s-p) \times (r-p)} \end{bmatrix} \right),$$

where we have indicated dimensions of zero entries for clarity. We observe that the matrix on the right, which we call  $\Phi_\varepsilon$ , has orthonormal columns forming a basis for  $V_\varepsilon$ . Since we have

$$(B.27) \quad \Phi_1^T \Phi_\varepsilon = \begin{bmatrix} \varepsilon I_p & 0 \\ 0 & I_{r-p} \end{bmatrix},$$

the principal angles  $\theta_i(V_1, V_\varepsilon)$  between these subspaces (see Björck and Golub [11]) satisfy

$$(B.28) \quad \cos(\theta_i(V_1, V_\varepsilon)) = \begin{cases} 1, & 1 \leq i \leq r-p, \\ \varepsilon, & r-p+1 \leq i \leq r. \end{cases}$$

Therefore, the geodesic distance between the subspaces (see Bendokat, Zimmermann, and Absil [5]) approaches

$$(B.29) \quad d(V_1, V_\varepsilon) = \sqrt{\sum_{i=1}^r \theta_i(V_1, V_\varepsilon)^2} \rightarrow \frac{\pi}{2} \sqrt{p}$$

as  $\varepsilon \rightarrow 0$ . This completes the proof of the theorem.

### Appendix C. Proof of Theorem 2.7.

*Proof of Lemma 2.6.* In the particular case when  $K \in C^2(\bar{\mathcal{X}} \times \bar{\mathcal{X}})$ , it is shown in step 2 of Theorem 1 in Zhou [68] ( $\alpha = 0$  case) that the difference quotients converge

$$(C.1) \quad \frac{1}{t} (K_{x+te_j} - K_x) \rightarrow (\partial^{e_j} K)_x \quad \text{in } \mathcal{H} \quad \text{as } t \rightarrow 0$$

for every  $x \in \mathcal{X}$ . Using these partial derivatives, we establish that  $\Phi_K$  is Fréchet differentiable by employing Proposition 1.1.4 in Kesavan [29]. To apply this result, it remains to show that each of the maps  $x \mapsto (\partial^{e_j} K)_x$  are continuous. By (2.29) (see Theorem 1 in [68]) and symmetry of the reproducing kernel, we have

$$(C.2) \quad \langle (\partial^{e_i} K)_x, (\partial^{e_j} K)_y \rangle = \partial^{e_i} (\partial^{e_j} K)_y(x) = \partial^{(e_j, e_i)} K(y, x) = \partial^{(e_i, e_j)} K(x, y),$$

which is a continuous function of  $x, y \in \mathcal{X}$ . Therefore,

$$(C.3) \quad \|(\partial^{e_j} K)_x - (\partial^{e_j} K)_y\|^2 = \partial^{(e_j, e_j)} K(x, x) - 2\partial^{(e_j, e_j)} K(x, y) + \partial^{(e_j, e_j)} K(y, y),$$

which approaches 0 as  $y \rightarrow x$  in  $\mathcal{X}$ . This proves that  $x \mapsto (\partial^{e_j} K)_x$  is continuous on  $\mathcal{X}$  and so  $\Phi_K$  is Fréchet differentiable on  $\mathcal{X}$ .

Since the feature map is Fréchet differentiable, the first expression in (2.30) follows from (C.1) and Proposition 1.1.3 in [29]. Given any  $v \in \mathbb{R}^n$  and  $f \in \mathcal{H}$ , (2.29) can be used to show that

$$(C.4) \quad \langle D\Phi_K(x)v, f \rangle = \sum_{j=1}^n v_j \langle \partial^{e_j} K \rangle_x, f \rangle = \sum_{j=1}^n v_j \partial^{e_j} f(x) = \langle v, \nabla f(x) \rangle,$$

which proves the second first expression in (2.30).

Using (2.30) and (C.2), we observe that

$$(C.5) \quad H(x, y) := D\Phi_K(x)^* D\Phi_K(y) = \begin{bmatrix} \partial^{(e_1, e_1)} K(x, y) & \cdots & \partial^{(e_1, e_n)} K(x, y) \\ \vdots & \ddots & \vdots \\ \partial^{(e_n, e_1)} K(x, y) & \cdots & \partial^{(e_n, e_n)} K(x, y) \end{bmatrix}$$

is a continuous matrix-valued function of  $x, y \in \mathcal{X}$ . Therefore,

$$\begin{aligned}
 \|\mathrm{D}\Phi_K(x) - \mathrm{D}\Phi_K(y)\|_{\mathrm{op}}^2 &= \sup_{\substack{v \in \mathbb{R}^n: \\ \|v\| \leq 1}} \|\mathrm{D}\Phi_K(x)v - \mathrm{D}\Phi_K(y)v\|^2 \\
 (C.6) \qquad &= \sup_{\substack{v \in \mathbb{R}^n: \\ \|v\| \leq 1}} v^T (H(x, x) - H(x, y) - H(y, x) + H(y, y))v \\
 &\leq \|H(x, x) - H(x, y) - H(y, x) + H(y, y)\|_{\mathrm{op}}
 \end{aligned}$$

which approaches 0 as  $y \rightarrow x$  in  $\mathcal{X}$ , proving that  $x \mapsto \mathrm{D}\Phi_K(x)$  is continuous with respect to the operator norm.  $\square$

*Proof of Theorem 2.7.* Since the kernel is continuous on  $\bar{\mathcal{X}}$ , so is the feature map. Because the feature map is continuous and injective on the compact set  $\bar{\mathcal{X}}$ , Corollary 13.27 in Sutherland [60] shows that  $\Phi_K$  is a topological embedding of  $\bar{\mathcal{X}}$  into  $\mathcal{H}$ . Since the restriction of a topological embedding is still an embedding,  $\Phi_K$  is a topological embedding of  $\mathcal{X}$ . Let  $\mathcal{X}'$  denote the image of  $\mathcal{X}$  under  $\Phi_K$ , and let  $\Psi_K : \mathcal{X}' \rightarrow \mathcal{X}$  denote the continuous inverse of  $\Phi_K|_{\mathcal{X}}$  on its image. To show that  $\Phi_K$  is a  $C^1$  diffeomorphism of  $\mathcal{X}$  onto  $\mathcal{X}'$ , it remains to show that  $\Psi_K$  is  $C^1$ .

Choose  $x \in \mathcal{X}$ . The expression (2.31) for the derivative Gram matrix  $G(x) = \mathrm{D}\Phi_K(x)^* \mathrm{D}\Phi_K(x)$  in terms of the kernel is verified directly using (2.30) and symmetry of the kernel. Since  $G(x)$  is positive-definite,  $\mathrm{D}\Phi_K(x)$  is injective. Moreover, the subspace  $\mathrm{Range}(\mathrm{D}\Phi_K(x))$  is finite-dimensional, hence it is closed in  $\mathcal{H}$ . Consequently, Proposition 3.2.8 in Margalef-Roig and Dominguez [40] shows that  $\Phi_K$  is an immersion at  $x$ . By Proposition 3.2.13 in [40] there is an open neighborhood  $\mathcal{U}$  of  $x$  in  $\mathcal{X}$  so that  $\Phi_K(\mathcal{U})$  is a  $C^1$  submanifold of  $\mathcal{H}$  and  $\Phi_K : \mathcal{U} \rightarrow \Phi_K(\mathcal{U})$  is a  $C^1$  diffeomorphism. Because  $\Phi_K$  is a topological embedding,  $\Phi_K(\mathcal{U})$  is an open subset of  $\mathcal{X}'$  and the inverse of  $\Phi_K|_{\mathcal{U}}$  on  $\Phi_K(\mathcal{U})$  agrees with  $\Psi_K$ . Therefore,  $\Psi_K$  is  $C^1$  on each  $\Phi_K(\mathcal{U})$ . Since these sets cover  $\mathcal{X}'$ , we have shown that  $\Psi_K$  is  $C^1$ , and so  $\Psi_K : \mathcal{X}' \rightarrow \mathcal{X}$  is a  $C^1$  diffeomorphism.

Consequently,  $F' = F \circ \Psi_K$  is well defined and Fréchet differentiable on  $\mathcal{X}'$ . Let  $\phi$  be any real-valued continuously differentiable function on  $\mathbb{R}^m$ . Differentiating the relation  $\phi \circ F = \phi \circ F' \circ \Phi_K$  at  $x \in \mathcal{X}$  gives

$$(C.7) \qquad \nabla(\phi \circ F)(x) = \mathrm{D}\Phi_K(x)^* \nabla(\phi \circ F')(K_x).$$

Since  $\nabla(\phi \circ F')(K_x)$  is an element of  $T_{K_x} \mathcal{X}' = \mathrm{Range}(\mathrm{D}\Phi_K(x))$ , there is a vector  $w \in \mathbb{R}^n$  so that

$$(C.8) \qquad \nabla(\phi \circ F')(K_x) = \mathrm{D}\Phi_K(x)w.$$

Since the derivative Gram matrix  $G(x)$  is invertible, the above expressions can be solved for  $w$ , yielding (2.32).  $\square$

## REFERENCES

- [1] P.-A. ABSIL, R. MAHONY, AND R. SEPULCHRE, *Riemannian geometry of Grassmann manifolds with a view on algorithmic computation*, Acta Appl. Math., 80 (2004), pp. 199–220.
- [2] S. AHUJA AND C. W. ROWLEY, *Feedback control of unstable steady states of flow past a flat plate using reduced-order estimators*, J. Fluid Mech., 645 (2010), pp. 447–478.
- [3] J. BAKER, M. EMBREE, AND J. SABINO, *Fast singular value decay for Lyapunov solutions with nonnormal coefficients*, SIAM J. Matrix Anal. Appl., 36 (2015), pp. 656–668, <https://doi.org/10.1137/140993867>.

- [4] A. BARBAGALLO, D. SIPP, AND P. J. SCHMID, *Closed-loop control of an open cavity flow using reduced-order models*, J. Fluid Mech., 641 (2009), pp. 1–50.
- [5] T. BENDOKAT, R. ZIMMERMANN, AND P.-A. ABSIL, *A Grassmann Manifold Handbook: Basic Geometry and Computational Aspects*, preprint, <https://arxiv.org/abs/2011.13699>, 2020.
- [6] P. BENNER AND P. GOYAL, *Balanced Truncation Model Order Reduction for Quadratic-Bilinear Control Systems*, preprint, <https://arxiv.org/abs/1705.00160>, 2017.
- [7] P. BENNER, P. GOYAL, AND S. GUGERCIN,  *$\mathcal{H}_2$ -quasi-optimal model order reduction for quadratic-bilinear control systems*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 983–1032, <https://doi.org/10.1137/16M1098280>.
- [8] P. BENNER, S. GUGERCIN, AND K. WILLCOX, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Rev., 57 (2015), pp. 483–531, <https://doi.org/10.1137/130932715>.
- [9] A. BERLINET AND C. THOMAS-AGNAN, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Springer, 2011.
- [10] D. BIGONI, Y. MARZOUK, C. PRIEUR, AND O. ZAHM, *Nonlinear Dimension Reduction for Surrogate Modeling Using Gradient Information*, preprint, arXiv:2102.10351, 2021.
- [11] Å. BJÖRCK AND G. H. GOLUB, *Numerical methods for computing angles between linear subspaces*, Math. Comput., 27 (1973), pp. 579–594.
- [12] J. BOUVRIE AND B. HAMZI, *Balanced Reduction of Nonlinear Control Systems in Reproducing Kernel Hilbert Space*, preprint, <https://arxiv.org/abs/1011.2952>, 2010.
- [13] J. BOUVRIE AND B. HAMZI, *Kernel methods for the approximation of nonlinear systems*, SIAM J. Control Optim., 55 (2017), pp. 2460–2492, <https://doi.org/10.1137/14096815X>.
- [14] R. CARON AND T. TRAYNOR, *The Zero Set of a Polynomial*, WSMR report, 2005, <https://www.researchgate.net/profile/Richard-Caron-3/publication/281285245-The-Zero-Set-of-a-Polynomial/links/55df56b608aeb1a7cc1a043/The-Zero-Set-of-a-Polynomial.pdf>.
- [15] L. H. Y. CHEN, *An inequality for the multivariate normal distribution*, J. Multivariate Anal., 12 (1982), pp. 306–315.
- [16] P. G. CONSTANTINE, *Active Subspaces: Emerging Ideas for Dimension Reduction in Parameter Studies*, SIAM, Philadelphia, 2015, <https://doi.org/10.1137/1.9781611973860>.
- [17] M. L. EATON, *Multivariate Statistics: A Vector Space Approach*, IMS Lecture Notes Monogr. Ser. 53, Institute of Mathematical Statistics, Beachwood, OH, 2007.
- [18] C. ECKART AND G. YOUNG, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211–218.
- [19] K. FUJIMOTO AND D. TSUBAKINO, *Computation of nonlinear balanced realization and model reduction based on Taylor series expansion*, Syst. Control Lett., 57 (2008), pp. 283–289.
- [20] A. GHADAMI AND B. I. EPUREANU, *Data-driven prediction in dynamical systems: Recent developments*, Philos. Trans. R. Soc. A, 380 (2022), 20210213.
- [21] N. HALKO, P. G. MARTINSSON, AND J. A. TROPP, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288, <https://doi.org/10.1137/090771806>.
- [22] T. HOFMANN, B. SCHÖLKOPF, AND A. J. SMOLA, *Kernel methods in machine learning*, Ann. Statist., 36 (2008), pp. 1171–1220.
- [23] P. HOLMES, J. L. LUMLEY, G. BERKOOZ, AND C. W. ROWLEY, *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*, Cambridge University Press, Cambridge, 2012.
- [24] J. T. HOLODNAK, I. C. F. IPSEN, AND R. C. SMITH, *A probabilistic subspace bound with application to active subspaces*, SIAM J. Matrix Anal. Appl., 39 (2018), pp. 1208–1220, <https://doi.org/10.1137/17M1141503>.
- [25] M. ILAK, *Model Reduction and Feedback Control of Transitional Channel Flow*, Ph.D. thesis, Princeton University, Princeton, NJ, 2009.
- [26] M. ILAK, S. BAGHERI, L. BRANDT, C. W. ROWLEY, AND D. S. HENNINGSON, *Model reduction of the nonlinear complex Ginzburg–Landau equation*, SIAM J. Appl. Dyn. Syst., 9 (2010), pp. 1284–1302, <https://doi.org/10.1137/100787350>.
- [27] S. J. ILLINGWORTH, A. S. MORGANS, AND C. W. ROWLEY, *Feedback control of flow resonances using balanced reduced-order models*, J. Sound Vibration, 330 (2011), pp. 1567–1581.
- [28] Y. KAWANO AND J. M. SCHERPEN, *Empirical differential Gramians for nonlinear model reduction*, Automatica J. IFAC, 127 (2021), 109534.
- [29] S. KESAVAN, *Nonlinear Functional Analysis: A First Course*, 2nd ed., Springer, Singapore, 2022.
- [30] V. KOLTCHINSKII AND K. LOUNICI, *Concentration inequalities and moment bounds for sample covariance operators*, Bernoulli, 23 (2017), pp. 110–133.

- [31] B. KRAMER, S. GUGERCIN, AND J. BORGGAARD, *Nonlinear Balanced Truncation: Part 1—Computing Energy Functions*, preprint, <https://arxiv.org/abs/2209.07645>, 2022.
- [32] B. KRAMER, S. GUGERCIN, AND J. BORGGAARD, *Nonlinear Balanced Truncation: Part 2—Model Reduction on Manifolds*, preprint, <https://arxiv.org/abs/2302.02036>, 2023.
- [33] S. LALL, J. E. MARSDEN, AND S. GLAVAŠKI, *A subspace approach to balanced truncation for model reduction of nonlinear control systems*, Internat. J. Robust Nonlinear Control, 12 (2002), pp. 519–535.
- [34] A. J. LAUB, M. T. HEATH, C. C. PAIGE, AND R. C. WARD, *Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms*, IEEE Trans. Automat. Control, 32 (1987), pp. 115–122.
- [35] J. M. LEE, *Introduction to Smooth Manifolds*, 2nd ed., Grad. Texts in Math. 218, Springer, New York, 2013.
- [36] K. LEE AND K. T. CARLBERG, *Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders*, J. Comput. Phys., 404 (2020), 108973.
- [37] J. L. LUMLEY, *The structure of inhomogeneous turbulent flows*, Atmos. Turbul. Radio Wave Propagation, (1967), pp. 166–177.
- [38] Z. MA, *Reduction and Reconstruction Methods for Simulation and Control of Fluids*, Ph.D. thesis, Princeton University, Princeton, NJ, 2010.
- [39] Z. MA, C. W. ROWLEY, AND G. TADMOR, *Snapshot-based balanced truncation for linear time-periodic systems*, IEEE Trans. Automat. Control, 55 (2010), pp. 469–473.
- [40] J. MARGALEF-ROIG AND E. O. DOMINGUEZ, *Differential Topology*, North-Holland Math. Stud. 173, Elsevier, 1992.
- [41] L. MIRSKY, *Symmetric gauge functions and unitarily invariant norms*, Quart. J. Math. Oxford Ser. (2), 11 (1960), pp. 50–59.
- [42] B. MOORE, *Principal component analysis in linear systems: Controllability, observability, and model reduction*, IEEE Trans. Automat. Control, 26 (1981), pp. 17–32.
- [43] R. OLIVEIRA, *Sums of random Hermitian matrices and an inequality by Rudelson*, Electron. Commun. Probab., 15 (2010), pp. 203–212.
- [44] S. E. OTTO, A. PADOVAN, AND C. W. ROWLEY, *Optimizing oblique projections for nonlinear systems using trajectories*, SIAM J. Sci. Comput., 44 (2022), pp. A1681–A1702, <https://doi.org/10.1137/21M1425815>.
- [45] A. PADOVAN AND C. W. ROWLEY, *Continuous-time balanced truncation for time-periodic fluid flows using frequential Gramians*, preprint, <https://arxiv.org/abs/2208.13245>, 2022.
- [46] M. T. PARENTE, J. WALLIN, AND B. WOHLMUTH, *Generalized bounds for active subspaces*, Electron. J. Stat., 14 (2020), pp. 917–943.
- [47] F. PEDREGOSA, G. VAROQUAUX, A. GRAMFORT, V. MICHEL, B. THIRION, O. GRISEL, M. BLONDEL, P. PRETTENHOFER, R. WEISS, V. DUBOURG, J. VANDERPLAS, A. PASSOS, D. COURNAPEAU, M. BRUCHER, M. PERROT, AND E. DUCHESNAY, *Scikit-learn: Machine learning in Python*, J. Mach. Learn. Res., 12 (2011), pp. 2825–2830.
- [48] T. PENZL, *A cyclic low-rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (1999), pp. 1401–1418, <https://doi.org/10.1137/S1064827598347666>.
- [49] F. ROMOR, M. TEZZELE, A. LARIO, AND G. ROZZA, *Kernel-based Active Subspaces with Application to CFD Parametric Problems Using Discontinuous Galerkin Method*, preprint, <https://arxiv.org/abs/2008.12083>, 2020.
- [50] C. W. ROWLEY, *Model reduction for fluids using balanced proper orthogonal decomposition*, Internat. J. Bifur. Chaos, 15 (2005), pp. 997–1013.
- [51] C. W. ROWLEY AND S. T. M. DAWSON, *Model reduction for flow analysis and control*, Annu. Rev. Fluid Mech., 49 (2017), pp. 387–417.
- [52] M. RUDELSON AND R. VERSHYNIN, *Sampling from large matrices: An approach through geometric functional analysis*, J. ACM, 54 (2007), 21.
- [53] H. SANDBERG AND A. RANTZER, *Balanced truncation of linear time-varying systems*, IEEE Trans. Automat. Control, 49 (2004), pp. 217–229.
- [54] J. M. A. SCHERPEN, *Balancing for nonlinear systems*, Syst. Control Lett., 21 (1993), pp. 143–153.
- [55] P. J. SCHMID AND D. S. HENNINGSON, *Stability and Transition in Shear Flows*, Appl. Math. Sci. 142, Springer-Verlag, New York, 2001.
- [56] B. SCHÖLKOPF, A. SMOLA, AND K.-R. MÜLLER, *Nonlinear component analysis as a kernel eigenvalue problem*, Neural Comput., 10 (1998), pp. 1299–1319.
- [57] J. R. SINGLER, *Optimality of balanced proper orthogonal decomposition for data reconstruction*, Numer. Funct. Anal. Optim., 31 (2010), pp. 852–869.
- [58] J. R. SINGLER, *Optimality of balanced proper orthogonal decomposition for data reconstruction II: Further approximation results*, J. Math. Anal. Appl., 421 (2015), pp. 1006–1020.

- [59] L. SIROVICH, *Turbulence and the dynamics of coherent structures. I. Coherent structures*, Quart. Appl. Math., 45 (1987), pp. 561–571.
- [60] W. A. SUTHERLAND, *Introduction to Metric and Topological Spaces*, 2nd ed., Oxford University Press, 2009.
- [61] L. N. TREFETHEN, A. E. TREFETHEN, S. C. REDDY, AND T. A. DRISCOLL, *Hydrodynamic stability without eigenvalues*, Science, 261 (1993), pp. 578–584.
- [62] E. I. VERRIEST AND W. S. GRAY, *Nonlinear balanced realizations*, in Proceedings of the 43rd IEEE Conference on Decision and Control (CDC), Vol. 2, IEEE, 2004, pp. 1164–1169.
- [63] R. VERSHYNIN, *Introduction to the non-asymptotic analysis of random matrices*, in Compressed Sensing: Theory and Applications, Y. C. Eldar and G. Kutyniok, eds., Cambridge University Press, Cambridge, 2012, pp. 210–268.
- [64] R. VERSHYNIN, *High-dimensional Probability: An Introduction with Applications in Data Science*, Cambridge University Press, Cambridge, 2018.
- [65] Y.-C. WONG, *Differential geometry of Grassmann manifolds*, Proc. Natl. Acad. Sci., 57 (1967), pp. 589–594.
- [66] O. ZAHM, P. G. CONSTANTINE, C. PRIEUR, AND Y. M. MARZOUK, *Gradient-based dimension reduction of multivariate vector-valued functions*, SIAM J. Sci. Comput., 42 (2020), pp. A534–A558, <https://doi.org/10.1137/18M1221837>.
- [67] O. ZAHM, T. CUI, K. LAW, A. SPANTINI, AND Y. MARZOUK, *Certified dimension reduction in nonlinear Bayesian inverse problems*, Math. Comput., 91 (2022), pp. 1789–1835.
- [68] D.-X. ZHOU, *Derivative reproducing properties for kernel methods in learning theory*, J. Comput. Appl. Math., 220 (2008), pp. 456–463.