2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

Enforcing structure in data-driven reduced modeling through nested Operator Inference

Nicole Aretz and Karen Willcox

***Abstract*— We introduce the data-driven nested Operator Inference method for learning projection-based reduced-order models (ROMs) from snapshot data of high-dimensional dynamical systems. These ROMs achieve significant computational speed-up by exploiting the intrinsic low-dimensionality of the full-order solution trajectory through projection onto a low-dimensional subspace. Our nested Operator Inference approach builds upon a nested structure of the projection-based reduced-order matrices and a hierarchy within the subspace's basis vectors to partition the Operator Inference learning problem into multiple regression problems defined on subspaces. Each regression problem is provably better conditioned than when all reduced-order operators are learned together, reducing the need for additional regularization. Since only $\mathcal{O}(1)$ unknowns are learned at a time, nested Operator Inference is particularly applicable to higher-order polynomial systems. We demonstrate our method for the shallow ice equations with eighth order polynomial operators.**

I. Introduction

We present a new, nested operator inference (OpInf) approach for learning reduced-order models (ROMs) for high-dimensional dynamical systems from snapshot data. ROMs are essential for enabling real-time predictions and uncertainty quantification in computationally expensive physical simulations. Projection-based ROMs with low-dimensional subspaces are well studied, with extensive analysis of stability properties, error bounds, and generalizability outside the training regime, and remain physically interpretable through their connection to the governing equations [1], [2], [3], [4] — a major advantage compared to black-box machine learning approaches. For applications with commercial or legacy code where a projection-based ROM cannot be constructed without intrusive access to the full-order operators, the OpInf method [5] infers the ROM from snapshot data of the full-order model (FOM) and the structure of the physical governing equations. The data-driven OpInf approach is physics-based, analyzed under common conditions [5], [6], [7], [8], [9], [10], [11], and has been applied successfully in various applications [12], [13], [14], [15], [16], [17].

While OpInf has been extended for various research directions including non-polynomial FOMs [18], [19], Hamiltonian systems [20], [21], [22], partial differential equations (PDEs) [23], learning on manifolds [24], [25], and parametric [26] or sparse [27] operators, a major challenge throughout is guaranteeing the stability of the OpInf learning problem. Regularization is one approach to address this challenge, but can come to dominate the learned ROM in low-data regimes, particularly for high-order polynomial FOMs. Our work addresses this challenge by imposing additional structure on the OpInf learning problem. We recognize that projection-based ROMs are nested, and we exploit the hierarchy of proper orthogonal decomposition (POD) basis vectors to partition the OpInf regression problem into multiple smaller problems, each of which is provably better conditioned. This approach is particularly applicable to nonlinear systems with high-order polynomial terms, since it shifts the number of entries learned at a time from polynomial scaling in the reduced dimension to $\mathcal{O}(1)$. This is an improvement because it allows increasing the reduced dimension to improve the accuracy of the ROM without diminishing the numerical stability of the learning problem.

Section II introduces the setting for the OpInf learning problem. In Section III we explain the reasoning behind the nested OpInf method, outlining the hierarchy in the POD basis, the nested structure of projection-based reduced-order operators, and the stability improvement of learning in subspaces. Section IV presents the nested OpInf algorithm, and Section V demonstrates the approach for the shallow ice equations—a nonlinear system with eighth-order polynomial terms. Section VI concludes the paper.

Notation: The most important variable names are (in order of appearance) the state $\mathbf{x}$, the polynomial operators $\mathbf{A}_\ell$, the reduced space $\mathbf{V}$ and its dimension r, the data matrix $\mathbf{D}$, and the right-hand-side matrix $\mathbf{R}$. Reduced-order variables (e.g., the reduced-order state $\hat{\mathbf{x}}$, and the reduced-order polynomial operators $\hat{\mathbf{A}}_\ell$) are distinguished from full-order variables by the hat notation. Superscripts are reserved for matrix and vector transposes (indicated by $\top$), and for exponential operations. Unless stated otherwise, lower-case subscripts are enumerations; we use the index ℓ to enumerate polynomial orders, k for time steps, and i, j for the position of entries within a matrix or vector. In particular, $\mathbf{x}_k$ denotes the k-th state snapshot. For the reduced space $\mathbf{V}$, an indexed set $\mathcal{I} \subset \{1, \ldots, r\}$ denotes the restriction to those columns with numbers in $\mathcal{I}$. For example, $\mathbf{V}_{\{1,3\}}$ contains the first and third column of $\mathbf{V}$. For all other matrices and vectors, the subscript $\mathcal{I}$ indicates a definition that employs the restricted basis $\mathbf{V}_\mathcal{I}$ in place of $\mathbf{V}$. For example, if $\hat{\mathbf{A}}_1 = \mathbf{V}^\top \mathbf{A}_1 \mathbf{V}$, then $\hat{\mathbf{A}}_{1,\mathcal{I}} = \mathbf{V}_\mathcal{I}^\top \mathbf{A}_1 \mathbf{V}_\mathcal{I}$.

Nicole Aretz and Karen Willcox are with the Oden Institute for Computational Engineering & Sciences, University of Texas at Austin, Austin, Texas, USA, nicole.aretz@austin.utexas.edu, kwillcox@oden.utexas.edu

The authors would like to thank John Jakeman for providing the model used in the numerical experiments. This work was supported in parts by the Department of Energy grants DE-SC0021239 and DE-SC002317, and the Air Force Office of Scientific Research grant FA9550-21-1-0084.

II. Learning ROMs via Operator Inference

Our goal is to learn a projection-based ROM with Galerkin structure from snapshot data of a FOM, without having access to the individual full-order operators. We consider a FOM of the form

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t)) := \sum_{\ell \in \mathcal{L}} \mathbf{A}_\ell \underbrace{(\mathbf{x}(t) \otimes \cdots \otimes \mathbf{x}(t))}_{\ell \text{ times}}, \quad (1)$$

where $\mathcal{L} = \{\ell_{\min} \leq \cdots \leq \ell_{\max}\} \subset \mathbb{N}$ such that $\mathbf{A}_\ell \in \mathbb{R}^{n \times n^\ell}$ are operators multiplying polynomial terms[1] of order $\ell \geq \ell_{\min} \geq 1$. For example, $\mathbf{A}_1$ denotes the $n \times n$ matrix corresponding to a linear term on the right-hand side of (1), $\mathbf{A}_2$ denotes the $n \times n^2$ matricized tensor corresponding to a quadratic term, and so on up to order $\ell_{\max}$. The FOM (1) describes the time evolution of a state vector $\mathbf{x}(t) \in \mathbb{R}^n$ from initial condition $\mathbf{x}(0) = \mathbf{x}_0$. We target high-dimensional systems ($n > 10^6$), such as those arising from discretization of PDEs, whose state solution $\mathbf{x}(t) \in \mathbb{R}^n$ is returned as part of an expensive FOM solve.

Given k_{tr} training snapshots[2] $\mathbf{x}_k = \mathbf{x}(t_k)$, $k = 1, \ldots, k_{\text{tr}}$, obtained by solving the FOM for time steps $t_1, \ldots, t_{k_{\text{tr}}}$, we construct an r-dimensional reduced space $\mathit{span}(\mathbf{v}_1, \ldots, \mathbf{v}_r)$ via POD with orthonormal basis functions $\mathbf{v}_i^\top \mathbf{v}_j = \delta_{i,j}$ where $\delta_{i,j}$ is the Kronecker delta. Letting $\mathbf{V} := (\mathbf{v}_1, \ldots, \mathbf{v}_r) \in \mathbb{R}^{n \times r}$ denote the basis matrix, with column j containing the jth basis vector $\mathbf{v}_j$, the Galerkin projection of (1) onto the subspace spanned by $\mathbf{V}$ is

$$\dot{\hat{\mathbf{x}}}(t) = \sum_{\ell \in \mathcal{L}} \hat{\mathbf{A}}_\ell \hat{\mathbf{x}}(t)^\ell, \quad (2)$$

to be solved for a reduced-order state $\hat{\mathbf{x}}(t) \in \mathbb{R}^r$ with initial condition $\hat{\mathbf{x}}(0) = \mathbf{V}^\top \mathbf{x}_0 \in \mathbb{R}^r$. Here, we adopt the notation from [29] where $\hat{\mathbf{x}}^\ell \in \mathbb{R}^{r^{(\ell)}}$, $r^{(\ell)} := \binom{r+\ell-1}{\ell}$, is an ℓ-fold repeated Kronecker product of any single vector $\hat{\mathbf{x}} \in \mathbb{R}^r$ with all redundant entries removed, and $\hat{\mathbf{A}}_\ell \in \mathbb{R}^{r \times r^{(\ell)}}$ is defined by the form of the projected polynomial terms, such that

$$\hat{\mathbf{A}}_\ell \hat{\mathbf{x}}^\ell = \mathbf{V}^\top \mathbf{A}_\ell \underbrace{(\mathbf{V}\hat{\mathbf{x}} \otimes \cdots \otimes \mathbf{V}\hat{\mathbf{x}})}_{\ell-\text{times}} \quad \forall\ \hat{\mathbf{x}} \in \mathbb{R}^r. \quad (3)$$

The OpInf approach infers the reduced-order operators $\hat{\mathbf{A}}_\ell$, directly from the snapshot data, without requiring access to the FOM operators $\mathbf{A}_\ell$ (see [5]). We define the matrix of operators to be learned as

$$\hat{\mathbf{O}} := \left(\hat{\mathbf{A}}_{\ell_{\min}}, \ldots, \hat{\mathbf{A}}_{\ell_{\max}}\right) \in \mathbb{R}^{r \times r_{\text{tot}}}, \quad (4)$$

where $r_{\text{tot}} = \sum_{\ell \in \mathcal{L}} r^{(\ell)}$ is the total number of columns over all ROM operators (noting that r_{tot} will scale with order $r^{\ell_{\max}}$). The OpInf learning problem is obtained from (2) by replacing $\hat{\mathbf{x}}(t_k)$ with the projection $\mathbf{p}_k := \mathbf{V}^\top \mathbf{x}_k \in \mathbb{R}^r$ of the snapshot $\mathbf{x}_k$, approximating the time derivative of snapshot $\mathbf{x}_k$ as $\dot{\hat{\mathbf{x}}}(t_k) \approx \mathbf{V}^\top \dot{\mathbf{x}}_k$, and minimizing the ensuing misfit over all snapshots as a least squares problem:

$$\min_{\hat{\mathbf{O}}} \|\mathbf{D}\hat{\mathbf{O}}^\top - \mathbf{R}\|_F^2. \quad (5)$$

Here, the kth row of the data matrix $\mathbf{D} \in \mathbb{R}^{k_{\text{tr}} \times r_{\text{tot}}}$ is defined as

$$\begin{aligned} \mathbf{D}_{k,:} &:= \left((\mathbf{p}_k^{\ell_{\min}})^\top, \ldots, (\mathbf{p}_k^{\ell_{\max}})^\top\right) \\ &= \left(((\mathbf{V}^\top \mathbf{x}_k)^{\ell_{\min}})^\top, \ldots, ((\mathbf{V}^\top \mathbf{x}_k)^{\ell_{\max}})^\top\right), \end{aligned} \quad (6)$$

and $\mathbf{R} \in \mathbb{R}^{k_{\text{tr}} \times r}$ is the matrix of approximated time derivatives

$$\mathbf{R} := (\dot{\mathbf{x}}_1, \ldots, \dot{\mathbf{x}}_{k_{\text{tr}}})^\top \mathbf{V} \in \mathbb{R}^{k_{\text{tr}} \times r}. \quad (7)$$

If $\mathbf{D}$ has full column rank, then (5) has a unique solution which, moreover, converges to the intrusive ROM operators in (3) for $\Delta t \to 0$ and $k_{\text{tr}} \to \infty$ (c.f., [5]). In practice, however, full column rank can often not be achieved as $k_{\text{tr}} \geq r_{\text{tot}}$ snapshots are required—a number that can easily become prohibitively large. In many cases, linear dependencies among snapshots cause more than r_{tot} snapshots to be required for achieving full column rank of $\mathbf{D}$. To illustrate the polynomial scaling in r, Table I lists the matrix dimensions $r^{(\ell)}$ for different values of r and ℓ. One major advantage of projection-based reduced modeling is that the ROM (2) preserves the polynomial structure of the FOM (1), which in turn embeds structure arising from the underlying governing physical equations; however, a disadvantage is that the dimensions of the operators $\hat{\mathbf{A}}_\ell$ grow rapidly, even for moderate r. This in turn poses a challenge in achieving a well-posed OpInf formulation.

While targeted regularization stabilizes the learning problem (5) (c.f., [17], [11]), its effects can introduce bias into the OpInf solution. This is particularly true for the inferred operators corresponding to higher order polynomial terms. The methodology introduced in this paper exploits additional structure in the ROM form to circumvent issues caused by the polynomial scaling of the ROM operator dimensions. This is an enabler for achieving accurate, stable OpInf solutions for systems with high-order polynomial terms, extending our ability to learn projection-based ROMs for a larger class of nonlinear systems.

	$r^{(2)}$	$r^{(3)}$	$r^{(4)}$	$r^{(5)}$	$r^{(6)}$	$r^{(7)}$	$r^{(8)}$	$r^{(9)}$
$r=1$	1	1	1	1	1	1	1	1
$r=2$	3	4	5	6	7	8	9	10
$r=3$	6	10	15	21	28	36	45	55
$r=4$	10	20	35	56	84	120	165	220
$r=5$	15	35	70	126	210	330	495	715
$r=6$	21	56	126	252	462	792	1287	2002
$r=7$	28	84	210	462	924	1716	3003	5005
$r=8$	36	120	330	792	1716	3432	6435	11440

TABLE I: Number of columns $r^{(\ell)} := \binom{r+\ell-1}{\ell}$ for a reduced-order operator $\hat{\mathbf{A}}_\ell \in \mathbb{R}^{r \times r^{(\ell)}}$ with polynomial order ℓ acting in a reduced space of dimension r.

[1] For FOMs with nonpolynomial terms this structure might still be achieved by lifting the variables in a preprocessing step, c.f., [23]. Input terms can be considered by including additional summands to (1), see [5].

[2] A setting where only partial state observations are available could be treated in combination with the learning approach in [28]; this idea will be explored in future work.

III. Nested Operator Inference

In our nested OpInf approach, we restructure the OpInf learning problem (5) such that the dynamics governing the most important POD modes are learned first and reliably through targeted snapshot acquisition that exploits ROM structure, while the remaining entries of $\hat{\mathbf{O}}$ are learned conservatively within a regularized problem using the approximated time derivatives for the original snapshot data. The approach is motivated by the three main observations (O1), (O2), (O3) described below. The nested OpInf algorithm is described in Section IV.

(O1) Hierarchical basis structure: The POD basis matrix $\mathbf{V}$ is optimal in the sense that its snapshot projection error

$$\sum_{k=1}^{k_{\text{tr}}} \|\mathbf{x}_k - \mathbf{V}\mathbf{V}^\top \mathbf{x}_k\|_2^2 = \sum_{j=r+1}^{k_{\text{tr}}} \sigma_j^2 \tag{8}$$

is minimal amongst all linear subspaces of dimension r. Here, σ_j denote the singular values of the snapshot matrix $(\mathbf{x}_1, \ldots, \mathbf{x}_{k_{\text{tr}}}) \in \mathbb{R}^{n \times k_{\text{tr}}}$, in which column k contains snapshot $\mathbf{x}_k$. Projection-based model reduction is most effective when the singular values decay exponentially fast, meaning that r can be chosen to be small leading to a computationally efficient ROM. This fast singular value decay also implies that, for recovering a full-order state through

$$\mathbf{x}(t) \approx \mathbf{V}\hat{\mathbf{x}}(t) = \sum_{j=1}^{r} (\hat{\mathbf{x}}(t))_j \mathbf{v}_j \tag{9}$$

where $(\hat{\mathbf{x}}(t))_j$ denotes the j-th component of the reduced state $\hat{\mathbf{x}}$ at time t, it is most important for the ROM to capture the dynamics governing those first few entries of $\hat{\mathbf{x}}(t)$ associated to the dominant POD modes.

In our nested OpInf approach we respect the hierarchical structure of the basis by spending any computational budget available on reliably and stably learning those entries in $\hat{\mathbf{O}}$ that evolve the first POD modes, starting with the first POD mode, $\mathbf{v}_1$. To specifically target these entries of $\hat{\mathbf{O}}$, we identify learning problems within subspaces of *span*$(\mathbf{v}_1, \ldots, \mathbf{v}_r)$. We write these problems using the following notation: Given a subset of indices $\mathcal{I} = \{i_1 \leq \cdots \leq i_d\} \subset \{1, \ldots, r\}$, where we select $d < r$ indices, we define the basis matrix $\mathbf{V}_\mathcal{I} := (\mathbf{v}_{i_1}, \ldots, \mathbf{v}_{i_d}) \in \mathbb{R}^{n \times d}$, which has as columns the basis vectors $\mathbf{v}_{i_j}$ with $i_j \in \mathcal{I}$. Using trial space $\mathbf{V}_\mathcal{I}$ and test space $\mathbf{V}$ in a Petrov-Galerkin projection would result in the operators $\hat{\mathbf{A}}_{\ell,\mathcal{I}} \in \mathbb{R}^{r \times d^{(\ell)}}$ such that

$$\hat{\mathbf{A}}_{\ell,\mathcal{I}} \hat{\mathbf{x}}_\mathcal{I}^\ell = \mathbf{V}^\top \mathbf{A}_\ell (\underbrace{\mathbf{V}_\mathcal{I}\hat{\mathbf{x}}_\mathcal{I} \otimes \cdots \otimes \mathbf{V}_\mathcal{I}\hat{\mathbf{x}}_\mathcal{I}}_{\ell-\text{times}}) \quad \forall\ \hat{\mathbf{x}}_\mathcal{I} \in \mathbb{R}^d,$$

noting that each $\hat{\mathbf{A}}_{\ell,\mathcal{I}}$ is rectangular (with r rows and $d^{(\ell)}$ columns) and the corresponding reduced state $\hat{\mathbf{x}}_\mathcal{I}$ has dimension $d < r$.

Remark 1: Note that we do not intend to infer a Petrov-Galerkin ROM for the reduced-order dimension r; however, we will shortly see that these $\hat{\mathbf{A}}_{\ell,\mathcal{I}}$ are sub-matrices of the $\hat{\mathbf{A}}_\ell$ operators that we aim to infer.

We collect these Petrov-Galerkin sub-matrix operators in $\hat{\mathbf{O}}_\mathcal{I} := (\hat{\mathbf{A}}_{\ell_{\min},\mathcal{I}}, \ldots, \hat{\mathbf{A}}_{\ell_{\max},\mathcal{I}}) \in \mathbb{R}^{r \times d_{\text{tot}}}$, where $d_{\text{tot}} := \sum_{\ell \in \mathcal{L}} d^{(\ell)}$. Using the projections of the snapshots $\mathbf{x}_k$ onto $\mathbf{V}_\mathcal{I}$, we also define the data matrix $\mathbf{D}_\mathcal{I} \in \mathbb{R}^{k_{\text{tr}} \times d_{\text{tot}}}$, which has as its kth row

$$(\mathbf{D}_\mathcal{I})_{k,:} := \left(((\mathbf{V}_\mathcal{I}^\top \mathbf{x}_k)^{\ell_{\min}})^\top, \ldots, ((\mathbf{V}_\mathcal{I}^\top \mathbf{x}_k)^{\ell_{\max}})^\top \right). \tag{10}$$

(O2) Nested structure: A property that all projection-based reduced-order operators share by construction is that $(\hat{\mathbf{A}}_\ell)_{i,j}$ (the (i,j)th entry of $\hat{\mathbf{A}}_\ell \in \mathbb{R}^{r \times r^{(\ell)}}$) depends on at most ℓ trial basis vectors (according to the particular Kronecker product term represented in column j) and exactly one test basis vector specified by the row i. This means in particular that when a smaller subspace defined by the basis $\mathbf{V}_\mathcal{I}$, $\mathcal{I} \subset \{1, \ldots, r\}$ is expanded to $\mathbf{V}$ by including the other basis vectors $\{1, \ldots, r\} \notin \mathcal{I}$, then the larger reduced-order operator matrix $\hat{\mathbf{A}}_\ell \in \mathbb{R}^{r \times r^{(\ell)}}$ includes $\hat{\mathbf{A}}_{\ell,\mathcal{I}}$ as a submatrix.

For our nested OpInf approach, observation (O2) implies that we can partition our inference into multiple steps. First, we can learn the entries of $\hat{\mathbf{A}}_{\ell,\mathcal{I}}$ with $\mathcal{I} = \{1, \ldots, d\}$ comprising the indices for the most important modes, through a OpInf learning problem of the form

$$\min_{\hat{\mathbf{O}}_\mathcal{I}} \|\mathbf{D}_\mathcal{I}\hat{\mathbf{O}}_\mathcal{I}^T - \mathbf{R}_\mathcal{I}\|_F^2. \tag{11}$$

Next, we can learn the remaining entries of $\hat{\mathbf{O}}$, enforcing the entries already learned as submatrices within each $\hat{\mathbf{A}}_\ell$. (Note that we could also consider a sequence of submatrix problems building up to the entire reduced dimension r.) Before we specify how $\mathbf{R}_\mathcal{I} \in \mathbb{R}^{k_{\text{tr}} \times r}$ should be chosen, we discuss the stability of the learning problem (11).

(O3) Stability improvement for smaller subspaces: Since $\mathbf{V}_\mathcal{I}^\top \mathbf{x}_k$ is a submatrix of $\mathbf{V}^\top \mathbf{x}_k$, the data matrix $\mathbf{D}_\mathcal{I}$ is a submatrix of $\mathbf{D}$ by construction (compare (10) to (6)). Consequently, there exists $\mathbf{P}_\mathcal{I} \in \{0,1\}^{r_{\text{tot}} \times d_{\text{tot}}}$ such that $\mathbf{D}_\mathcal{I} = \mathbf{D}\mathbf{P}_\mathcal{I}$. Using this relationship, we bound the minimum singular value $\sigma_{\min}(\mathbf{D}_\mathcal{I})$ of $\mathbf{D}_\mathcal{I}$ through

$$\begin{aligned}
\sigma_{\min}(\mathbf{D}_\mathcal{I})^2 &= \min_{\mathbf{y} \in \mathbb{R}^{d_{\text{tot}}}} \frac{\mathbf{y}^\top \mathbf{D}_\mathcal{I}^\top \mathbf{D}_\mathcal{I} \mathbf{y}}{\|\mathbf{y}\|_2^2} \\
&= \min_{\mathbf{y} \in \mathbb{R}^{d_{\text{tot}}}} \frac{\mathbf{y}^\top \mathbf{P}_\mathcal{I}^\top \mathbf{D}^\top \mathbf{D} \mathbf{P}_\mathcal{I} \mathbf{y}}{\|\mathbf{y}\|_2^2} \\
&\geq \min_{\mathbf{y} \in \mathbb{R}^{r_{\text{tot}}}} \frac{\mathbf{y}^\top \mathbf{D}^\top \mathbf{D} \mathbf{y}}{\|\mathbf{y}\|_2^2} \\
&= \sigma_{\min}(\mathbf{D})^2.
\end{aligned} \tag{12}$$

Analogously, the largest singular value $\sigma_{\max}(\mathbf{D}_\mathcal{I})$ of $\mathbf{D}_\mathcal{I}$ is no larger than $\sigma_{\max}(\mathbf{D})$, that is $\sigma_{\max}(\mathbf{D}_\mathcal{I}) \leq \sigma_{\max}(\mathbf{D})$. Together, these bounds imply that the least squares learning problem (16) is at least as stable as (11). Further, as the dimension of $\mathcal{I}$ decreases, the conditioning of (11) will improve.

For our nested OpInf approach, we conclude from observation (O3) that, for stability purposes, it is most beneficial to learn the entries of $\hat{\mathbf{O}}$ in the smallest possible subproblems of the form (11). In combination with the hierarchical structure of the POD basis (observation (O1)), we should thus first

set $d = 1$. We set $\mathcal{I} = \{1\}$ to learn $\hat{\mathbf{O}}_{\{1\}}$, and we set $\mathcal{I} = \{2\}$ to learn $\hat{\mathbf{O}}_{\{2\}}$. Next, we set $d = 2$, we enforce $\hat{\mathbf{O}}_{\{1\}}$ and $\hat{\mathbf{O}}_{\{2\}}$ to be submatrices of $\hat{\mathbf{O}}_{\{1,2\}}$, and we learn the remaining entries of $\hat{\mathbf{O}}_{\{1,2\}}$. By following this structure, each learning problem (11) is as small as possible; in fact, after first learning the entries of $\hat{\mathbf{O}}_{\mathcal{I}}$ associated to all $\tilde{\mathcal{I}} \subsetneq \mathcal{I}$, only $\sum_{\ell \in \ell, \ell \geq d} \binom{\ell}{d}$ entries of $\hat{\mathbf{O}}_{\mathcal{I}}$ are left to be learned in each column. As this scaling is independent of the reduced dimension r of the final ROM (2), it is a strong improvement over the original OpInf learning problem (5) where all r_{tot} column entries are learned together (see Table I).

The main complication with learning entries of $\hat{\mathbf{O}}$ first through small subspaces $\mathbf{V}_{\mathcal{I}}$ before enforcing them in larger subspace problems or in $\hat{\mathbf{O}}$ itself, is that any errors introduced in $\hat{\mathbf{O}}_{\mathcal{I}}$ will be propagated into the subsequent learning problems. It is therefore of utmost importance that $\hat{\mathbf{O}}_{\mathcal{I}}$ are learned reliably in (11) with as little error as possible. To guarantee this, we follow the re-projection approach of [6]. Specifically, we define the time-derivative matrix $\mathbf{R}_{\mathcal{I}} \in \mathbb{R}^{k_{\text{tr}} \times r}$ in (11) as

$$\mathbf{R}_{\mathcal{I}} := \left(f(\mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^{\top}\mathbf{x}_1), \ldots, f(\mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^{\top}\mathbf{x}_{k_{\text{tr}}})\right)^{\top} \mathbf{V}. \tag{13}$$

If $\mathbf{D}_{\mathcal{I}}$ has full column rank, then $\hat{\mathbf{O}}_{\mathcal{I}}$ is indeed the unique solution to (11) (see [6], Corollary 3.2), and we avoid introducing an error. The downside of the choice (13) is that it requires simulating the FOM to evaluate f at the requisite states. In many applications it is possible to compute $f(\mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^{T}\mathbf{x}_k)$ for a given $\mathcal{I}$, either because the full right-hand side f of (1) can be evaluated, or because it can be approximated sufficiently well[3] by computing a single time step of the FOM with initial condition $\mathbf{x}(0) = \mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^{\top}\mathbf{x}_k$ (c.f., [6], [7]). Either option increases the computational cost of training, due to the additional evaluations of the FOM. This increased computational cost can be mitigated by introducing to the nested OpInf approach a constraint on the number of extra evaluations of f (or its approximation) beyond the initial snapshot set. Under a constrained training budget, these evaluations can be prioritized to learn the dynamics governing the most important POD modes first.

An advantage of using the re-projection approach to define the time-derivative matrix $\mathbf{R}_{\mathcal{I}}$ in (13) is that we are guaranteed to recover the intrusive reduced-operators $\hat{\mathbf{A}}_{\ell}$, $\ell \in \mathcal{L}$. Through their definition in (3) as a Galerkin projection, the intrusive operators inherit some stability properties from the full-order operators. For example, if the FOM includes a negative definite linear operator $\mathbf{A}_1$, then its intrusive ROM counterpart $\hat{\mathbf{A}}_1 = \mathbf{V}^{\top}\mathbf{A}_1\mathbf{V}$ is also negative definite. Our nested approach hence not only improves the numerical stability of each learning problem, but also may improve the stability properties of the learned ROM itself.

IV. Algorithm

Our proposed nested OpInf algorithm is outlined in Algorithm 1, with a call `SubmatrixLearning` to Algorithm 2 for the learning problems within the nested subspaces. The algorithms are divided into sections through comments ("\\"), with each section explained individually below, starting with those in Algorithm 1. A reference implementation is provided at https://github.com/nicolearetz/Nested-OpInf.

Algorithm 1: Nested Operator Inference

Input: training snapshots $\mathcal{S}_{\text{tr}} = \{\mathbf{x}_k\}_{k=1}^{k_{\text{tr}}}$, POD space $\mathbf{V} \in \mathbb{R}^{n \times r}$, reduced dimension r, budget $K_+ \in \mathbb{N}$, submatrix buffer $\beta_0 \in \mathbb{N}$

```
\\Main variables
```
Construct the data matrix $\mathbf{D} \in \mathbb{R}^{k_{\text{tr}} \times r_{\text{tot}}}$ for the training snapshots in $\mathcal{S}_{\text{tr}}$ using (6) where $r_{\text{tot}} = \sum_{\ell \in \mathcal{L}} \binom{r+\ell-1}{\ell}$ for the polynomial orders $\mathcal{L}$ of the FOM (1)
Build $\mathbf{R} \in \mathbb{R}^{k_{\text{tr}} \times r}$ from (7) using a finite difference approximation of $\dot{\mathbf{x}}_k$ for each $\mathbf{x}_k \in \mathcal{S}_{\text{tr}}$
Initialize $\mathcal{S}_+ = \emptyset$
Initialize $\mathbf{T} = \mathbf{0} \in \mathbb{R}^{r \times r_{\text{tot}}}$, $\mathbf{b} = (1, \ldots, 1)^{\top} \in \{0,1\}^{r_{\text{tot}}}$

```
\\Nested structure
```
for $i = 1, \ldots, r$ **do**
  for $d = 1, \ldots, i$ **do**
    for $\mathcal{I} \subset \{1, \ldots, i\}$, $i \in \mathcal{I}$, $|\mathcal{I}| = d$ **do**
      $\mathbf{T}$, $\mathbf{b}$, $\mathcal{S}_{++}$, $\mathbf{R}_+$, K_+ = `SubmatrixLearning`$(\mathcal{S}_{\text{tr}}, K_+, \mathbf{T}, \mathcal{I}, \mathbf{V}, \mathbf{b}, \beta_0)$
      Expand $\mathcal{S}_+ \leftarrow \mathcal{S}_+ \cup \mathcal{S}_{++}$
      Expand $\mathbf{R} \leftarrow (\mathbf{R}^{\top}, \mathbf{R}_+^{\top})^{\top}$
      if $K_+ = 0$ **then**
        break all for loops
      if $\mathcal{S}_{++} = \emptyset$ **then**
        break the two inner for loops

```
\\Final learning problem on V
```
Construct $\mathbf{D}_+ \in \mathbb{R}^{|\mathcal{S}_+| \times r_{\text{tot}}}$ with (6) using the states contained in $\mathcal{S}_+$ in place of the snapshots in $\mathcal{S}_{\text{tr}}$
Expand $\mathbf{D} \leftarrow \left(\mathbf{D}^{\top}, \mathbf{D}_+^{\top}\right)^{\top}$
Compute weight matrix $\mathbf{W}$ with (15)
Solve learning problem (14) for $\hat{\mathbf{O}}$

return $\hat{\mathbf{O}}$

Inputs: The inputs to Algorithm 1 include a set of training snapshots, denoted $\mathcal{S}_{\text{tr}}$, and a reduced space $\mathbf{V} \in \mathbb{R}^{n \times r}$ with r orthonormal basis functions in each column obtained through POD. The input budget variable K_+ counts the maximum number of available evaluations of the full-order time derivative function f, either in the form of a direct call, or through re-projection of the FOM (1) (c.f., [6]). The budget variable K_+ is counted down throughout the algorithm. The final input β_0 is a buffer variable ensuring that all least-squares subspace learning problems are over-

[3]Note that for $|\mathcal{I}|$ small enough, $\mathbf{D}_{\mathcal{I}}$ is well conditioned such that an $\mathcal{O}(\Delta t)$ approximation error can be tolerated, as discussed in observation (O3).

determined as an additional means to provide stability.

Main variables: We initialize the data matrix $\mathbf{D}$ from (6) of the standard OpInf learning problem for the snapshots $\mathcal{S}_{\text{tr}}$. Throughout the algorithm, $\mathbf{D}$ is first used to obtain the submatrices $\mathbf{D}_{\mathcal{I}}$ for each encountered subset $\mathcal{I} \subset \{1, \dots, r\}$, and in a final OpInf learning problem for $\hat{\mathbf{O}}$ blending all available data. We initialize the time derivative matrix $\mathbf{R}$ from (7) using a finite difference approximation of $\dot{\mathbf{x}}_k$ for each training snapshot $\mathbf{x}_k \in \mathcal{S}_{\text{tr}}$. The set $\mathcal{S}_+$ is initialized as empty; throughout the nested learning loop, it collects states for which the exact time derivative is computed. The matrix $\mathbf{T} \in \mathbb{R}^{r \times r_{\text{tot}}}$ is the current approximation to the intrusive $\hat{\mathbf{O}}$; its rows are populated with the learned entries of $\hat{\mathbf{O}}$ during the nested loop. The Boolean vector $\mathbf{b} \in \{0,1\}^{r_{\text{tot}}}$ keeps track which rows $\mathbf{T}_{j,:}$ have already been learned ($\mathbf{b}_j = 0$), and which are still unknown ($\mathbf{b}_j = 1$).

Nested structure: The nested loops prioritize the first modes (outer loop over mode index i) to respect the hierarchical basis structure (O1). The nested structure (O2) is respected by increasing the subspace dimension (middle loop over dimension d) only once operators have been learned for all possible subspaces of this dimension (inner loop over index subsets $\mathcal{I}$, $|\mathcal{I}| = d$). This looping order guarantees that we are always learning as few entries together as possible to obtain best learning stability (O3), starting with the interactions of the most dominant modes. The operator entries themselves are learned within the call `SubmatrixLearning` to Algorithm 2 where the budget K_+, the operator matrix $\mathbf{T}$ and the row index tracker $\mathbf{b}$ are updated. Moreover, the call returns a set $\mathcal{S}_{++}$ of states $\mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^\top \mathbf{x}_k$ for which the exact projected time derivative $\mathbf{V}^T f(\mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^\top \mathbf{x}_k)$ has been computed, and a matrix $\mathbf{R}_+$ in which these derivatives are stored. Both $\mathcal{S}_{++}$ and $\mathbf{R}_+$ are used to extend $\mathcal{S}_+$ and $\mathbf{R}$, respectively, for later use in a final OpInf learning problem employing all available information. The latter is reached when all nested loops have terminated or if the budget of calls to f has been depleted ($K_+ = 0$), While budget is still available, i.e. $K_+ > 0$, we break the nested structure only when no new information was learned ($\mathcal{S}_{++} = \emptyset$). This case arises when the data matrix $\mathbf{D}_{\mathcal{I}}$ within the call `SubmatrixLearning` to Algorithm 2 is ill-posed. If so, we proceed with the next iteration of the outermost loop, hence resetting the subspace dimension back to $d = 1$. This reset brings the next encountered data matrix back down to width $|\mathcal{L}|$, for which it is most likely to be again well-posed.

Final learning problem on $\mathbf{V}$: Once no more calls to the FOM are available, i.e., $K_+ = 0$, we return to the full reduced space $\mathbf{V}$ for an OpInf learning problem for all entries of $\hat{\mathbf{O}}$. This final learning problem incorporates all available information: The original snapshots in $\mathcal{S}_{\text{tr}}$ and their approximated time derivatives, and the states in $\mathcal{S}_+$ for which the action of f was computed within Algorithm 1. To this end, we first expand the data matrix $\mathbf{D}$ with the data matrix $\mathbf{D}_+$ that is obtained from (6) when using the states in $\mathcal{S}_+$ as snapshots. We then solve the minimization problem

$$\min_{\hat{\mathbf{O}}} \|\mathbf{W}^{\frac{1}{2}}(\mathbf{D}\hat{\mathbf{O}} - \mathbf{R})\|_F^2 + \lambda \|\hat{\mathbf{O}} - \mathbf{T}\|_F^2. \tag{14}$$

The tracker variable $\mathbf{T}$ is used as regularization to encourage nested structure. Since $\mathbf{T}$ was initialized with zeros, any entries that were not inferred during the nested part of the algorithm are regularized with standard Tikhonov regularization. The strength $\lambda \geq 0$ of the regularization can be optimized in an outer loop if necessary (see [17]).[4] The least squares problem (14) is weighted with

$$\mathbf{W} := \operatorname{diag}(\underbrace{\Delta t^2, \dots, \Delta t^2}_{k_{\text{tr}} \text{ times}}, \underbrace{1, \dots, 1}_{|\mathcal{S}_+| \text{ times}}). \tag{15}$$

This weighting is necessary because the first k_{tr} entries of $\mathbf{R}$ for the original snapshots include additional projection and time derivative approximation error, while the action of f was explicitly computed for all states in $\mathcal{S}_+$.

We continue with the call `SubmatrixLearning` to Algorithm 2 which infers those entries of $\hat{\mathbf{O}}_{\mathcal{I}}$ that have not been learned in previous interations within the nested structure of Algorithm 1, and stores them in $\mathbf{T}$.

Submatrix setup: We first restrict the data matrix $\mathbf{D}$, the operator matrix $\mathbf{T}$, and the position vector $\mathbf{b}$ to only those entries that are associated to the trial subspace $\mathbf{V}_{\mathcal{I}}$. The exact implementation of this step depends on the order chosen for the exponential operation $\hat{\mathbf{x}}^\ell$, but can generally be expressed through a matrix multiplication. The matrix $\mathbf{B} = \mathbf{D}_{\mathcal{I}}\mathbf{T}_{\mathcal{I}}$ contains the action of all operator entries learned thus far on the projected snapshots. Since we are only interested in learning those entries of $\hat{\mathbf{O}}_{\mathcal{I}}$ that have not yet been learned, i.e., those for which $\mathbf{b}_{\mathcal{I}}$ is still equal to one, we construct the matrix $\mathbf{I}_{\mathbf{b}}$ such that the multiplication $\mathbf{D}_{\mathcal{I}}\mathbf{I}_{\mathbf{b}}$ removes all other columns of $\mathbf{D}_{\mathcal{I}}$.

Stability check: Before computing the entries of the time derivative matrix $\mathbf{R}_{\mathcal{I}}$ using expensive calls to the full-order f, it is important to note that $\mathbf{D}_{\mathcal{I}}\mathbf{I}_{\mathbf{b}}$ can have redundant rows or be ill-conditioned. To not needlessly deplete the budget K_+, we aim to achieve a minimal stability with a condition number above Δt^{-2} without using more than β_0 calls to f than strictly necessary given the number $\|\mathbf{b}_{\mathcal{I}}\|_1$ of entries to be inferred. To this end, we perform a QR decomposition $\mathbf{Q}\mathbf{L}^\top = \mathbf{I}_{\mathbf{b}}^\top \mathbf{D}_{\mathcal{I}}^\top \mathbf{P}$ with pivoting: The first β columns of the permutation matrix $\mathbf{P}$ identify $\beta = \min\{\|\mathbf{b}_{\mathcal{I}}\|_1 + \beta_0, K_+\}$ linearly independent rows of $\mathbf{D}_{\mathcal{I}}$ (see [30] for details). We save these columns in the matrix $\mathbf{I}_\beta \in \{0,1\}^{k_{\text{tr}} \times \beta}$, and let $k_1, \dots, k_\beta$ be the identified row indices. The multiplication $\mathbf{I}_\beta^\top \mathbf{D}_{\mathcal{I}}\mathbf{I}_{\mathbf{b}}$ then eliminates all rows of $\mathbf{D}_{\mathcal{I}}\mathbf{I}_{\mathbf{b}}$ but rows $k_1, \dots, k_\beta$. If the condensed matrix $\mathbf{I}_\beta^\top \mathbf{D}_{\mathcal{I}}\mathbf{I}_{\mathbf{b}}$ satisfies the stability condition $\kappa_2(\mathbf{I}_{\mathbf{b}}^\top \mathbf{D}_{\mathcal{I}}^\top \mathbf{I}_\beta \mathbf{I}_\beta^\top \mathbf{D}_{\mathcal{I}}\mathbf{I}_{\mathbf{b}}) > \Delta t^{-2}$, we continue learning the remaining entries in $\mathbf{T}_{\mathcal{I}}$; otherwise, we return to the outer algorithm 1 without any changes or waste of computational budget.

[4] The regularization term can be expanded to target different parts of each reduced operator in $\hat{\mathbf{O}}$ separately, in particular to enforce learned entries in $\mathbf{T}$ or to drive higher-order polynomial operators more strongly towards zero (see [17], [11]).

Algorithm 2: Submatrix learning

Input: training snapshots $\mathcal{S}_{\text{tr}} = \{\mathbf{x}_k\}_{k=1}^{k_{\text{tr}}}$, training data matrix $\mathbf{D}$, budget K_+, operator estimate $\mathbf{T}$, learned position tracker $\mathbf{b}$, index set $\mathcal{I}$, reduced basis $\mathbf{V}$, submatrix buffer β_0

\\Submatrix setup
Construct the projection matrix $\mathbf{P}_{\mathcal{I}} \in \{0,1\}^{r_{\text{tot}} \times |\mathcal{I}|_{\text{tot}}}$ from (O3)
Compute $\mathbf{D}_{\mathcal{I}} = \mathbf{D}\mathbf{P}_{\mathcal{I}}$, $\mathbf{T}_{\mathcal{I}} = \mathbf{P}_{\mathcal{I}}^\top \mathbf{T}$, $\mathbf{b}_{\mathcal{I}} = \mathbf{P}_{\mathcal{I}}^\top \mathbf{b}$
Compute action $\mathbf{B} = \mathbf{D}_{\mathcal{I}}\mathbf{T}_{\mathcal{I}}$ of the operator entries that have already been learned
Set $\mathbf{I}_{\mathbf{b}} = \text{diag}(\mathbf{b}_{\mathcal{I}})$, and remove all zero-columns

\\Stability check
Set $\beta = \min\{\|\mathbf{b}_{\mathcal{I}}\|_1 + \beta_0, K_+\}$
Compute QR decomposition $\mathbf{Q}\mathbf{L}^\top = \mathbf{I}_{\mathbf{b}}^\top \mathbf{D}_{\mathcal{I}}^\top \mathbf{P}$ with pivoting and let $\mathbf{I}_\beta \in \{0,1\}^{k_{\text{tr}} \times \beta}$ be the first β columns of the permutation matrix $\mathbf{P} \in \{0,1\}^{k_{\text{tr}} \times k_{\text{tr}}}$

if $\kappa_2(\mathbf{I}_{\mathbf{b}}^\top \mathbf{D}_{\mathcal{I}}^\top \mathbf{I}_\beta \mathbf{I}_\beta^\top \mathbf{D}_{\mathcal{I}} \mathbf{I}_{\mathbf{b}}) > \Delta t^{-2}$ **then**
  return $\mathbf{T}$, $\mathbf{b}$, $\emptyset$, $\mathbf{0} \in \mathbb{R}^{0 \times r}$, K_+

\\Time derivative information
Initialize $\mathbf{R}_{\mathcal{I}} = \mathbf{0} \in \mathbb{R}^{\beta \times r}$, $\mathcal{S}_{++} = \emptyset$
Let $k_1, \ldots, k_\beta$ be the unique indices of the non-zero rows in $\mathbf{I}_\beta$
for $i = 1, \ldots, \beta$ **do**
  Compute $f(\mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^\top \mathbf{x}_{k_i})$
  Set the ith row of $\mathbf{R}_{\mathcal{I}}$ to be $f(\mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^\top \mathbf{x}_{k_i})^\top \mathbf{V}$
  Store new state: $\mathcal{S}_{++} \leftarrow \mathcal{S}_{++} \cup \{\mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^\top \mathbf{x}_{k_i}\}$
  Adjust remaining budget: $K_+ \leftarrow K_+ - 1$

\\Subspace learning problem
Solve minimization problem (16) for $\mathbf{T}_+$
Populate $\mathbf{T}$ with the values in $\mathbf{T}_+$ such that $\mathbf{I}_{\mathbf{b}}^\top \mathbf{P}_{\mathcal{I}}^\top \mathbf{T} = \mathbf{T}_+$ and all other entries remain the same
Set the values for each learned position in $\mathbf{b}$ to 0 such that $\mathbf{I}_{\mathbf{b}}^\top \mathbf{P}_{\mathcal{I}}^\top \mathbf{b} = \mathbf{0}$ and all other entries remain the same

return $\mathbf{T}$, $\mathbf{b}$, $\mathcal{S}_+$, $\mathbf{R}_{\mathcal{I}}$, K_+

Time derivative information: For each of the identified row indices k_i, $i = 1, \ldots, \beta$ we compute the projected time derivative $f(\mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^\top \mathbf{x}_{k_i})^\top \mathbf{V}$ and store it as a row of the time derivative matrix $\mathbf{R}_{\mathcal{I}}$. These computations can be performed either through a direct call to f, or through the re-projection method (c.f., [6]). In either case, the remaining budget of FOM calls is reduced for each evaluation, and the set $\mathcal{S}_{++}$ is expanded to include the additional state $\mathbf{V}_{\mathcal{I}}\mathbf{V}_{\mathcal{I}}^\top \mathbf{x}_{k_i}$ for use in the final learning problem of the outer Algorithm 1.

Subspace learning problem: Finally, we learn the missing reduced-order operator entries in the least squares problem

$$\min_{\mathbf{T}_+ \in \mathbb{R}^{r \times \|\mathbf{b}_{\mathcal{I}}\|_1}} \|\mathbf{I}_\beta^\top \mathbf{D}_{\mathcal{I}} \mathbf{I}_{\mathbf{b}} \mathbf{T}_+^\top - (\mathbf{R}_{\mathcal{I}} - \mathbf{I}_\beta^\top \mathbf{B})\|_F^2. \tag{16}$$

The inclusion of $\mathbf{I}_{\mathbf{b}}$ in the first term in (16) restricts $\mathbf{D}_{\mathcal{I}}$ to those columns for which operator entries still need to be learned, while the second term is adjusted to account for those parts of the operator that have already been learned in previous iterations. The multiplication with $\mathbf{I}_\beta^\top$ restricts the problem to using only β equations instead of k_{tr}. Since $\mathbf{I}_\beta^\top \mathbf{D}_{\mathcal{I}} \mathbf{I}_{\mathbf{b}}$ has passed the stability check, and because $\mathbf{R}_{\mathcal{I}}$ was computed using the full-order operator f, the learned entries $\mathbf{T}_+$ are guaranteed to recover the corresponding entries of $\hat{\mathbf{O}}$ with high precision. Once these entries $\mathbf{T}_+$ are computed, we store them in $\mathbf{T}$, and update the position vector $\mathbf{b}$ to mark which entries are computed. We return the updated operator matrix $\mathbf{T}$, the updated position vector $\mathbf{b}$, and the remaining budget K_+ of calls to f, as well as the time derivative matrix $\mathbf{R}_{\mathcal{I}}$ and the set $\mathcal{S}_{++}$ of the states of which $\mathbf{R}_{\mathcal{I}}$ contains the time derivative information.

The key advantage of the proposed approach is that it transforms the OpInf learning problem from a single regression problem with a large number of unknowns into multiple smaller regression problems that exploit the nested structure of the reduced operators that are being inferred. This has the advantage of improving numerical conditioning, which is particularly important for reduction of nonlinear systems with terms of high polynomial order.

V. Numerical Examples

To demonstrate the nested OpInf method, we construct a ROM for the shallow ice equations

$$\frac{\partial x}{\partial t} = \frac{\rho g}{\alpha} x^2 \partial_\xi x + \frac{2\gamma \rho^3 g^3}{5} x^5 |\partial_\xi x|^2 \partial_\xi x =: f(x) \tag{17}$$

for $\xi \in [0, 1000]$, and initial condition

$$x(\xi, 0) = 10^{-2} + 630\left(\frac{\xi}{2000} + 0.25\right)^4 \left(\frac{\xi}{2000} - 0.75\right)^4, \tag{18}$$

constant rate factor $\gamma = 10^{-4}[s^{-1}Pa^{-3}]$, ice density $\rho = 910[kg/m^3]$, gravitational acceleration $g = 9.81[m/s^2]$, and homogeneous Neumann boundary conditions. The state $x(\xi, t)$ describes the variation in space ξ and time t of the thickness of an ice sheet on even ground under no slip conditions ($\alpha = 1e-15$) (see [31], section 5.6.1). We discretize (17) in `pyapprox` [32] with linear finite elements of dimension $n = 512$. This leads to a FOM of the form (1), which we solve using implicit Euler time stepping with $\Delta t = 10^{-3}$. The runtime from $t = 0$ to $t = 10$ is $103s$.

The model (17) has two polynomial terms: one of order $\ell = 3$ and one of order $\ell = 8$, i.e., $\mathcal{L} = \{3, 8\}$. We construct a ROM of the form (2), whose reduced-order operators $\hat{\mathbf{A}}_3 \in \mathbb{R}^{r \times r^{(3)}}$ and $\hat{\mathbf{A}}_8 \in \mathbb{R}^{r \times r^{(8)}}$ act on

$$\begin{aligned}\hat{\mathbf{x}}^{(3)} &= \left(x_1^3, x_1^2 x_2, x_1 x_2^2, x_2^3, \ldots\right)^\top \in \mathbb{R}^{r^{(3)}}, \\ \hat{\mathbf{x}}^{(8)} &= \big(x_1^8, x_1^7 x_2, x_1^6 x_2^2, x_1^5 x_2^3, \\ &\qquad x_1^4 x_2^4, x_1^3 x_2^5, x_1^2 x_2^6, x_1 x_2^7, x_2^8, \ldots\big)^\top \in \mathbb{R}^{r^{(3)}},\end{aligned} \tag{19}$$

where the redundant terms (e.g., $x_2 x_1^2$, which duplicates $x_1^2 x_2$) have been removed (see Table I for reference). For

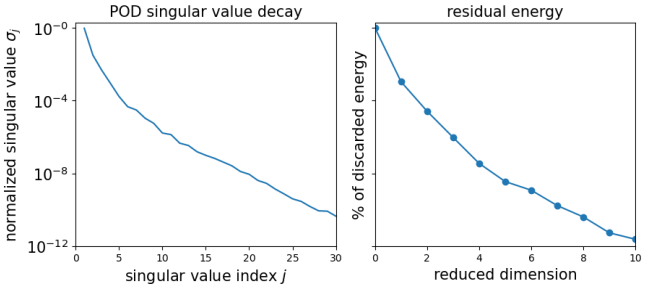

Fig. 1: Left: POD singular value decay after snapshot rescaling, normalized by the sum of all singular values σ_j. Right: Remaining energy $\sum_{j=1}^{r}\sigma_j^2 / \sum_{j=1}^{k_{\text{tr}}}\sigma_j^2$.

training, we run the FOM from $t = 0$ until $t = 2$ for a total of $k_{\text{tr}} = 2001$ snapshots. We apply POD to construct the reduced basis $\mathbf{V}$. Figure 1 (left) shows the singular value decay of the training snapshots. Based on the residual energy (Figure 1, right), it can be seen that with a POD basis of dimension $r = 4$ the snapshots can be reconstructed with relative error less than 10^{-7}. Increasing the reduced dimension to $r = 7$ reduces the relative reconstruction error further to approximately 10^{-10}.

Based on these singular values, we conduct a numerical study of ROMs with dimensions $r = 4, 5, 6$ and 7. Consequently, the OpInf data matrix $\mathbf{D}$ has $k_{\text{tr}} = 2001$ rows and $r_{\text{tot}} = 185$ ($r = 4$), $r_{\text{tot}} = 530$ ($r = 5$), $r_{\text{tot}} = 1413$ ($r = 6$), or $r_{\text{tot}} = 3087$ ($r = 7$) columns. In each case, the OpInf learning problem (5) is vastly underdetermined with $\text{rank}(\mathbf{D})$ between 20 and 25.

To obtain a comparative reference using the standard OpInf approach, we compute $\mathbf{R}_{\mathcal{I}}$ using $K_+ := k_{\text{tr}}$ calls to f. Note that since $\mathbf{D}$ is rank deficient, there does not exist a unique solution to (5) and Corollary 3.2 in [6] does not apply. Instead, we apply Tikhonov regularization as described in [17], [11], with regularization parameters chosen in a grid search to minimize the reconstruction misfit of the snapshots. Figure 2 shows that the best standard OpInf ROM is obtained for $r = 4$, where only $r_{\text{tot}} = 185$ entries need to be learned in each row of $\hat{\mathbf{O}}$. For this case, enough snapshot data are available to learn a reasonable ROM, although the quality of the ROM prediction beyond the training interval degrades quickly. Increasing the reduced dimension r introduces additional richness into the reduced approximation. Figure 2 shows that this reduces the error over the training interval, but it leads to ROMs that exhibit blowup as we attempt to predict further in time. While these results for the standard OpInf approach could potentially be improved through better tuning of the regularization parameters, they illustrate the challenges of inferring operators for nonlinear systems with high-degree polynomial terms.

For our nested OpInf approach we proceed as described in Section IV with a computational budget of $K_+ = 2001$. This budget is exhausted within the loop for $i = 6$, at which point we solve the final, extended OpInf problem (14) where the time derivative matrix $\mathbf{R}$ is estimated from the

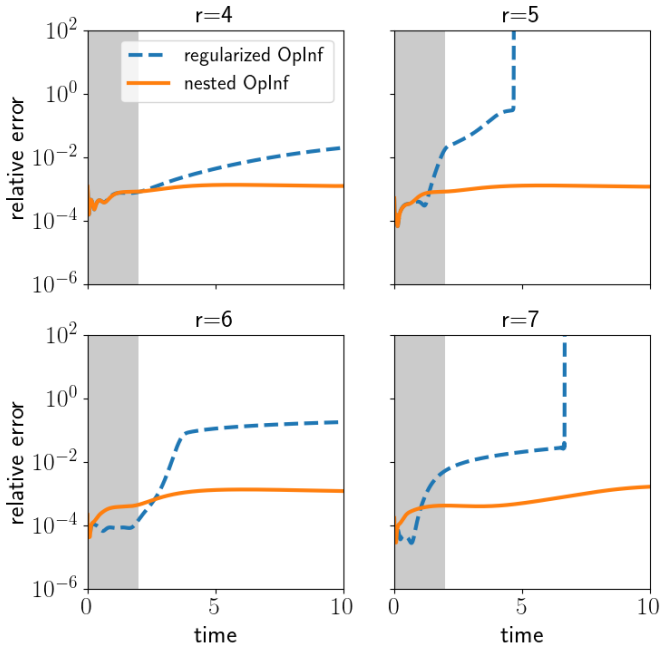

Fig. 2: Relative approximation error $\|\mathbf{x}(t) - \mathbf{V}\hat{\mathbf{x}}(t)\|_2 / \|\mathbf{x}(t)\|_2$ for $t \in [0, 10]$ of the nested and regularized OpInf ROMs for different reduced dimensions r. The grey area marks the training interval.

original snapshots using central finite differences and thereby polluted with $\mathcal{O}(\Delta t)$ error.

Remark 2: Since the FOM uses implicit Euler time stepping, the backward finite difference method would recover $f(\mathbf{x}_k) \approx f(\mathbf{V}\mathbf{V}^\top \mathbf{x}_k)$ for each snapshot. While advantageous in practice, here we explicitly refrain from this trick to provide a fair comparison with the regularized OpInf approach where the budget K_+ was spent to compute $f(\mathbf{V}\mathbf{V}^\top \mathbf{x}_k)$.

Figure 2 shows that the ROMs learned with the nested OpInf approach overcome these limitations. Within the training interval, the nested OpInf ROMs exhibit a similar reconstruction error to the standard OpInf ROMs. Outside the training interval, the nested OpInf ROMs remain stable and this error level is maintained. Notable in Figure 2 is that the nested OpInf ROMs incur slightly larger training reconstruction errors but achieve vastly improved prediction performance. This indicates that the nested OpInf formulation helps to avoid overfitting. The magnitude of the nested OpInf prediction error could be improved further with a larger budget K_+ available for computing f during training.

In these results, the ROMs were solved using the Runge-Kutta midpoint method. A ROM solve from $t = 0$ until $t = 10$ takes up to $4.6s$, leading to computational speed-up factors of 22 and greater.

VI. Conclusion

This paper introduces the nested OpInf method for learning physics-based ROMs from snapshot data while exploiting the nested structure of projection-based ROMs and the hierarchy of basis functions spanning the reduced space.

Given a budget of calls to the full-order solver, nested OpInf first learns the dynamics of the most important modes, and fills the remaining entries conservatively using approximated data once the budget is depleted. As a result, the nested OpInf approach solves multiple smaller regression problems that are numerically better conditioned that the single large regression problem in the standard OpInf method. Numerical results for the shallow ice equations illustrate the importance of these advancements, particularly for nonlinear systems with high-order polynomial operators where an all-at-once approach to OpInf leads to unstable ROMs due to extreme ill-conditioning.

In this paper we have considered the case of having full state snapshots that arise from a high-fidelity simulation; however, the approach could be extended to the case of noisy and/or partial snapshot data. A possible direction of future work would be to combine the Bayesian OpInf approach [33] or the goal-oriented learning approach [28] with the exploitation of nestedness as proposed here. This would, for example, permit the learning of ROMs from experimental data.

Code availability: The code and both training and test data used to generate the results in this paper are available at https://github.com/nicolearetz/Nested-OpInf.

REFERENCES

[1] P. Benner, S. Gugercin, and K. Willcox, "A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM review*, vol. 57, pp. 483–531, 2015.

[2] P. Benner, M. Ohlberger, A. Cohen, and K. Willcox, *Model reduction and approximation: theory and algorithms*. SIAM, 2017.

[3] C. Gräßle, M. Hinze, and S. Volkwein, "Model order reduction by proper orthogonal decomposition," 2020.

[4] J. S. Hesthaven, G. Rozza, and B. Stamm, *Certified reduced basis methods for parametrized partial differential equations*, vol. 590. Springer, 2016.

[5] B. Peherstorfer and K. Willcox, "Data-driven operator inference for nonintrusive projection-based model reduction," *Computer Methods in Applied Mechanics and Engineering*, vol. 306, pp. 196–215, 2016.

[6] B. Peherstorfer, "Sampling low-dimensional Markovian dynamics for preasymptotically recovering reduced models from data with operator inference," *SIAM Journal on Scientific Computing*, vol. 42, pp. A3489–A3515, 2020.

[7] W. I. T. Uy and B. Peherstorfer, "Operator inference of non-markovian terms for learning reduced models from partially observed state trajectories," *Journal of Scientific Computing*, vol. 88, pp. 1–31, 2021.

[8] O. Ghattas and K. Willcox, "Learning physics-based models from data: perspectives from inverse problems and model reduction," *Acta Numerica*, vol. 30, pp. 445–554, 2021.

[9] W. I. T. Uy and B. Peherstorfer, "Probabilistic error estimation for non-intrusive reduced models learned from data of systems governed by linear parabolic partial differential equations," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 55, pp. 735–761, 2021.

[10] B. Kramer, "Stability domains for quadratic-bilinear reduced-order models," *SIAM Journal on Applied Dynamical Systems*, vol. 20, pp. 981–996, 2021.

[11] N. Sawant, B. Kramer, and B. Peherstorfer, "Physics-informed regularization and structure preservation for learning stable reduced models from data with operator inference," *Computer Methods in Applied Mechanics and Engineering*, vol. 404, p. 115836, 2023.

[12] H. Sharma, D. A. Najera-Flores, M. D. Todd, and B. Kramer, "Lagrangian operator inference enhanced with structure-preserving machine learning for nonintrusive model reduction of mechanical systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 423, p. 116865, 2024.

[13] Y. Filanova, I. P. Duff, P. Goyal, and P. Benner, "An operator inference oriented approach for linear mechanical systems," *Mechanical Systems and Signal Processing*, vol. 200, p. 110620, 2023.

[14] O. Issan and B. Kramer, "Predicting solar wind streams from the inner-heliosphere to earth via shifted operator inference," *Journal of Computational Physics*, vol. 473, p. 111689, 2023.

[15] P. Jain, S. McQuarrie, and B. Kramer, "Performance comparison of data-driven reduced models for a single-injector combustion process," p. 3633, 2021.

[16] R. Swischuk, B. Kramer, C. Huang, and K. Willcox, "Learning physics-based reduced-order models for a single-injector combustion process," *AIAA Journal*, vol. 58, pp. 2658–2672, 2020.

[17] S. McQuarrie, C. Huang, and K. Willcox, "Data-driven reduced-order models via regularised operator inference for a single-injector combustion process," *Journal of the Royal Society of New Zealand*, pp. 1–18, 2021.

[18] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, "Lift & Learn: Physics-informed machine learning for large-scale nonlinear dynamical systems," *Physica D: Nonlinear Phenomena*, vol. 406, p. 132401, 2020.

[19] P. Benner, P. Goyal, B. Kramer, B. Peherstorfer, and K. Willcox, "Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms," *Computer Methods in Applied Mechanics and Engineering*, vol. 372, p. 113433, 2020.

[20] H. Sharma, Z. Wang, and B. Kramer, "Hamiltonian operator inference: Physics-preserving learning of reduced-order models for canonical Hamiltonian systems," *Physica D: Nonlinear Phenomena*, vol. 431, p. 133122, 2022.

[21] H. Sharma and B. Kramer, "Preserving Lagrangian structure in data-driven reduced-order modeling of large-scale dynamical systems," *Physica D: Nonlinear Phenomena*, p. 134128, 2024.

[22] A. Gruber and I. Tezaur, "Canonical and noncanonical Hamiltonian operator inference," *Computer Methods in Applied Mechanics and Engineering*, vol. 416, p. 116334, 2023.

[23] E. Qian, I.-G. Farcas, and K. Willcox, "Reduced operator inference for nonlinear partial differential equations," *SIAM Journal on Scientific Computing*, vol. 44, pp. A1934–A1959, 2022.

[24] R. Geelen, L. Balzano, S. Wright, and K. Willcox, "Learning physics-based reduced-order models from data using nonlinear manifolds," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 34, 2024.

[25] R. Geelen, S. Wright, and K. Willcox, "Operator inference for non-intrusive model reduction with quadratic manifolds," *Computer Methods in Applied Mechanics and Engineering*, vol. 403, p. 115717, 2023.

[26] S. McQuarrie, P. Khodabakhshi, and K. Willcox, "Nonintrusive reduced-order models for parametric partial differential equations via data-driven operator inference," *SIAM Journal on Scientific Computing*, vol. 45, pp. A1917–A1946, 2023.

[27] L. Gkimisis, T. Richter, and P. Benner, "Adjacency-based, non-intrusive model reduction for vortex-induced vibrations," *Computers & Fluids*, p. 106248, 2024.

[28] P. Mlinarić and S. Gugercin, "$\mathcal{L}_2$-optimal reduced-order modeling using parameter-separable forms," *SIAM J. Sci. Comput.*, vol. 45, pp. A554–A578, Apr. 2023.

[29] B. Kramer, B. Peherstorfer, and K. Willcox, "Learning nonlinear reduced models from data with operator inference," *Annual Review of Fluid Mechanics*, vol. 56, pp. 521–548, 2024.

[30] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2013.

[31] R. Greve and H. Blatter, *Dynamics of ice sheets and glaciers*. Springer Science & Business Media, 2009.

[32] J. D. Jakeman, "Pyapprox: A software package for sensitivity analysis, Bayesian inference, optimal experimental design, and multi-fidelity uncertainty quantification and surrogate modeling," *Environmental Modelling & Software*, vol. 170, p. 105825, 2023.

[33] M. Guo, S. McQuarrie, and K. Willcox, "Bayesian operator inference for data-driven reduced-order modeling," *Computer Methods in Applied Mechanics and Engineering*, vol. 402, p. 115336, 2022.