Structural and Multidisciplinary Optimization (2026) 69:151
https://doi.org/10.1007/s00158-026-04355-1

RESEARCH PAPER

Toward adaptive non-intrusive reduced-order models: design and challenges

Amirpasha Hedayat[1] · **Alberto Padovan**[2] · **Karthik Duraisamy**[1]

Received: 5 November 2025 / Revised: 3 May 2026 / Accepted: 8 May 2026

Abstract
Projection-based reduced-order models (ROMs) are typically deployed as static surrogates, which limits their practical utility once the system leaves the training manifold. In this work, we formalize and study adaptive non-intrusive ROMs that update both the latent subspace and the reduced dynamics online. Building on ideas from static non-intrusive ROMs, specifically, operator inference (OpInf) and the recently introduced non-intrusive trajectory-based optimization of reduced-order models (NiTROM), we propose three adaptive formulations: adaptive OpInf, which performs sequential basis and operator refits; adaptive NiTROM, which jointly optimizes the encoder, decoder, and polynomial dynamics via Riemannian optimization; and a hybrid approach that initializes NiTROM with an OpInf update. We analyze the structure and computational cost of these adaptive models, including the role of the lookback data window, adaptation frequency, and optimization budget. The methods are evaluated on a transiently perturbed lid-driven cavity flow, where static ROMs drift or destabilize when forecasting beyond the training regime. In contrast, adaptive OpInf can robustly suppress amplitude drift at modest cost, while adaptive NiTROM can attain near-exact energy tracking under frequent updates but remains sensitive to initialization and optimization depth. The hybrid method proves most reliable under regime changes and limited offline data, producing physically coherent fields and bounded energy over the prediction horizons considered. We argue that predictive claims for ROMs must be cost-aware and transparent, with clear separation of training/adaptation/deployment regimes and explicit reporting of online budgets. Overall, this work provides a practical template for building adaptive non-intrusive ROMs that incorporate online corrections and can extend predictive capability beyond the initial training manifold.

Keywords Reduced order models (ROMs) · Adaptive ROMs · Non-intrusive model reduction · Operator inference · Manifold optimization · Scientific machine learning

1 Introduction

The cost of repeatedly solving large systems of differential equations often limits the applicability of numerical simulations in several engineering tasks including optimization, uncertainty quantification, control, and digital twins. This problem can be addressed by replacing (or complementing) the expensive high-fidelity solver with a reduced-order model (ROM) that can be solved at a fraction of the original computational cost (Benner et al. 2015; Antoulas et al. 2020; Rowley and Dawson 2017). Simply put, the field of model reduction concerns itself with the development of models that can maximize computational gains while minimizing loss of accuracy. While ROMs have garnered increasing attention in the research community, their adoption in practical engineering applications remains limited.

In this work, we approach the problem of *non-intrusive* projection-based reduced-order modeling to accelerate costly simulations of physical systems. The goals of this work are two-fold: first, to characterize the challenges in achieving predictive capabilities with current ROM approaches; and second, to recommend ways forward through adaptive non-intrusive formulations. Specifically, we explore the advantages and drawbacks of formulations based on least-squares regression, as well as a novel paradigm based on a *joint*, *dynamics-aware* adaptation of the trial and test sub-

Responsible Editor: Boris Krämer.

✉ Amirpasha Hedayat
ahedayat@umich.edu

[1] Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA

[2] Department of Mechanical and Industrial Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA

Published online: 02 June 2026

spaces and latent-space dynamics. These formulations build upon Operator Inference Peherstorfer and Willcox (2016), and the recently introduced Non-intrusive Trajectory-based optimization of Reduced-Order Models (NiTROM) Padovan et al. (2024). The primary contribution of this work is therefore the development and systematic evaluation of a practical online adaptation workflow for existing non-intrusive ROMs, together with a novel model that combines the ideas of Operator Inference and NiTROM. In the rest of the section, we provide a brief review of existing static and adaptive, intrusive and non-intrusive reduced-order modeling strategies.

Given a high-fidelity model with state dimension $n \gg 1$, static and adaptive projection-based ROMs are typically obtained by constraining the state to evolve on a low-dimensional subspace with dimension $r \ll n$. Since the adaptive ROM utilizes online information, it generally operates on a lower-dimensional subspace than the static ROM. This subspace, and the latent-space coordinates on it, are implicitly defined by the choice of the so-called decoders and encoders, respectively, which are maps from the r-dimensional space to the ambient n-dimensional space, and vice versa. The evolution of the state on the target subspace is governed by a r-dimensional dynamical system, which is much cheaper to integrate than the original high-fidelity solver. Adaptive ROMs pursue interface with the high-fidelity operator to update the latent space. These building blocks (i.e., the encoder, decoder, latent-space dynamics and manifold updates) can be computed using intrusive, or non-intrusive algorithms, as discussed below.

1.1 Intrusive reduced-order models

An intrusive ROM is one where the computation of the encoder, decoder, and latent-space dynamics requires direct access to the underlying high-fidelity operator. In most projection-based ROMs, the encoder and decoder are linear maps defining a rank-r projection onto the span of the decoder, and the latent-space dynamics are obtained by projecting the full-order dynamics. Classical examples are Galerkin models obtained by *orthogonal* projection onto Proper Orthogonal Decomposition (POD) modes (Lumley 1967; Sirovich 1987; Holmes et al. 1996) associated with a representative training data set (Rowley et al. 2004; Barone et al. 2009). This procedure is "weakly" intrusive, since the computation of the encoder and decoder is purely data-driven, while the computation of the latent-space dynamics requires access only to the high-fidelity dynamics. Despite the widespread use of these models, it is well known that orthogonally projecting the system onto high-energy/high-variance subspaces can cause the accidental truncation of low-energy/low-variance features that are nonetheless dynamically important (Chomaz 2005; Rowley and Dawson 2017; Ahmed et al. 2021; Rezaian et al. 2022). This can ultimately lead to models with poor predictive accuracy, especially along transient trajectories.

Petrov–Galerkin models constructed using appropriately defined *oblique* projections can address this shortcoming. Notable examples of ROM frameworks performing such oblique projections include the following. The Least-Squares Petrov–Galerkin (LSPG) method (Carlberg et al. 2011, 2017) builds a reduced model by minimizing the residual of the discrete governing equations in a least-squares sense, leading to stable time-discrete ROMs. Another prominent approach is Balanced Truncation (BT) (Moore 1981; Willcox and Peraire 2002), which constructs a ROM by balancing the controllability and observability Gramians of the linearized system, thereby retaining the most dynamically important modes. Moreover, Balanced POD (BPOD) (Rowley 2005) provides an approximate version of BT that uses snapshots of primal and adjoint simulations to compute empirical Gramians.

More recent formulations include Trajectory-based Optimization of Oblique Projections (TrOOP) (Otto et al. 2022), which poses the ROM construction as an optimization problem seeking the best oblique projection to minimize the forecasting error along training trajectories, and Covariance Balancing Reduction using Adjoint Snapshots (CoBRAS) (Otto et al. 2023; Zanardi et al. 2025), which balances covariance operators derived from primal and adjoint snapshot ensembles to achieve robust and balanced ROMs. These methods are highly intrusive since model computation and assembly requires access to the nonlinear dynamics, as well as to the linearized adjoint of the latter around (possibly time-varying) solutions of the full-order model. This hinders the applicability of these formulations to high-dimensional systems simulated using a black-box solver (e.g. commercial and legacy codes), where access to the source code is limited for proprietary reasons.

1.2 Non-intrusive reduced-order models

Non-intrusive formulations seek to address the aforementioned problem by developing algorithms that rely exclusively on simulation (or experimental) data, without requiring access to the source code of the high-fidelity solver. These algorithms are therefore broadly applicable to systems that are simulated using commercial and legacy codes. Early examples include data-driven modal-decomposition-based methods such as Dynamic Mode Decomposition (DMD) (Schmid 2010; Rowley et al. 2009), as well as more recent extensions that employ machine-learning architectures and operator learning (Hesthaven and Ubbiali 2018; Bhattacharya et al. 2021; Xu and Duraisamy 2020). Further, Balanced Truncation can also be achieved in a non-intrusive setting Rezaian et al. (2022); Rezaian and Duraisamy (2023). Among the most widely adopted non-intrusive approaches

is Operator Inference (OpInf), which recasts the reduced model identification as a least-squares regression problem that learns polynomial operators directly in the reduced space (Peherstorfer and Willcox 2016; Kramer et al. 2024).

OpInf has been applied to complex nonlinear dynamical systems (Qian et al. 2020, 2019; Mcquarrie et al. 2021; Swischuk et al. 2020; Benner et al. 2020) and parametric problems (McQuarrie et al. 2023), and has also been extended to quadratic manifolds (Geelen et al. 2023). However, this framework seeks models in the span of an orthogonal projection operator defined in terms of high-energy/high-variance POD modes, and this can ultimately lead to models with poor forecasting accuracy. The recently proposed Non-intrusive Trajectory-based optimization of Reduced-Order Models (NiTROM) framework (Padovan et al. 2024) provides a non-intrusive learning setting that seeks to address this drawback. NiTROM jointly optimizes oblique-projection operators and reduced-order polynomial dynamics against available high-fidelity trajectory data. This approach has been shown to offer significant improvements in model forecasting accuracy, especially in systems exhibiting high sensitivity to low-energy features and large-amplitude transient growth (e.g., advection-dominated and high-shear flows).

1.3 Adaptive model reduction

A number of researchers have explored adaptive ROMs within the intrusive setting. In such paradigms, adaptation typically involves updating the reduced basis and the hyper-reduction sampling locations, while the reduced operators are automatically refreshed through their dependence on the FOM operators. Early works by Amsallem and Farhat (2008, 2011); Amsallem et al. (2012) pioneered interpolation-based ROMs that adapt across parameters by constructing local reduced bases and interpolating operators on matrix manifolds. These approaches tailor a group of reduced subspaces to different regions of the parameter domain, rather than relying on a single global subspace. Subsequent methods, such as the "Localized Discrete Empirical Interpolation Method" (LDEIM) of Peherstorfer et al. (2014), extended this idea to the hyper-reduction stage by clustering snapshots and constructing multiple local DEIM spaces and sampling points, enabling efficient nonlinear model evaluations across distinct dynamical regimes. Beyond these, Peherstorfer and collaborators advanced the field toward *truly* online adaptive frameworks. In their "Dynamic Data-Driven ROMs" Peherstorfer and Willcox (2015a), they proposed performing low-rank updates to the reduced operators directly from sensor data. This direction was further developed through the Adaptive DEIM (ADEIM) Peherstorfer and Willcox (2015b) and its successor, AADEIM Peherstorfer (2020), which perform on-the-fly, low-rank updates to both the reduced basis and the interpolation points based on localized residual information.

Other strategies, such as the lookahead sampling technique of Singh et al. (2023), predict near-future system states to guide data collection and update scheduling. From a geometric optimization viewpoint, Zimmermann et al. (2018) proposed employing the "Grassmannian Rank-One Update Subspace Estimation" (GROUSE) method (Balzano et al. 2010; Balzano and Wright 2013), which performs continuous rank-one updates to the reduced basis along geodesics on the Grassmann manifold, yielding closed-form subspace adaptation formulas. Similarly, Huang and Duraisamy (2023) introduced a rank-one adaptive ROM formulation that incrementally corrects the subspace to enforce zero projection error for each new system snapshot. Their results demonstrated that this efficient one-step update preserves long-term stability and physical fidelity, even in chaotic reacting-flow systems. Recently, Mohaghegh and Huang (2026) developed feature-guided sampling that introduces physics-informed criteria to direct online sampling toward dynamically active regions (in their cases, shocks and flame fronts), further enhancing adaptivity in hyper-reduction. Beyond these, many other efforts have explored adaptive ROMs, including hybrid snapshot and FOM/ROM coupling approaches (Zucatti and Zahr 2024; Bai and Wang 2022; Feng et al. 2021), adaptive model-reduction methods for optimization (Yano et al. 2021), basis refinement (Carlberg 2015; Etter and Carlberg 2020), and time-dependent basis frameworks Patil and Babaee (2023); Jung et al. (2025); Ramezanian et al. (2021). Despite these advances, a key limitation of all such methods is their reliance on intrusive access to the FOM. Specifically, they require explicit evaluation of residuals, Jacobians, or projection operators to perform sampling and basis updates. In non-intrusive settings, where only state and input data are available, such information is not accessible, and any parameter update must be performed in a purely data-driven sense. Another source of complexity arises from the fact that in an adaptive non-intrusive ROM, one must not only update the reduced basis but also re-identify the reduced operators themselves, which are learned directly from data rather than obtained through projection.

Some recent efforts have explored limited forms of adaptivity within non-intrusive ROMs. For instance, Geelen and Willcox (2022) proposed a localized operator inference approach in which multiple local reduced models are trained offline and an online classifier adaptively selects the most appropriate local ROM during prediction. In this model, the main adaptation mechanism is switching between pre-trained local ROMs rather than updating the operators online. More recently, Aretz and Willcox (2025) introduced the nested operator inference method, which exploits hierarchical structure in the reduced space to enable efficient offline updates and warm-started refinements of learned operators.

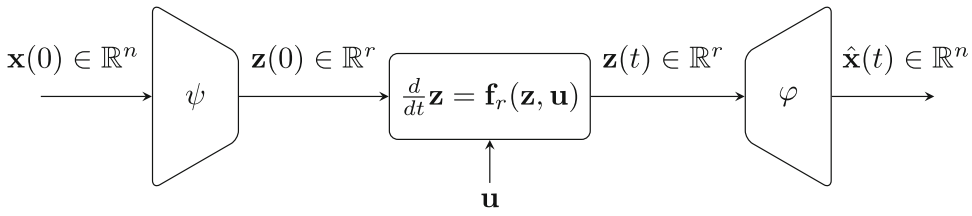


Fig. 1 Schematic of a ROM with encoder ψ, decoder φ and latent-space dynamics $\mathbf{f}_r(\mathbf{z}, \mathbf{u})$

While these works demonstrate promising directions toward more flexible non-intrusive ROMs, neither performs continual online adaptation of model parameters. Therefore, the present work introduces—to the best of our knowledge—the first set of ideas toward a general framework for adaptive non-intrusive reduced-order modeling.

Our framework combines principles of online data assimilation and continual learning with reduced-order modeling to enable models that can refine themselves on-the-fly. Building upon the OpInf and NiTROM foundations, we develop three concrete adaptive formulations: (i) Adaptive OpInf, which performs sequential updates to the basis and operators; (ii) Adaptive NiTROM, which executes joint manifold optimization; and (iii) Adaptive (hybrid) OpInf–NiTROM, which combines fast regression-based updates with short optimization-based refinements. It is important to note that all the proposed models assume a coupled forecasting-correction loop, where the ROM advances the solution over a short horizon, after which a high-fidelity state snapshot is obtained and used to update the reduced basis and operators. This assumption naturally restricts the framework to settings such as those in simulation environments where the high-fidelity operator can be queried, or in digital twin scenarios where measurements or sensor data provide state information. Conversely, the proposed framework is not intended for fully autonomous long-horizon prediction without external data, as its predictive stability relies on these periodic corrections. Through comprehensive numerical experiments on a canonical nonlinear flow problem, we assess the accuracy and practicality of these adaptive formulations. Our observations indicate that, when the system remains close to its training manifold, all tested adaptive models prevent amplitude drift and stabilize long-term predictions. Moreover, when the system undergoes regime changes—here referring to transitions between qualitatively different dynamical behaviors, such as the shift from transient growth to sustained oscillatory dynamics—or is minimally trained offline, hybrid adaptation provides the most robust recovery and physically meaningful results over the considered prediction horizons. Note that these horizons remain moderate in length, as prediction errors inevitably accumulate over time and the ROM trajectory eventually deviates from the true dynamics. These findings suggest that non-intrusive online adaptation has the potential to extend the predictive range of ROMs, marking a promising step toward *truly* predictive reduced models. That said, we emphasize that the present frameworks are still at an early stage of development, and issues of computational efficiency remain open. It is the authors' opinion that the ROM community (including our own work) require a stricter, cost-aware standard for claims of *prediction,* with clear separation of training and deployment regimes and full reporting of offline/online computational budgets. Reproducing previously seen dynamics may be useful in some circumstances, but it should not be presented as evidence of generalization without validated out-of-manifold performance under stated resource and time constraints. This work aims to provide some quantification in this sense. Public Python implementations of the algorithms discussed in this work are hosted at https://github.com/APHedayat/Adaptive-NiTROM.

The remainder of this paper is organized as follows. Section 2 presents problem formulation and the design choices in an adaptive non-intrusive ROM framework in detail. Section 3 outlines the full-order model, numerical setup, and evaluation metrics used to test the adaptive ROMs. Section 4 reports and analyzes the results across progressively more challenging scenarios. Finally, Sec. 5 summarizes the main findings, discusses limitations, and outlines future directions.

2 Methodology

Consider a nonlinear high-dimensional dynamical system

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) \;\; ; \;\; \mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)), \tag{1}$$

with state $\mathbf{x}(t) \in \mathbb{R}^n$, control input $\mathbf{u}(t) \in \mathbb{R}^m$, and measured output $\mathbf{y}(t) \in \mathbb{R}^q$. This system represents the full-order model (FOM) that is often too expensive to simulate when the state dimension n is very large. A reduced-order model (ROM) is another dynamical system, usually described by the pipeline in Fig. 1. The key components of a ROM are (i) an encoder, which is a possibly nonlinear map from the n-dimensional space to the r-dimensional space (with $r \ll n$), (ii) a decoder, which is a possibly nonlinear map from the r-dimensional space to the n-dimensional space, and (iii) the latent-space dynamics $\mathbf{f}_r$ used to evolve the reduced-order state $\mathbf{z}$. In this paper, we constrain our attention to the case where the encoder and decoder are linear maps, and the encoder is a left inverse of the decoder. Then,

we may express the encoder as $\psi = \Psi^\top$, and the decoder as $\varphi = \Phi(\Psi^\top\Phi)^{-1}$, where $\Phi,\ \Psi \in \mathbb{R}^{n\times r}$ define, respectively, the decoder basis (spanning the *trial* space) and the encoder basis (spanning the *test* space). By virtue of the fact that the encoder is a left inverse of the decoder, the encoder/decoder pair define a projection operator

$$\mathbb{P} \triangleq \Phi(\Psi^\top\Phi)^{-1}\Psi^\top : \mathbb{R}^n \to \mathbb{R}^n, \quad \mathbb{P}^2 = \mathbb{P}. \tag{2}$$

The ROM in Fig. 1 approximates the FOM solution in the range of $\mathbb{P}$. When $\text{span}(\Phi) \equiv \text{span}(\Psi)$, we say that the projection is orthogonal, and it is oblique otherwise. We remark that encoders and decoders can also be defined by nonlinear functions (Lee and Carlberg 2020; Fresca et al. 2021; Conti et al. 2023; Otto et al. 2023), although strongly enforcing that the encoder ψ be a left inverse of the decoder φ requires some care. In any case, this is beyond the scope of this work.

The latent-space dynamics $\mathbf{f}_r$ in Fig. 1 can be defined in several ways. Perhaps the most well known is via Petrov–Galerkin projection, where given a projection operator $\mathbb{P}$ defined as in (2), the latent-space dynamics is obtained by projecting the full-order dynamics $\mathbf{f}$ as follows:

$$\mathbf{f}_r(\mathbf{z},\mathbf{u}) = \Psi^\top\mathbf{f}(\Phi(\Psi^\top\Phi)^{-1}\mathbf{z},\mathbf{u}). \tag{3}$$

This procedure is generally considered (weakly) intrusive, because it requires direct access to the full-order dynamics $\mathbf{f}$. Alternatively, the latent-space dynamics can be inferred from data using non-intrusive approaches that require no access to $\mathbf{f}$. For instance, this is the case of OpInf (discussed in greater detail in the following section), where, assuming dynamics of polynomial form

$$\mathbf{f}_r(\mathbf{z},\mathbf{u}) = A_r\mathbf{z} + H_r : (\mathbf{z}\mathbf{z}^\top) + B_r\mathbf{u} + \ldots, \tag{4}$$

the reduced-order tensors $A_r \in \mathbb{R}^{r\times r}$, $H_r \in \mathbb{R}^{r\times r\times r}$, $B_r \in \mathbb{R}^{r\times m}$, etc. are learned from data by solving a least-squares optimization problem.

Clearly, whether we are operating in a static setting or in an adaptive setting, the encoder, decoder, and latent-space dynamics determine the accuracy of the ROM and its ability to forecast the system's dynamics. In this paper, we consider adaptive, *non-intrusive* models with polynomial latent-space dynamics as in Eq. (4), and with linear encoders/decoders defining (possibly oblique) projectors as in Eq. (2). In particular, we will consider three adaptive paradigms: one based on OpInf, one on NiTROM, and one on a combination of these. Before presenting the adaptive methodologies, it is instructive to review their static counterparts.

2.1 Static ROMs

2.1.1 Operator inference

As previously mentioned, Operator Inference (OpInf) (Peherstorfer and Willcox 2016; Kramer et al. 2024) is a non-intrusive model-reduction framework where one usually seeks a polynomial ROM evolving on the range of an *orthogonal* projection defined by (orthonormal) Proper Orthogonal Decomposition (POD) modes, Φ. In the terminology of Fig. 1, the encoder is defined as $\psi = \Phi^\top$, the decoder as $\varphi = \Phi$, and the latent-space dynamics $\mathbf{f}_r$ as in Eq. (4). Mathematically, given the leading r orthonormal POD modes $\Phi \in \mathbb{R}^{n\times r}$ computed from a high-fidelity training data set $\{\mathbf{x}^{(i)}(t_j)\}$ (where $\mathbf{x}^{(i)}(t_j) \in \mathbb{R}^n$ is the solution vector at time t_j along trajectory i), we compute the latent-space coordinates $\mathbf{z} = \Phi^\top\mathbf{x}$ and their time derivatives $d\mathbf{z}/dt = \Phi^\top d\mathbf{x}/dt$, and we solve the following regularized least-squares optimization problem:

$$\begin{aligned}\min_{\Theta\in\mathcal{M}_{\text{OpInf}}} J = \frac{1}{ML}\sum_{i=0}^{L-1}\sum_{j=0}^{M-1}\Big\| \frac{d}{dt}\mathbf{z}^{(i)}(t_j) - \mathbf{f}_r(\mathbf{z}^{(i)}(t_j),\\ \mathbf{u}^{(i)}(t_j))\Big\|^2 + \lambda\|\text{Mat}(H_r)\|_F^2,\\ \Theta = (A_r, H_r, B_r, \ldots),\end{aligned} \tag{5}$$

where $\mathbf{f}_r$ is defined as in Eq. (4) and $\mathcal{M}_{\text{OpInf}} = \mathbb{R}^{r\times r} \times \mathbb{R}^{r\times r\times r} \times \mathbb{R}^{r\times m} \times \ldots$. Here, L and M are the number of training trajectories and training snapshots per trajectory, respectively. The second term in Eq. (5) corresponds to Tikhonov regularization applied to the matricized quadratic tensor H_r, where λ is the regularization parameter. While OpInf is an elegant formulation with a closed-form solution, it was recently shown (Padovan et al. 2024) that it often struggles to accurately forecast trajectories through large-amplitude transients. This is because, although optimal at statically reconstructing the state, the underlying orthogonal projection $\mathbb{P} = \Phi\Phi^\top$ onto POD modes can accidentally truncate low-energy features that are nonetheless important for the dynamical evolution of the system.

2.1.2 NiTROM

The NiTROM framework recently introduced in (Padovan et al. 2024) is a non-intrusive formulation developed to address the shortcomings of non-intrusive models obtained by orthogonal projection onto high-energy/high-variance subspaces. More specifically, leveraging the fact that *oblique* projections have been shown to be better suited for capturing the dynamically relevant mechanisms of FOMs exhibiting large-amplitude transient growth, NiTROM *simultane-*

ously seeks optimal encoders $\psi = \Psi^\top$, decoders $\varphi = \Phi\left(\Psi^\top\Phi\right)^{-1}$, and polynomial latent-space dynamics (4) by solving an optimization problem on a matrix manifold. Given training data $\{\mathbf{x}^{(i)}(t_j)\}$, where, once again, $\mathbf{x}^{(i)}(t_j) \in \mathbb{R}^n$ is the solution vector at time t_j along trajectory i, we solve the following problem:

$$\begin{aligned} \min_{\Theta\in\mathcal{M}_{\text{NiTROM}}} \quad & J = \frac{1}{ML}\sum_{i=0}^{L-1}\sum_{j=0}^{M-1}\|\mathbf{y}^{(i)}(t_j) - \hat{\mathbf{y}}^{(i)}(t_j)\|^2 \\ \text{subject to:} \quad & \frac{d}{dt}\mathbf{z}^{(i)} = \mathbf{f}_r(\mathbf{z}^{(i)}, \mathbf{u}^{(i)}), \quad \mathbf{z}^{(i)}(t_0) = \Psi^\intercal\mathbf{x}^{(i)}(t_0) \\ & \hat{\mathbf{y}}^{(i)} = \mathbf{h}\left(\Phi\left(\Psi^\top\Phi\right)^{-1}\mathbf{z}^{(i)}\right) \\ & V = \text{Range}\left(\Phi\right), \end{aligned} \tag{6}$$

with $\mathbf{y}^{(i)}(t_j) = \mathbf{h}\big(\mathbf{x}^{(i)}(t_j)\big)$, parameters $\Theta = (V, \Psi, A_r, H_r, B_r, \ldots)$, and $\mathcal{M}_{\text{NiTROM}}$ a matrix manifold defined as

$$\mathcal{M}_{\text{NiTROM}} = \mathcal{G}_{n,r} \times S_{n,r} \times \mathbb{R}^{r\times r} \times \mathbb{R}^{r\times r\times r} \times \mathbb{R}^{r\times m} \times \ldots. \tag{7}$$

Here, $\mathcal{G}_{n,r}$ is the Grassmann manifold (i.e., the space of r-dimensional subspaces of $\mathbb{R}^n$) and $S_{n,r}$ is the Stiefel manifold (i.e., the space of $n \times r$ orthonormal matrices). This particular parameterization arises from the nature of the optimization problem. In particular, we optimize subspaces over the Grassmann manifold because the optimization problem (6) is a function of the subspace V spanned by the decoder $\Phi\left(\Psi^\top\Phi\right)^{-1}$, rather than of the matrix representative $\Phi\left(\Psi^\top\Phi\right)^{-1}$ itself. The encoder $\Psi^\top$, on the other hand, is optimized over the Stiefel manifold to strongly enforce the necessary condition $\text{rank}(\Psi) = r$ in order to define a projection $\mathbb{P}$ as in Eq. (2). The other (linear) spaces (e.g., $\mathbb{R}^{r\times r}$, etc.) in the definition of $\mathcal{M}_{\text{NiTROM}}$ are used to parameterize the reduced-order tensors defining the latent-space polynomial dynamics in Eq. (4).

This formulation has several advantages over the previously discussed OpInf. First, by allowing for the encoder and decoder to span different spaces, the model is more expressive and can therefore detect the low-energy sensitivity mechanisms that are important for accurate forecasting. Second, the cost function measures the solution error in the original high-dimensional space, rather than measuring deviations in the time-rate-of-change of the latent-space vector, ultimately encouraging stability of the latent-space dynamics. Third, all ROM components (encoder, decoder, and dynamics) are optimized simultaneously, thus promoting the computation of *dynamics-aware* projection operators leading to models that are capable of predicting the FOM solution along transient trajectories. A drawback of this formulation, however, is that the optimization problem is non-convex, meaning that the optimization landscape may contain multiple local minima and non-unique minimizers. Additionally, unlike OpInf, the problem does not admit a closed-form solution. Nonetheless, it was shown in (Padovan et al. 2024) that the gradient of the cost function J with respect to the parameters Θ can be computed in closed-form using a fully non-intrusive adjoint-based formulation. This implies that the optimization problem can be solved efficiently without relying on possibly expensive automatic differentiation.

2.2 Adaptive ROMs

Adaptive model-reduction methods rely on interactions with the full-order, high-fidelity solver to update the model parameters and maintain a desired level of accuracy. The adaptation setup used throughout this paper is best described graphically in Fig. 2. Here, Δt is the FOM time step, and $Z\Delta t$ is the length of the ROM forecast window. In other words, time advancement from time $t_j - Z\Delta t$ to t_j is performed using the reduced-order model. Every $Z\Delta t$ time units, the ROM interacts with the FOM and collects a new sample of high-fidelity data, which is stored in a data matrix

$$\begin{aligned} \mathcal{D}_j &= \left\{Q(t_{j-(M-1)Z}),\ \ldots,\ Q(t_{j-Z}),\ Q(t_j)\right\}, \\ Q(t_i) &= (\mathbf{x}(t_i), \dot{\mathbf{x}}(t_i), \mathbf{u}(t_i)). \end{aligned} \tag{8}$$

$\mathcal{D}_j$ is a moving-window data matrix since, at any given time t_j, it contains M high-fidelity samples, and when a new sample becomes available, the "oldest" sample is discarded. This way, older samples that are no longer representative of the current state of the system are naturally forgotten, enabling the model to emphasize the most recent and relevant patterns. The time derivative of the new snapshot is obtained by evaluating the right-hand side of the full-order model (Eq. (1)) at that snapshot. Specifically, given high-dimensional snapshots $x(t_j)$, the corresponding derivative is computed as $\dot{x}(t_j) = f(x(t_j))$, where $f(\cdot)$ denotes the FOM dynamics operator. We note that this evaluation introduces a degree of intrusiveness, as it requires access to the FOM right-hand side. We adopt this choice here to isolate the behavior of the adaptive ROM framework from finite difference approximation errors. In a fully non-intrusive deployment, the time derivatives can instead be approximated from finite differences of state snapshots. It is worth mentioning that performing multiple consecutive FOM steps per adaptation is also possible, and that the single-step choice here reflects a cost-aware design decision.

The goal of each adaptation is to update the ROM parameters Θ_{j-1} to a refined set Θ_j that better represents the newly observed dynamics contained in $\mathcal{D}_j$. This update is the solu-

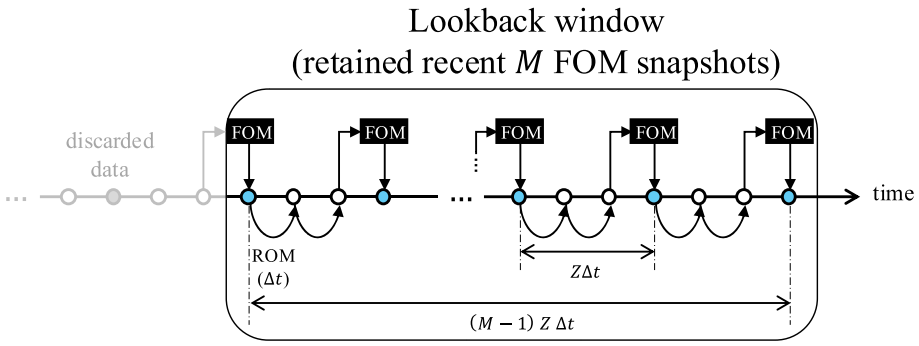


Fig. 2 Schematic for the adaptation setup. Highlighted steps show FOM snapshots kept in the moving window

tion of a localized learning problem of the form

$$\Theta_j = \arg\min_{\Theta \in \mathcal{M}} J_j(\Theta), \tag{9}$$

where J_j is some objective cost function formulated over the training data in $\mathcal{D}_j$. For instance, when the model is adapted according to an OpInf-based scheme, the cost function J_j is given by the cost function J in Eq. (5) with $L = 1$, and $\mathcal{M} = \mathcal{M}_{\text{OpInf}}$. Similarly, when the model is adapted using a NiTROM-like rule, the cost function is given by Eq. (6), also with $L = 1$ and with $\mathcal{M} = \mathcal{M}_{\text{NiTROM}}$.

Adaptations are triggered at discrete intervals during online integration. In the simplest case, updates occur every Z time steps. This fixed schedule simplifies analysis and allows direct control over computational cost. An alternative strategy could employ adaptive triggering based on residual norms, which is a natural extension for future work. The three main hyperparameters controlling the overall behavior of the adaptive ROM are therefore (M, Z, K), where M is the number of high-quality snapshots retained in the moving (lookback) window, Z is the number of time steps between successive adaptations (adaptation window), and K is the number of optimization steps (if doing gradient-based online learning, as in the NiTROM framework) per adaptation. These quantities jointly control the trade-off between accuracy and computational efficiency. The adaptation framework is described in pseudocode in Alg. 1, where we describe the online forecasting phase and the ROM–FOM interaction.

Remark 1 We emphasize that decoding a ROM-predicted state produces a state in the decoder range, which, in general, does not span the FOM solution manifold. The role of periodic FOM interaction is to correct accumulated model error before it grows irreversibly, by injecting newly observed dynamics back into the reduced model. In practice, this requires that the ROM prediction remains sufficiently close to the true trajectory between adaptation events, a condition that is controlled by the adaptation window Z. Our numerical results in Sec. 4 demonstrate that when this condition is respected, a one-step FOM advancement from the lifted state produces physically consistent snapshots that enable effective adaptation.

Algorithm 1 Adaptive Non-Intrusive ROM Framework

Require: Initial training data set $\mathcal{D}_0$ with M samples sampled at times $t \in [t_0 - (M-1)Z\Delta t, t_0]$; reduced dimension r; adaptation window Z; lookback window M; optimization budget K (if applicable);

$j \leftarrow 0$
$\Theta_j \leftarrow$ InitializeModel$(\mathcal{D}_j, r)$
for $t \in t_0 + \Delta t\{0, 1, 2, \ldots\}$ **do**
  $\mathbf{z}(t) \leftarrow$ AdvanceROMForOneTimeStep$(\Theta_j, \mathbf{z}(t - \Delta t))$
  if $(t - t_0 + \Delta t)$ mod $(Z\Delta t) = 0$ **then**
    Decode $\hat{\mathbf{x}}(t - \Delta t) \leftarrow \Phi_j \left(\Psi_j^\top \Phi_j\right)^{-1} \mathbf{z}(t - \Delta t)$
    $\mathbf{x}(t) \leftarrow$ AdvanceFOMForOneTimeStep$(\hat{\mathbf{x}}(t - \Delta t))$
    $\mathcal{D}_{j+1} \leftarrow$ UpdateDataMatrix$(\mathcal{D}_j, \mathbf{x}(t))$
    $\Theta_{j+1} \leftarrow$ AdaptModel$(\Theta_j, \mathcal{D}_{j+1}, Z, M, K)$
    $j \leftarrow j + 1$
  end if
end for

2.2.1 Adaptive operator inference

Adaptive OpInf decomposes the online update in Eq. (9) into two sequential stages: (1) update the reduced basis using the new information contained in the moving window $\mathcal{D}_j$, and (2) refit the reduced operators by solving a least-squares problem on the same window.

Stage 1: Basis update. At the jth adaptation event, the most recent M high-quality snapshots form the data window $\mathcal{D}_j$ as defined in Eq. (8). As $\mathcal{D}_j$ is updated in time, its span changes, and so should the POD modes Φ used for OpInf (see Sec. 2.1.1 for details). Several strategies are available to update the basis.

- *Windowed SVD:* Compute the singular value decomposition (SVD) over the window and select the r leading left singular vectors. This approach ignores the previous basis and re-builds the basis from scratch at every adaptation. This provides the optimal rank-r subspace (in the ℓ_2 sense) for the current window, but requires a full SVD at every adaptation. This method has been used throughout

our experiments to establish a solid baseline, even though more efficient techniques (described below) exist.

- *Incremental SVD (iSVD):* Update the basis with the new snapshot $\mathbf{x}(t_j)$, using recursive low-rank updates. This provides an efficient, incremental technique that avoids recomputing the SVD (See, e.g., Brand's iSVD algorithm Brand 2002).
- *Rank-one update:* The approach of Huang and Duraisamy (2023) provides an efficient rank-one correction to the old basis that effectively removes the projection error from the new snapshot. This is a rather aggressive approach, as old snapshots have no direct contribution to the rank-one update and influence the new basis only indirectly through the previous basis. The method is effective when spatially localized updates are made to the basis at each time step, which typically requires intrusive access to the FOM operators at selected points in the domain. This operation is not feasible in the non-intrusive setting considered here, and therefore, rank-one updates are not explored in this work.

Stage 2: Operator refit. Once the basis Φ_j has been updated, the reduced operators $\Theta_j = (A_{r\,j}, H_{r\,j}, B_{r\,j})$ are re-computed by solving the optimization problem outlined in Sec. 2.1.1 with $L = 1$. The sequential two-stage update of Adaptive OpInf provides a lightweight yet effective mechanism for online learning of model parameters.

Remark 2 In the OpInf setting, uniqueness of the least-squares solution requires the regression data matrix to have full column rank, which typically imposes a lower bound on the number of snapshots relative to the number of operator coefficients. In the present adaptive setting, we could face cases where the number of snapshots kept in the moving window does not satisfy the full column rank condition. To improve conditioning in these cases, we employ regularized least squares (as in Eq. (5)), which yields a well-posed solution suitable for short-horizon prediction. To avoid repeated hyperparameter tuning during deployment, the same regularization parameter found over the offline training set is reused for the online operator inference updates. While this choice may not be optimal, it provides an acceptable baseline. There exist potential alternatives, such as truncated SVD used in Benner et al. (2022), that are left for future work.

2.2.2 Adaptive NiTROM

Adaptive NiTROM extends the offline NiTROM optimization framework to the online, streaming context. Unlike Adaptive OpInf, which treats the basis and reduced operators as separate entities, Adaptive NiTROM performs a joint optimization, as discussed in Sec. 2.1.2. This enables coherent corrections of both the reduced subspaces and the latent dynamics in a single optimization step, ensuring that the basis and operators remain mutually consistent with the evolving data. Formally, at each adaptation event, we solve the NiTROM problem in Sec. 2.1.2 against the training data in $\mathcal{D}_j$. The optimization is warm-started from the previous parameters Θ_{j-1}. The adaptive update proceeds using Riemannian gradient-based steps on the product manifold $\mathcal{M}_{\text{NiTROM}}$.

2.2.3 Adaptive (hybrid) OpInf–NiTROM

While Adaptive NiTROM provides the most principled framework for trajectory-level adaptation, its convergence and cost depend strongly on the initialization point Θ_{j-1}. If the system dynamics have changed significantly since the last adaptation, starting directly from Θ_{j-1} can lead to poor local minima or slow convergence. On the other hand, Adaptive OpInf can rapidly compute an approximate update using least squares on the new data window, but lacks the geometry-aware coupling between basis and operators. The central idea of a hybrid approach, which we call Adaptive OpInf–NiTROM, is to use a fast OpInf update as a *jump-start* for the subsequent manifold optimization, thereby providing a high-quality initialization that places the optimization algorithm near a favorable region of the cost landscape.

2.3 Computational considerations

The computational cost of an adaptive ROM is a critical factor in assessing its practicality. While model reduction is motivated by the desire for faster prediction, adaptivity introduces additional overhead during the online phase. Here we outline the computational scaling of each major component of the proposed adaptive framework.

Between two successive adaptations, the ROM is integrated entirely in the reduced space of dimension r. For a ROM with a polynomial nonlinearity of degree α, the number of distinct degree-α monomials in r variables is $s = \binom{r+\alpha-1}{\alpha} = \mathcal{O}(r^\alpha)$, and therefore the evaluation of the right-hand side of Eq. (4) at each time step scales as $\mathcal{O}(r^{\alpha+1})$, dominated by the highest-degree term. Since $r \ll n$, this cost is negligible compared with both the high-fidelity solver and the subsequent adaptation step.

At each adaptation event, a new high-quality snapshot must be obtained. In a digital twin setting, this corresponds to sensor data and therefore incurs no computational cost. In simulation-based environments, however, a new snapshot is obtained by lifting the ROM prediction to the full space and performing a one-step FOM advancement. The lifting cost scales as $\mathcal{O}(nr)$, while the FOM advancement typically scales as $\mathcal{O}(n^\beta)$, where $\beta > 1$ for implicit schemes, depending on the solver type and sparsity structure of the system

matrices. In realistic CFD solvers, this one-step FOM query often dominates the total cost of an adaptation cycle.

After acquiring the new snapshot, the ROM parameters are updated using one of the adaptive formulations discussed previously. For Adaptive OpInf, the total per-update cost includes the basis update, the projection between full and reduced spaces, and the least-squares refit of the reduced operators. The basis update can be performed using several strategies: a windowed SVD with cost $\mathcal{O}(nM^2)$, an incremental SVD (iSVD) with $\mathcal{O}(nr)$ cost, or a rank-one update with similar scaling to iSVD. Projecting the data to and from the reduced space costs $\mathcal{O}(Mnr)$ per window. For a polynomial model of degree α, assembling the OpInf data matrices over M snapshots incurs $\mathcal{O}(Mr^{\alpha})$ work. The least-squares refit then has an approximate upper bound of $\mathcal{O}(Mr^{3\alpha+1})$, which reflects the exponential growth in α (see Sec. 3.5 of Peherstorfer and Willcox 2016). Putting these pieces together, a single Adaptive OpInf update (with windowed SVD) scales as

$$\mathcal{O}(nM^2 + Mnr + Mr^{3\alpha+1}).$$

The Adaptive NiTROM model, by contrast, involves a joint manifold optimization with K Riemannian iterations per adaptation. Each iteration requires one forward and one adjoint integration of the reduced model across a window of M time steps. The corresponding cost scales as $\mathcal{O}(KMr^{\alpha+1})$, where α is the polynomial degree of the ROM. Additional terms arise from matrix–vector products involving the encoder and decoder bases (Φ, Ψ) on the Stiefel and Grassmann manifolds, which approximately add $\mathcal{O}(Knr^2)$ per update (see Sec. 2.4 of Padovan et al. 2024). Hence, the total per-adaptation complexity of Adaptive NiTROM is

$$\mathcal{O}(Knr^2 + KMr^{\alpha+1}).$$

The Adaptive OpInf–NiTROM method combines the two mechanisms by first performing a rapid OpInf update, followed by a small number of NiTROM refinement steps. Its total cost can be estimated as

$$\mathcal{O}(nM^2 + Mnr + Mr^{3\alpha+1}) + \mathcal{O}(Knr^2 + KMr^{\alpha+1}).$$

In practice, the cost is dominated by the number of NiTROM iterations, which is typically small ($K \sim$ 5–10) since the OpInf stage provides a near-optimal initialization.

Table 1 summarizes the asymptotic per-adaptation costs of all adaptive models considered in this work. Note that only the terms scaling with the full dimension n are kept, as all other terms are negligible. A quantitative comparison is reported in Sec. 4 (see Table 2), where wall-clock time of each step is presented. In general, the computational overhead introduced by adaptation remains small relative to high-fidelity simulation, but it is non-negligible for frequent updates or several optimization steps, and thus represents a practical constraint on real-time deployment.

3 Experimental setup

3.1 Full-order model (FOM)

Our test problem is the canonical lid-driven cavity flow at Reynolds number Re $= 8300$, identical to that in Sec. 5 of (Padovan et al. 2024). The system is governed by the two-dimensional incompressible Navier–Stokes equations in non-dimensional form

$$\begin{aligned} \frac{\partial \mathbf{V}}{\partial t} + \mathbf{V} \cdot \nabla \mathbf{V} &= -\nabla P + \mathrm{Re}^{-1} \nabla^2 \mathbf{V}, \\ \nabla \cdot \mathbf{V} &= 0, \end{aligned} \tag{10}$$

where $\mathbf{V} = (U, V)$ is the two-dimensional velocity vector, P is the pressure, and the solution domain is the unit square $D = [0, 1] \times [0, 1]$ with zero velocity at all walls except $U = 1$ at the top lid. Spatial discretization is performed using a second-order finite-volume scheme on a 100×100 fully staggered uniform grid, through which the projection step automatically enforces the divergence-free constraint and removes pressure from the state variables. The resulting semi-discrete system contains $n = 2 \times 10^4$ unknowns. In our experiments, the observation operator $\mathbf{h}$ (see Eq. (1)) is chosen to be the identity, giving the full velocity field $\mathbf{v}$ as the output.

The system is driven by a spatially localized input acting on the x-momentum equation,

$$\begin{aligned} B(x, y)\, w(t) &= \exp\big\{ -5000\big((x - x_c)^2 + (y - y_c)^2\big)\big\}\, w(t), \\ x_c &= y_c = 0.95, \end{aligned} \tag{11}$$

where $B(x, y)$ defines the input map and $w(t)$ is a scalar forcing function. Time integration uses a two-step Adams–Bashforth scheme with step $\Delta t = 0.0025$ to retain stability. The flow admits a stable steady state $\overline{\mathbf{V}}$ at this Reynolds number but exhibits strong transient growth when perturbed. We initialize the simulation from the steady state and excite the system with a sinusoidal forcing,

$$w(t) = 0.1 \sin(4t), \tag{12}$$

which injects momentum near the upper-right corner of the cavity, at $x_c = y_c = 0.95$, as mentioned in Eq. (11). This continuous forcing generates rich transient and oscillatory behavior (Fig. 3), providing a suitable benchmark for testing adaptive ROMs.

Table 1 Asymptotic computational cost per adaptation. Here, n is the full dimension, r the reduced dimension, α the polynomial degree of the reduced operators, M the number of snapshots in the moving window, and K the number of optimization iterations

Model	Operation	Cost
Adaptive OpInf	Basis + operator update	$\mathcal{O}\big(n(M^2 + Mr)\big)$
Adaptive NiTROM	Joint manifold update	$\mathcal{O}(Knr^2)$
Adaptive OpInf–NiTROM	Hybrid	$\mathcal{O}(n(M^2 + Mr + Kr^2))$
FOM one-step advancement	High-fidelity solve	$\mathcal{O}(n^\beta),\ \beta > 1$

Table 2 Average wall-clock time (in milliseconds) for a single execution of each operation in the adaptive non-intrusive ROM framework. All wall-clock times were measured on a local machine equipped with an Apple M3 Pro chip, and are averaged over 10 runs to mitigate measurement noise. Reported values correspond to a lookback window of $M = 10$ snapshots. Note that methods are initialized from the same offline phase; therefore, the offline training cost is identical across methods and is not included here

Operation	Description	Wall-clock time [ms]
ROM step	one reduced-order model time step	0.322
FOM step	one full-order model time step	2.33
SVD*	one complete SVD over the window	2.85
OpInf*	one least-squares refit of reduced operators	7.49
NiTROM*	one manifold optimization iteration	31.2

* represents the fact that these algorithms have not been modified from their offline versions

Remark 3 This problem was chosen because it exhibits a range of features (e.g., transient growth and traveling-wave-like structures) that appear in less canonical flow configuration and pose a challenge to the development of accurate ROMs. Ablation studies and extensive numerical experiments in different training/testing regimes expose strengths and weaknesses of the proposed non-intrusive adaptive ROMs approaches, and shine a light on the research questions that must be addressed before these ROMs can be successfully deployed on more large-scale problems.

3.2 Training and test windows

Throughout the rest of the paper, we test our adaptive frameworks in three different regimes with progressively shorter offline training windows. When we force the flow from rest with the forcing signals in Eqs. (11) and (12), the flow undergoes a sharp transient energy growth (due to the underlying non-normality of the dynamics) before settling onto a (forced) time-periodic limit cycle. This behavior is clearly visible in Fig. 4, where the energy $E(t)$ will be defined later. The three training/testing regimes are discussed below, with training happening over the temporal interval $[t_0, t_1]$ and testing over $(t_1, t_2]$.

1. **Case 1 (rich training):** the final offline time t_1 is well inside the oscillatory regime, providing a rich data set that already contains most of the information of the post-transient dynamics (see Fig. 4a).
2. **Case 2 (regime change):** the final offline time t_1 falls at the very beginning of the post-transient phase, requiring the ROM to adapt to new oscillatory patterns during online prediction (see Fig. 4b). Here, "regime change" denotes the transition from transient to sustained oscillatory dynamics that are absent from the offline training window.
3. **Case 3 (minimal training):** t_1 is before the transient onset, corresponding to extremely limited and low-energy training data, where adaptation must recover unseen dynamics from scratch (see Fig. 4c).

Remark 4 The final test time t_2 is chosen separately for each case to maintain a comparable ratio between the online prediction phase and the available offline training data. In cases 2 and 3, the offline training window ends significantly earlier, and fixing t_2 to the value used in case 1 would impose a disproportionately longer extrapolation horizon. Moreover, since online adaptation relies on FOM queries initialized from lifted ROM predictions, prediction errors inevitably accumulate over long horizons, particularly in challenging regimes. Accordingly, we report results over shorter but still highly challenging prediction intervals in these cases.

It is reasonable to question whether these tests are fair for static ROMs, which, by definition, rely on a larger corpus of training data. However, our experimentation with a wider range of training data on more complex problems (e.g., Huang et al. 2022; Arnold-Medabalimi et al. 2022) has shown that these test cases are representative of challenges faced

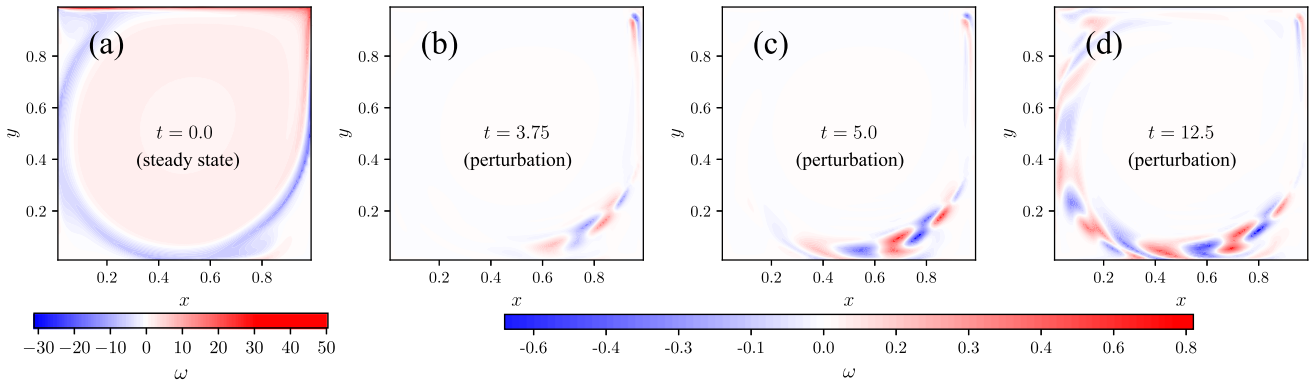


Fig. 3 Transient 2-D lid-driven cavity flow. The leftmost field depicts the vorticity field from the steady-state solution of the system at $Re = 8300$, and the remaining plots show vorticity fluctuations, showing how the dynamics evolve in time

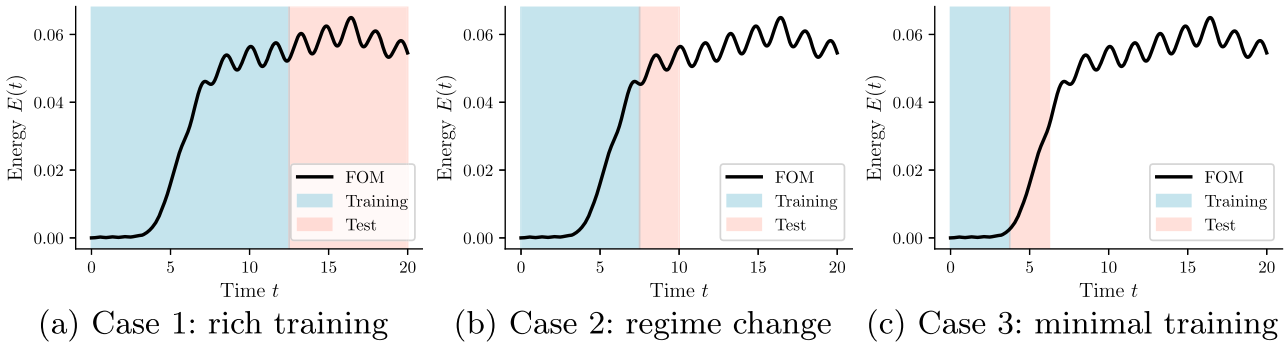


Fig. 4 Illustration of the three training/test window configurations. Each panel shows the temporal evolution of the system energy, with the blue-shaded region denoting the offline (training) window $[t_0, t_1]$ and the red-shaded portion showing the online (test) window $(t_1, t_2]$. The training interval progressively shortens from case 1 to case 3

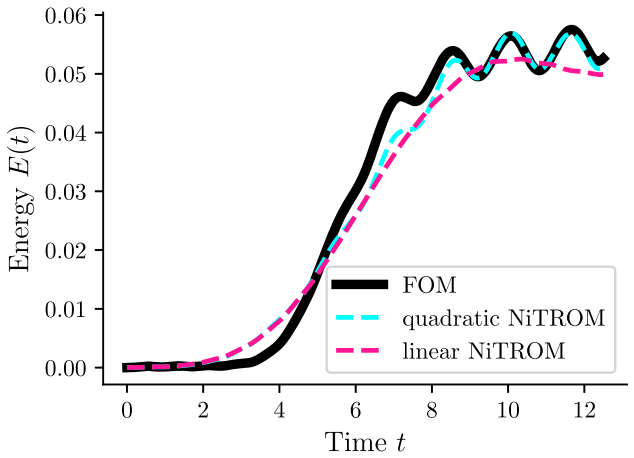


Fig. 5 Performance of 10-dimensional linear vs. quadratic static NiTROM models against the training trajectory. We notice that the linear model is unable to track the oscillatory motion of the energy, while the quadratic ROM captures the true trajectory

by static ROMs in settings with strong transients, transport-dominated physics, and particularly in chaotic problems where every snapshot is unique, and out-of-manifold.

3.3 Offline/static ROM setup

Static models (Galerkin, OpInf, and NiTROM) are trained on the snapshots from $[t_0, t_1]$. The static Galerkin ROM is constructed using the same POD basis as the static OpInf model. In all cases, we design a quadratic ROM as given in Eq. (4), since a purely linear formulation was found to be too restrictive. In fact, it fails to reproduce the cyclic energy behavior observed during training, whereas the quadratic model accurately captures these in-distribution oscillations (Fig. 5). Indeed, the presence of quadratic terms in the reduced dynamics is physically expected, since the governing Navier–Stokes equations in Eq. (10) contain a quadratic convection term. The reduced dimension r is chosen as $r = 10$, which captures the dominant energetic modes without excessive computational cost. While in classical static ROMs, r is often selected based on cumulative POD energy content, this restriction is less critical in the adaptive context because

the model parameters evolve over time and can gradually absorb new information. For static OpInf, the Tikhonov regularization parameter λ in Eq. (5) is selected by sweeping over a range of candidate values and, for each candidate, evaluating the resulting OpInf model using the NiTROM trajectory-based forecasting loss in Eq. (6) over the training trajectory. The value of λ that minimizes this forecasting loss is then retained. The resulting values are $\lambda = 10^{-4}$ for case 1, $\lambda = 10^{-2}$ for case 2, and $\lambda = 10^{-1}$ for case 3. The resulting static OpInf model is then used to initialize the static NiTROM optimization problem, where a fixed number of offline optimization steps is taken over the manifold to ensure accurate modeling of the training trajectory. Figure 6 shows the convergence behavior of the offline NiTROM optimization for all three cases considered in this work. In case 1, convergence is achieved within approximately 60 iterations, while case 2 requires 50 iterations and case 3 converges after only 10 steps. These three static NiTROM models are then used as the initial condition for corresponding adaptive ROMs in each case.

3.4 Online/adaptive ROM setup

During the online phase over the testing window $(t_1, t_2]$, we activate the adaptive mechanisms described in Sec. 2.2 and Alg. 1. Between two adaptation events, the ROM propagates forward for Z time steps. Then, the adaptation is triggered and the model interacts with the FOM to obtain a new snapshot before updating its parameters using the data window $\mathcal{D}_j$ of length M. We perform ablation studies by sweeping over the main hyperparameters (M, Z, K) to assess the impact of adaptation frequency, window length, and optimization budget on stability and predictive accuracy. Moreover, as mentioned earlier in Remark 2, the regularization parameter λ used during online OpInf updates is kept fixed and equal to the value identified during the offline training stage.

3.5 Evaluation metrics

The predictive quality of each ROM is assessed using both quantitative and qualitative measures. The primary diagnostic is the perturbation energy,

$$E(t) = \|\mathbf{v}\|_2^2,$$

which directly measures the amplitude of the velocity fluctuations $\mathbf{v}$ around the underlying steady-state velocity profile $\overline{\mathbf{V}}$ shown in Fig. 3a. In addition to energy, we also compute the absolute field error

$$e(t) = \|\mathbf{v}_{\mathrm{FOM}}(t) - \mathbf{v}_{\mathrm{ROM}}(t)\|_2$$

that provides a global measure of accuracy over the spatial domain. To complement these global metrics, we examine the u-velocity profile along a horizontal slice located at $y = 0.05\,L_y$ (here, $L_y = 1$) above the bottom wall, where most of the dominant flow activity occurs. This provides a compact, quantitative representation of the local amplitude and phase of the dynamics, closely matching what is observed in the two-dimensional field plots. Finally, we visually inspect the global vorticity and velocity fields to check whether the phase and amplitude of the predicted flow structures match those of the full-order model.

4 Results and discussion

4.1 Case 1: rich training

In this case, the models are trained over [0, 12.5] and then used to predict the subsequent (12.5, 20] interval (Fig. 4a). The training window contains the entire transient and several oscillatory periods of the flow, meaning that most relevant spatial and temporal structures have already been observed. This makes the online stage relatively simple, where only small amplitude and phase corrections are needed.

4.1.1 Static ROMs

Figure 7 shows the evolution of the energy and the corresponding field error for the static Galerkin, OpInf, and NiTROM models. All three models reproduce the general oscillatory pattern of the FOM but quickly drift away, indicating amplitude overgrowth that would likely lead to instability if the integration were continued. Note that Galerkin projection intrusively derives the reduced operators by projecting the full-order governing equations onto the POD basis, whereas OpInf identifies the reduced dynamics from data via regression. Under certain assumptions on the snapshot data and in the absence of regularization, it has been shown (Peherstorfer 2020) that operator inference recovers the same ROM as intrusive Galerkin projection. In the present work, however, these assumptions are not strictly satisfied and the regression problem is regularized to improve conditioning (see Remark 2). As a result, the operators identified by OpInf differ from the Galerkin-projected operators, which explains the differences observed between the two models despite the shared basis. Within the training window, static NiTROM achieves the best fit to the data, reflecting the benefit of its offline manifold optimization and oblique projection. However, in the prediction window, all static models behave comparably. This shows that the extrapolation accuracy achieved through offline optimization is limited, and an online adaptation mechanism is required to maintain accuracy. Inspection of the vorticity fields in Fig. 10 at the end of

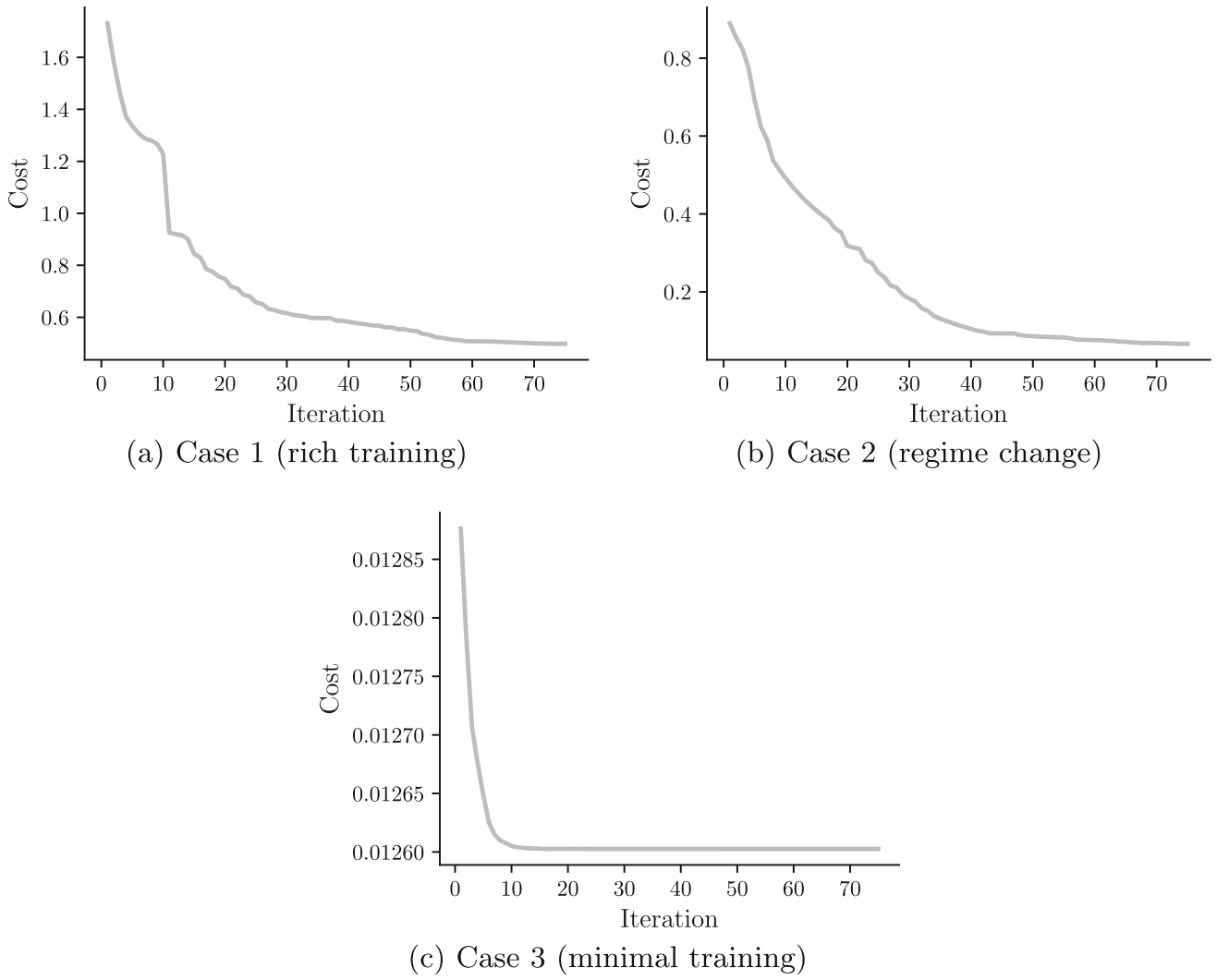


Fig. 6 Convergence histories of the offline NiTROM optimization used to train the static ROM for each case

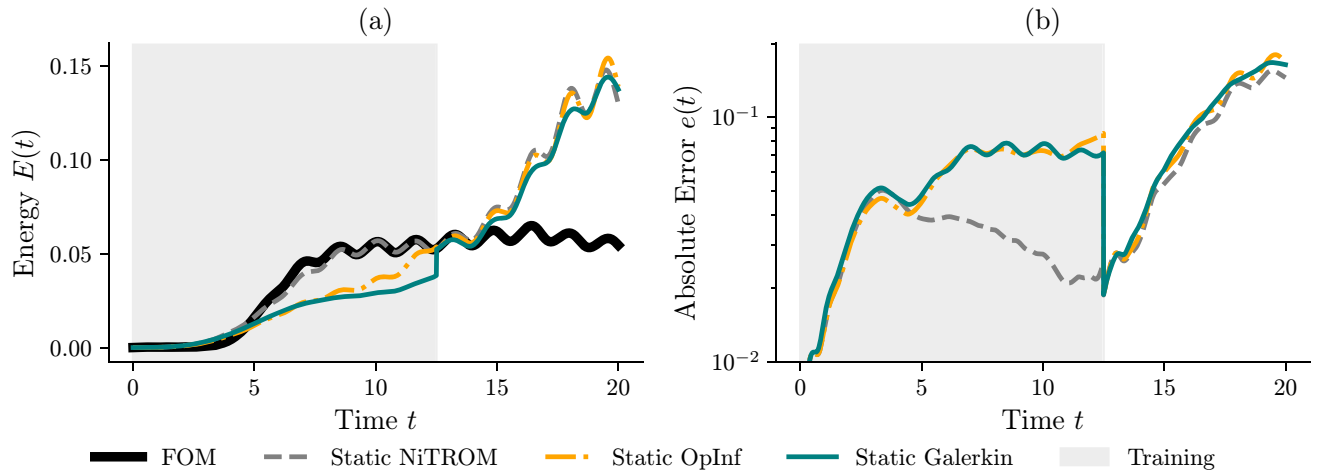


Fig. 7 Energy and field-error evolution for static ROMs in case 1 (rich training). All static models (Galerkin, OpInf, NiTROM) match the general oscillatory pattern but gradually overpredict energy, leading to growing amplitude and loss of stability beyond the training horizon. Note that NiTROM performs much better compared to the other static models inside the training window

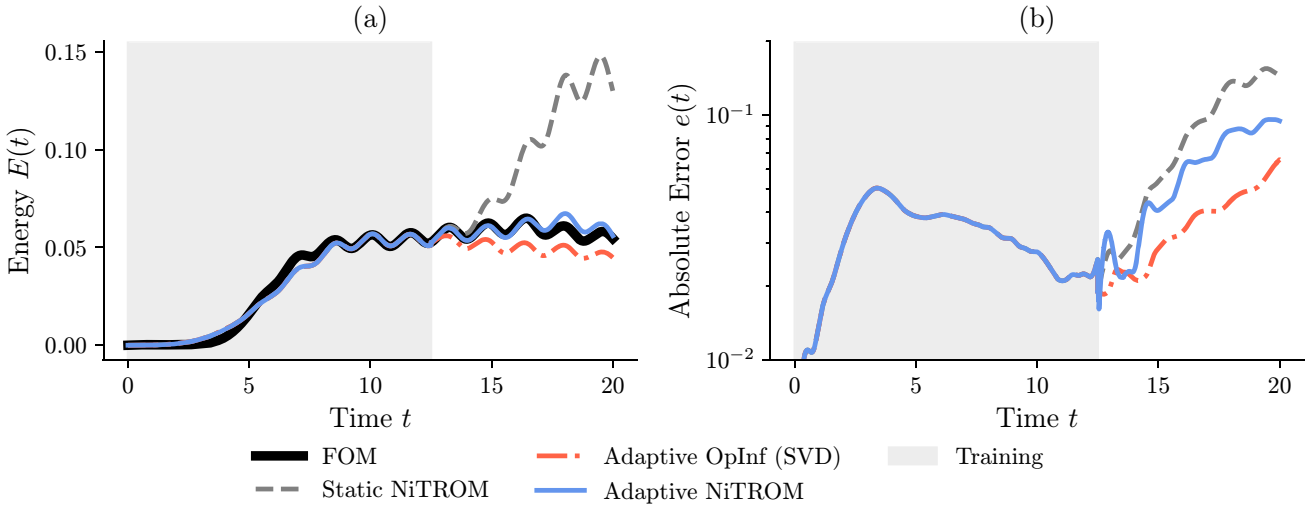


Fig. 8 Energy and field-error evolution for adaptive ROMs ($Z = 10$, $M = 100$) in case 1 (rich training). Both adaptive models suppress the non-physical energy growth of static ROMs. Adaptive OpInf exhibits mild damping, while Adaptive NiTROM (with $K = 10$) nearly matches the FOM energy

the prediction horizon confirms this trend. While the dominant flow structures and phase patterns are captured correctly, the predicted fields exhibit visibly larger amplitudes than the fields from the FOM, and this is consistent with the observed energy overshoot. Appendix A contains (perturbed) velocity component fields u and v for all methods tested in this paper, across all cases.

4.1.2 Adaptive ROMs at frequent updates

Next, we examine the adaptive models with an adaptation window of $Z = 10$ and a lookback window of $M = 100$ snapshots. For Adaptive OpInf, the basis is updated using windowed SVD, and for Adaptive NiTROM, $K = 10$ optimization iterations are performed per update. Looking at the energy and error plots in Fig. 8, both approaches successfully suppress the non-physical energy growth observed in the static models. Adaptive OpInf shows a slight decay in energy, suggesting mild over-damping that could lead to amplitude fading over very long horizons. Adaptive NiTROM, in contrast, almost perfectly matches the FOM energy curve. Velocity slice (Fig. 9) and vorticity field (Fig. 10) comparisons reveal that both adaptive models reproduce the dynamics accurately and maintain amplitude control, while Adaptive NiTROM introduces a small phase lag relative to the FOM. This lag could be mitigated in principle by augmenting NiTROM's loss function with a phase-alignment term.

4.1.3 Adaptive ROMs at less frequent updates

To investigate the effect of less frequent adaptation, we increased the adaptation window to $Z = 50$ and reduced

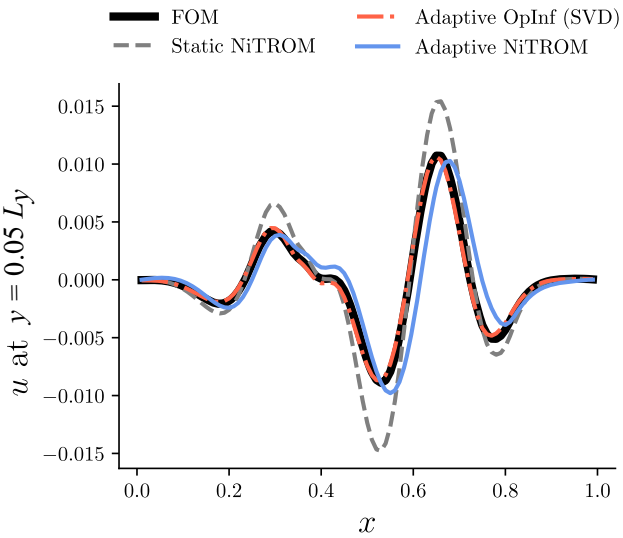


Fig. 9 u-velocity slice for adaptive ROMs ($Z = 10$, $M = 100$) at a horizontal location $y = 0.05\,L_y$ above the bottom wall in case 1 (rich training) at $t = 20$. Both Adaptive OpInf and Adaptive NiTROM accurately reproduce the amplitude of the near-wall velocity, confirming their ability to control energy growth while maintaining physical coherence. Adaptive OpInf also captures the phase correctly, while Adaptive NiTROM causes a small phase shift

the lookback to $M = 20$. The energy and error plots are shown in Fig. 11. In this configuration, Adaptive OpInf continues to control the energy growth, though, just like the previous case, the overall amplitude decreases over time. Adaptive NiTROM, now performing $K = 20$ optimization steps per update, struggles to recover the correct energy trajectory. This is due to its optimization nature and its warm-start initialization from the previously learned parameters. When the system evolves substantially between two adapta-

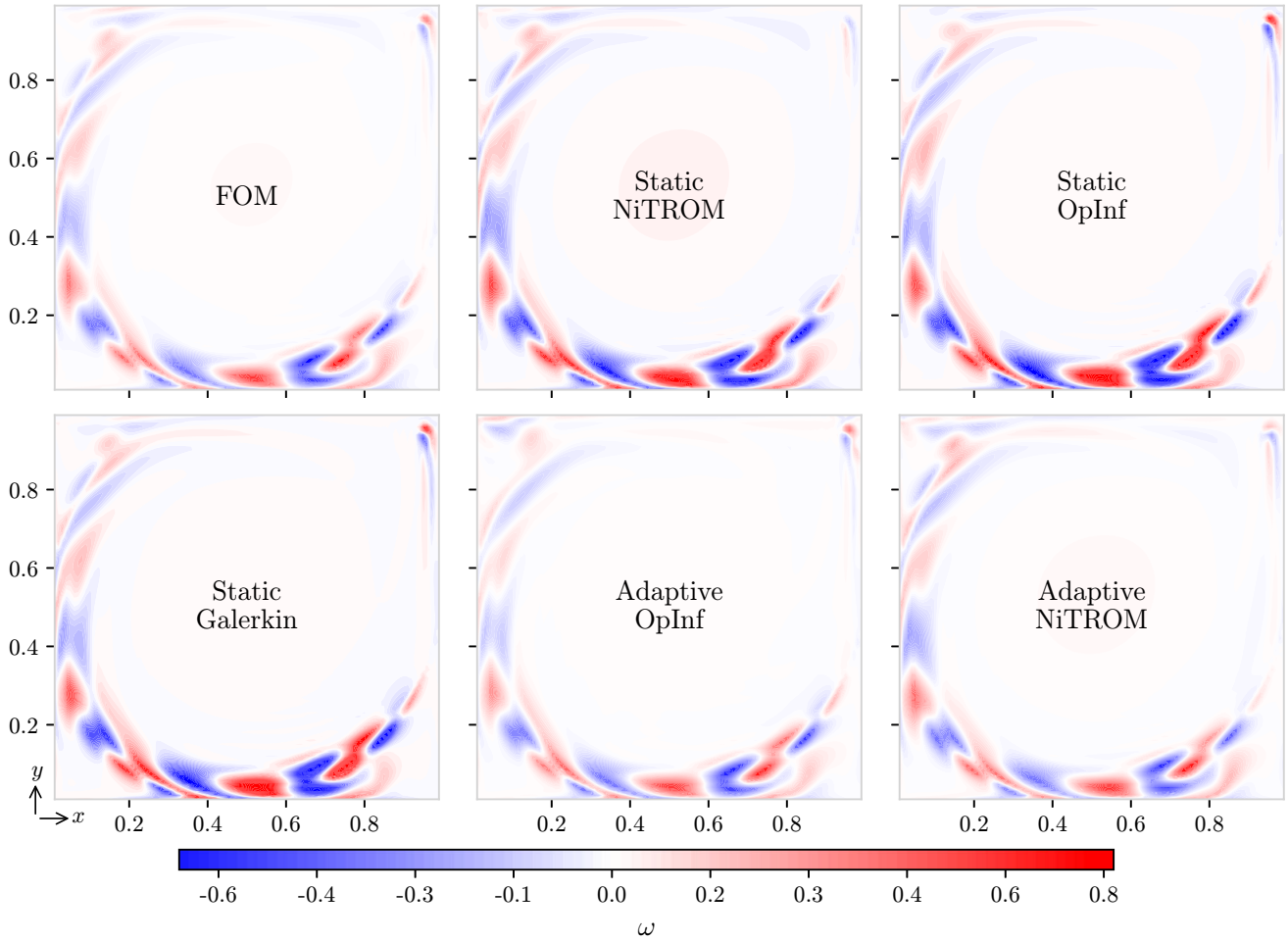


Fig. 10 Ground truth (FOM) and predicted (ROM) vorticity fields at the end of the online window ($t = 20$) in case 1. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 10$, a lookback window $M = 100$, and a per-adaptation optimization budget $K = 10$

tion events, the new optimal parameters may lie far from the previous ones in the optimization landscape, causing the algorithm to converge slowly or fall into local minima. This observation highlights a key limitation of purely optimization-based adaptation. To address this issue, we tested our Adaptive OpInf–NiTROM method that performs a quick OpInf refit followed by $K = 10$ NiTROM optimization steps. As expected, this configuration performs better than either adaptive method alone. The OpInf stage rapidly re-learns the operators based on the current window, effectively providing a good initialization, while the NiTROM refinement fine-tunes the parameters over the manifold. The resulting energy trajectory in Fig. 11 aligns closely with the FOM, confirming that such hybrid adaptation can effectively handle situations where the system changes substantially between updates. Nevertheless, this configuration does not fully showcase the potential of the hybrid method. As evident from the velocity slice in Fig. 12 and qualitative field comparison in Fig. 13, Adaptive OpInf already reproduces the flow structures accurately, leaving little room for improvement through additional manifold optimization steps, despite the slight gains observed in the energy curve. The following cases (2 and 3) introduce more challenging conditions where the hybrid method demonstrates its clear advantage over the individual approaches.

4.1.4 Ablation studies

A detailed ablation study was performed to assess the sensitivity of the adaptive models to their hyperparameters and design choices. The complete set of experiments is presented in Figs. 14–16. The following conclusions summarize the main findings:

- **Effect of adaptation window (Z):** Adaptive OpInf remains robust over a wide range of Z values and maintains stable predictions even with large update intervals (see Fig. 14a). Adaptive NiTROM performs best for

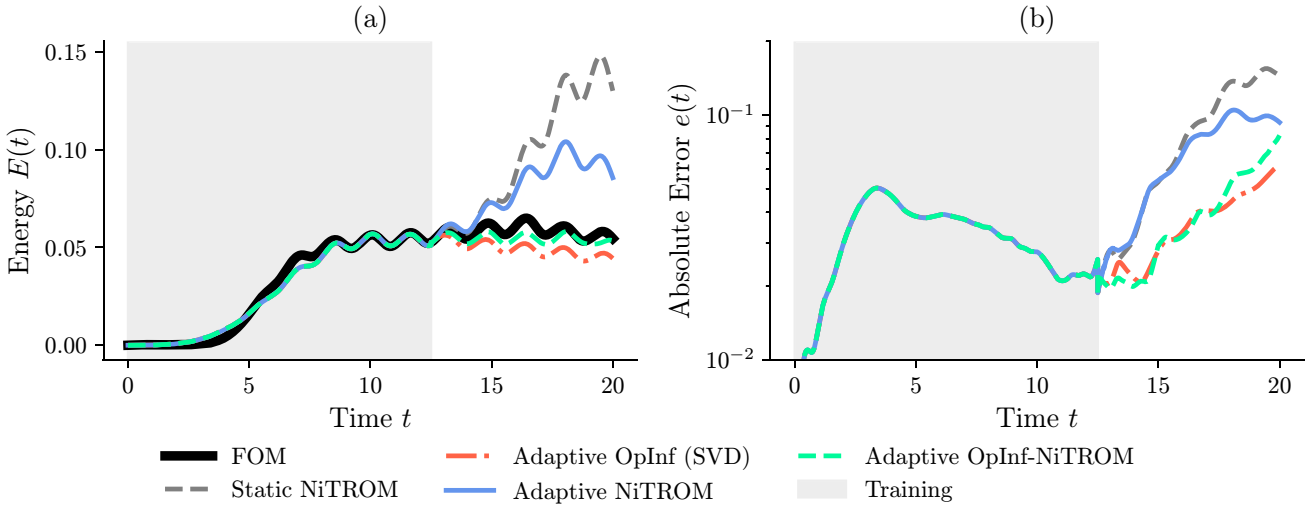


Fig. 11 Energy and field-error evolution for adaptive ROMs ($Z = 50$, $M = 20$) in case 1 (rich training). Adaptive OpInf continues to stabilize the solution but slightly over-damps energy. Adaptive NiTROM (with $K = 20$) struggles due to its warm-start sensitivity, while the Adaptive OpInf–NiTROM (with $K = 10$) restores near-perfect energy tracking

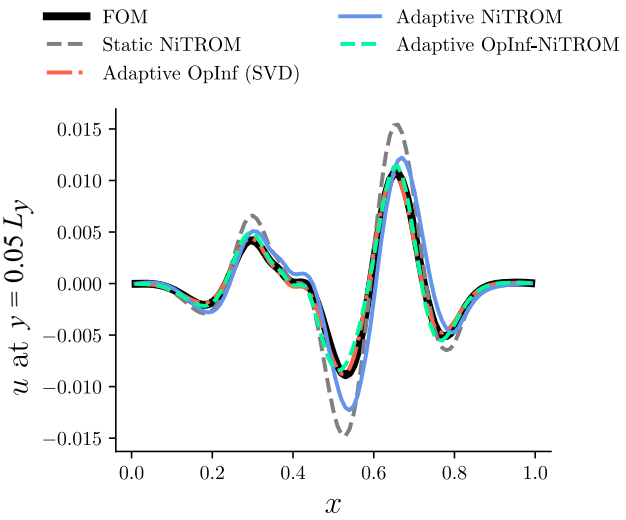


Fig. 12 u-velocity slice for adaptive ROMs ($Z = 50$, $M = 20$) at a horizontal location $y = 0.05\,L_y$ above the bottom wall in case 1 (rich training) at $t = 20$. The hybrid model preserves both phase and amplitude of the oscillations, similar to Adaptive OpInf. Adaptive NiTROM causes a small phase shift and amplitude growth

small Z, where the optimizer starts close to the current minimum, and its accuracy degrades as Z increases (see Fig. 15a). Note that in Figs. 14a and 15a, the window size M is chosen for each value of Z such that all models span the same temporal interval. This ensures that observed differences are not influenced by mismatched lookback horizons; further details are discussed below.

- **Effect of lookback window (M):** The lookback window should be chosen consistently with Z. Once Z is chosen, the quantity $(M-1)Z\Delta t$ effectively determines the temporal span of the system's past history considered during each adaptation (Fig. 2). The appropriate choice of M is therefore problem-dependent and should be informed by the characteristic physical timescales of the underlying dynamics. For example, we see in Figs. 14b and 15b that a small M fails to capture enough temporal variation for effective learning.
- **Basis-update strategy:** In this work, we employ the windowed SVD as the default basis update strategy, as it provides a clear and consistent baseline that does not rely on additional algorithmic choices. The ablation results in Fig. 14c, however, suggest that iSVD may offer a promising direction for future exploration, since it delivers similar performance to the windowed SVD in case 1, while incurring substantially lower computational cost. As discussed earlier, rank-one basis updates were found to be ineffective in this non-intrusive setting, where local intrusive modifications to the basis are not possible.
- **Optimization steps (K) in NiTROM:** Increasing K generally improves adaptation accuracy (as seen in Fig. 16), but overly large values add unnecessary computational burden. In this work, we limited the number of iterations per adaptation as a practical heuristic to maintain responsiveness and avoid overfitting the data within the current window. A more principled solution would involve modifying the online optimization problem itself, for example, through setting a threshold on cost decay or up-weighting recent data, so that the learned parameters remain responsive to imminent regime changes without relying on manually limiting the optimization steps.

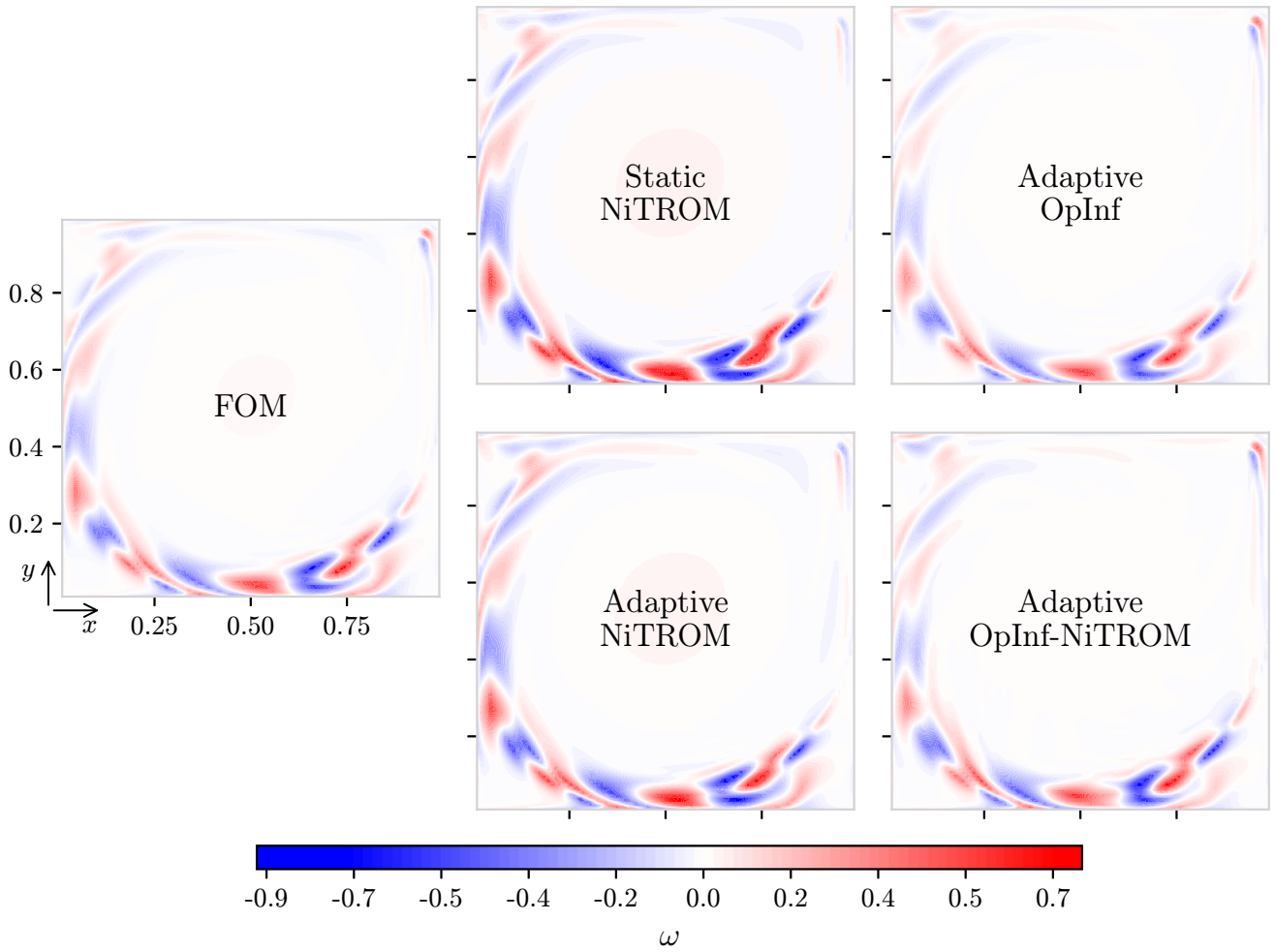


Fig. 13 Ground truth (FOM) and predicted (ROM) vorticity fields at the end of the online window ($t = 20$) in case 1. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 50$ and a lookback window $M = 20$. For Adaptive NiTROM and Adaptive OpInf–NiTROM, per-adaptation optimization budget is set to $K = 20$ and $K = 10$, respectively

4.2 Case 2: regime change

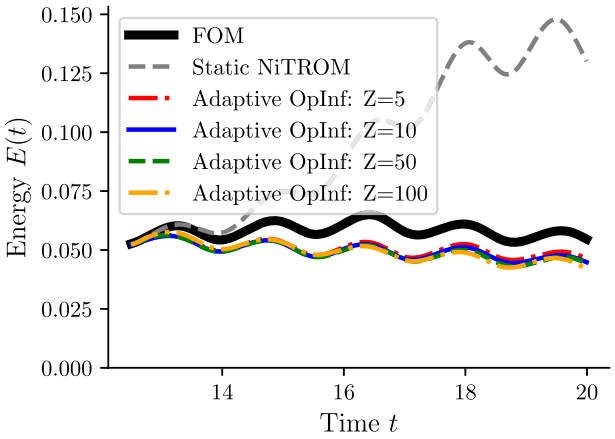
We tested the model in another setting, where the offline window spans $t = [0, 7.5]$ and the online prediction covers the following $t = (7.5, 10]$, corresponding to 1000 extrapolation steps (Fig. 4b). Thus, the models are trained only on the low-energy and transient-growth phases and must extrapolate into the subsequent oscillatory regime. For the adaptive runs we fix the adaptation window to $Z = 10$, the lookback window to $M = 10$, and (for NiTROM-based updates) the optimization budget to $K = 10$ iterations per adaptation. Adaptive OpInf uses windowed SVD for the basis update.

4.2.1 Static ROMs

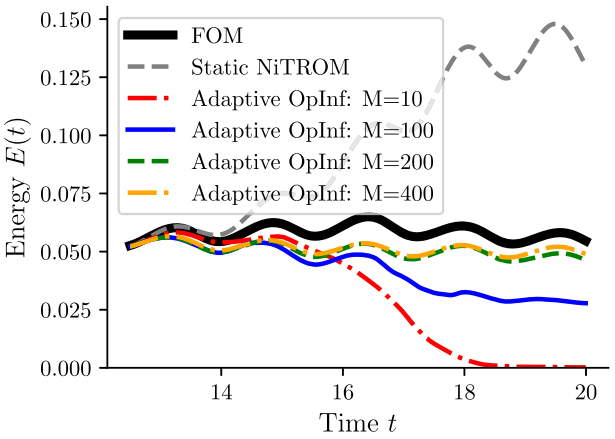
Figure 17 shows that all static ROMs fail to reproduce the new oscillatory regime. Static Galerkin and static OpInf diverge rapidly in energy. Static NiTROM tracks the very first oscillation, but then the energy increases uncontrollably and the solution departs from the physical attractor, similar to the other static models. Looking at the predicted fields at the end of the online phase for Adaptive NiTROM in Fig. 20, we see that a static model is unable to faithfully represent the true dynamics, confirming the non-physical nature of the static predictions once the system leaves the training distribution.

4.2.2 Adaptive ROMs

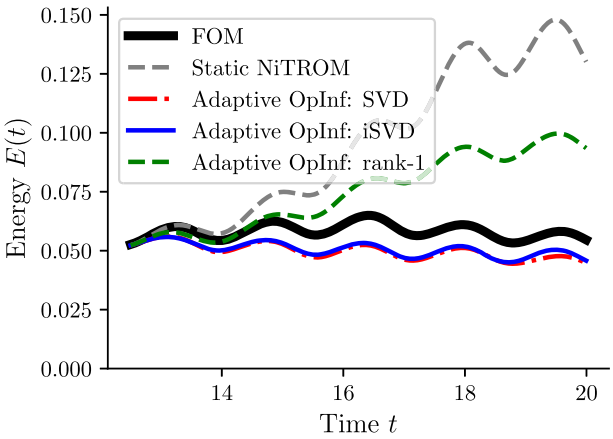
With $Z = 10$ and $M = 10$, the Adaptive OpInf curve in Fig. 18 shows a clear energy drop in the online window, showing that dynamics gradually fade away. While this stabilizes the solution, it underestimates the amplitude of the true oscillations and therefore does not capture the regime change quantitatively. Adaptive NiTROM, using $K = 10$ iterations



(a) Effect of varying Z (with variable M for each Z, to ensure the window spans $1,000$ time steps)



(b) Effect of varying M (with frozen $Z = 10$)



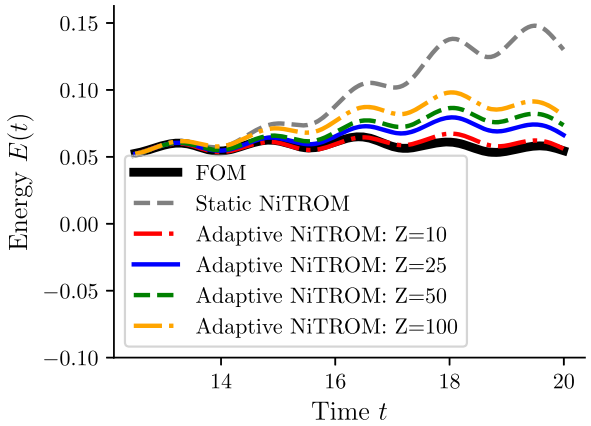
(c) Effect of basis adaptation method (with frozen $Z = 10$ and $M = 100$)

Fig. 14 Ablation studies for Adaptive OpInf applied to case 1. Here, Z and M denote, respectively, adaptation window and lookback window. Shown are performances over the online (prediction) window only, as training is identical for all models

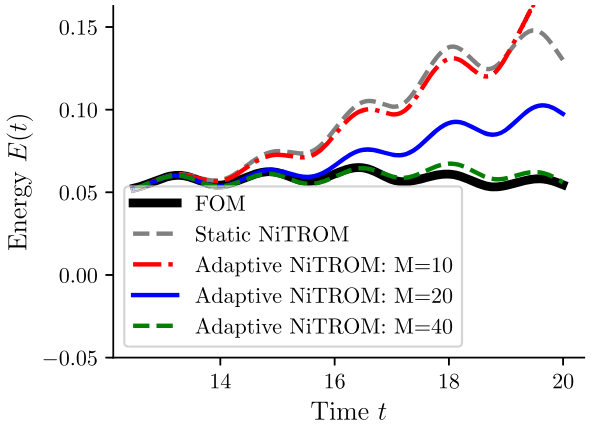
per update, yields essentially no improvement over static NiTROM, as the energy grows similarly after the first cycle. This mirrors the behavior observed in case 1 with sparse updates and is consistent with the mechanism of NiTROM's warm-started manifold optimization. We note that we experimented with several hyperparameter configurations for both Adaptive OpInf and Adaptive NiTROM, including variations in the adaptation frequency, adaptation window size, and number of optimization steps. Across these experiments, both methods consistently failed to capture the evolving dynamics in this regime. These observations indicate that the difficulty in case 2 is not primarily due to hyperparameter choices, but rather reflects a fundamental limitation of these individual adaptive strategies in this extrapolative regime.

In contrast, the Adaptive OpInf–NiTROM configuration retains the overall energy level over the prediction window. Although the hybrid curve does not perfectly match the detailed oscillatory behavior, the total energy is maintained rather than damped (as in Adaptive OpInf) or amplified (as in Adaptive NiTROM).

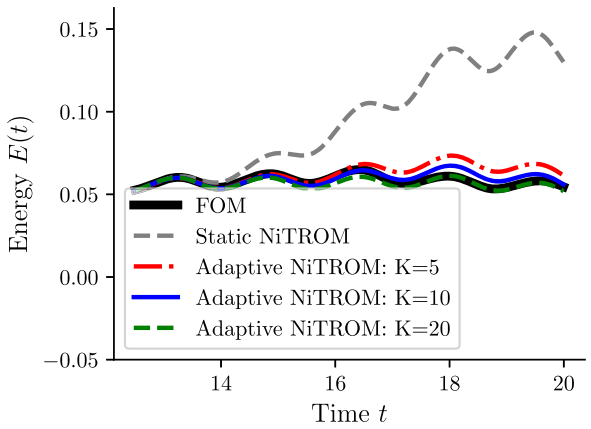
Velocity slices (Fig. 19) and field visualizations (Fig. 20) clarify these differences. Adaptive OpInf, while suppressing energy, introduces noticeable phase error in the oscillatory motion and exhibits small numerical wiggles in portions of the domain. The considerable phase shift is also visible in the u-slice. We also see that Adaptive NiTROM alone adds no visible improvement over the static counterpart, matching our observation from the energy plot.



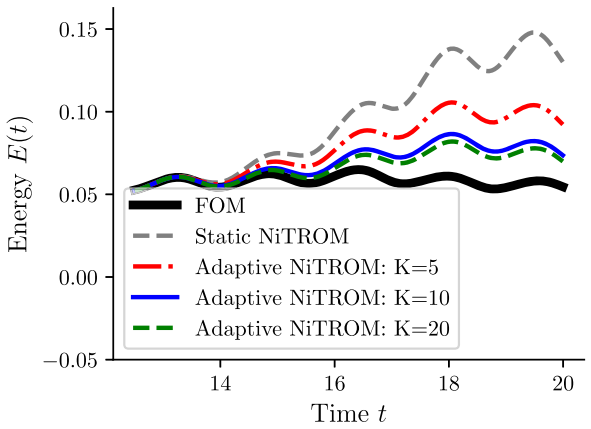
(a) Effect of varying Z (with frozen $K = 10$ and variable M for each Z, to ensure the window spans $1,000$ time steps)



(b) Effect of varying M (with frozen $Z = 10$ and $K = 10$). Note that Adaptive NiTROM converges to the ground truth with significantly smaller window sizes compared to Adaptive OpInf (see Fig. 14b).



(c) Effect of varying K (with frozen $Z = 10$ and $M = 40$)



(d) Effect of varying K (with frozen $Z = 50$ and $M = 10$)

Fig. 15 Ablation studies for Adaptive NiTROM applied to case 1. Here, Z, M, and K denote, respectively, adaptation window, lookback window, and per-adaptation manifold optimization steps. Shown are performances over the online (prediction) window only, as training is identical for all models

The hybrid method yields the most compelling fields. We observe clean vortical structures with accurate placement and strength, and the u-slice shows good amplitude control and phase alignment relative to the FOM. Importantly, the small oscillations present in Adaptive OpInf predictions are substantially reduced after the brief NiTROM refinement, though faint remnants persist in the same regions. This is a representative case where purely numerical metrics can be misleading. Despite the hybrid energy curve not perfectly tracing the oscillatory motion, the qualitative field assessment shows that the hybrid model reconstructs the regime change most faithfully.

4.3 Case 3: minimal training

In this most challenging configuration, the models are trained on $t = [0, 3.75]$ and then tested on $t = (3.75, 6.25]$ (Fig. 4c). The training data correspond to the initial low-energy regime of the flow, containing almost no dynamic evolution. Immediately after the training window, the FOM exhibits a sharp energy growth as the flow transitions into the transient phase. This setting therefore pushes the limits of any ROM methodology, where the basis and operators have seen almost no variability, and the subsequent dynamics evolve rapidly in both amplitude and spatial structure. Adaptive parameters are fixed to $Z = 10$, $M = 10$, and $K = 10$. The smaller look-

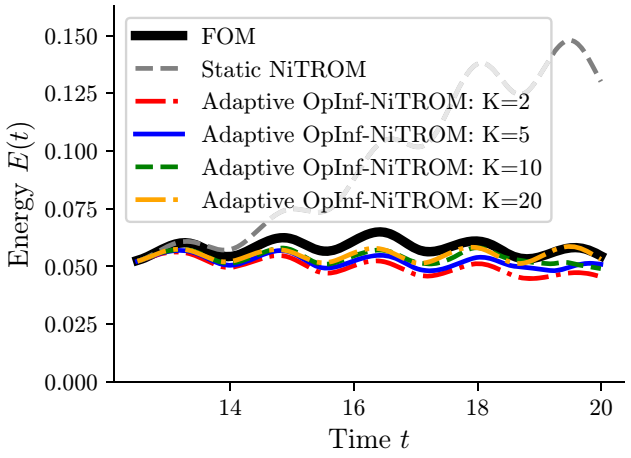


Fig. 16 Effect of number of per-adaptation manifold optimization steps K for Adaptive (hybrid) OpInf–NiTROM applied to Case 1, with frozen $Z = 50$ and $M = 20$. Shown are performances over the online (prediction) window only, as training is identical for all models

back window ensures that the model relies only on the most recent information rather than outdated, non-representative states.

4.3.1 Static ROMs

Unlike the training data, the static ROMs do not remain low-energy during prediction. Figure 21 shows that the energy begins to grow almost immediately in the online phase and diverges from the FOM toward the end of the prediction window. Inspection of static NiTROM's flow fields at the end of the online window (in Fig. 24) reveals that the predicted dynamics remain localized along the right wall, where the basis was constructed from the limited low-energy training data. Because the basis never observed the broader transient evolution, the predicted vortices fail to propagate into the domain interior.

4.3.2 Adaptive ROMs

The adaptive models are also challenged by the rapid changes in this case. As shown in Fig. 22, Adaptive OpInf produces a noticeable decay in energy, similar to its behavior in case 2. The model remains stable but underestimates the energy amplitude, as the limited history in the window constrains the temporal context used for operator inference. Adaptive NiTROM, despite its more expressive optimization, again shows no significant improvement over the static NiTROM. Its energy curve follows a similar growth pattern and fails to reproduce the transient dynamics. As in case 2, we tested multiple hyperparameter configurations for both Adaptive OpInf and Adaptive NiTROM, and across these experiments, both methods consistently failed. These results highlight the limitations of these individual adaptive approaches in this regime and motivate the hybrid formulation.

In contrast, the Adaptive OpInf–NiTROM model again achieves the most stable and physically meaningful results. The energy level remains bounded and consistent with the FOM, neither decaying to zero nor diverging.

Now we investigate velocity slice comparison in Fig. 23 and field visualizations in Fig. 24 to confirm these quantitative trends. By close inspection of Adaptive OpInf's predicted fields, we can notice that it captures only localized amplification near the right wall and fails to reproduce interior vortex propagation. Adaptive NiTROM, on the other hand, shows no noticeable improvement over its static version, consistent with the trends seen in the energy plot. The Adaptive OpInf–NiTROM approach, however, displays clear vortex formation and spatial patterns that closely follow the FOM. The brief NiTROM refinement once again "cleans up" the

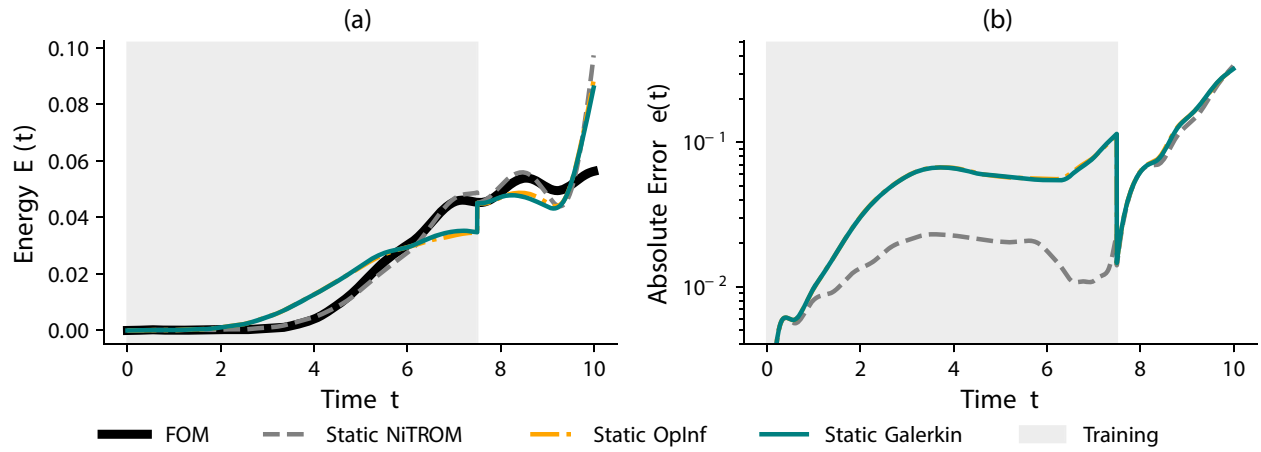


Fig. 17 Energy and field-error evolution for static ROMs in case 2 (regime change). All static models fail to reproduce the regime change. While static NiTROM captures the first oscillation, it soon diverges, highlighting the inability of static ROMs to generalize to unseen dynamical regimes

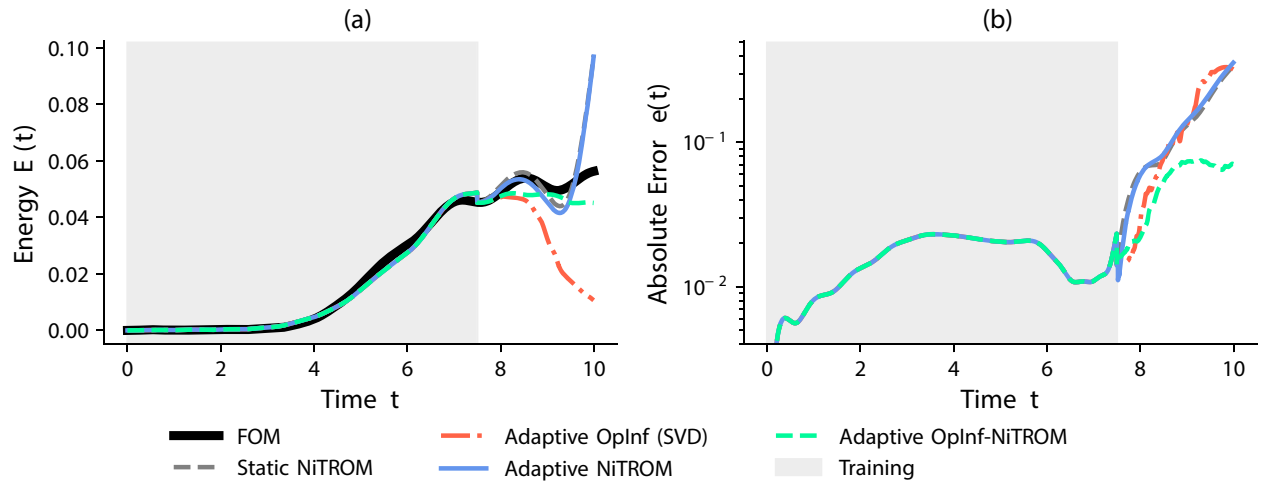


Fig. 18 Energy and field-error evolution for adaptive ROMs ($Z = 10$, $M = 10$) in case 2 (regime change). Adaptive OpInf stabilizes the trajectory but underpredicts the oscillation amplitude. Adaptive NiTROM (with $K = 10$) adds no improvement over static NiTROM, whereas the Adaptive OpInf–NiTROM (with $K = 10$) approach maintains a bounded, physically consistent energy evolution

small oscillatory artifacts produced by the OpInf stage, yielding smoother and more physically coherent fields. The u-slice captures both the amplitude growth and the approximate phase of the transient oscillation, showing that the hybrid method can track the dynamics even when the offline model has seen virtually nothing.

4.4 Key observations and takeaways

The three cases collectively provide a comprehensive picture of how different adaptive non-intrusive formulations behave under increasing levels of difficulty, from minor corrections in a well-trained model to full extrapolation from nearly featureless data. Several consistent trends and insights emerge from these experiments.

i. Static ROMs are limited by training coverage. Across all cases, static Galerkin, OpInf, and NiTROM models perform well only when the training window already contains the dominant flow dynamics. Outside that regime, they quickly lose physical consistency. This limitation underscores the need for online correction mechanisms once the system evolves beyond the distribution represented in the offline data.

ii. Adaptive OpInf is robust and efficient. In cases with rich training data (case 1), Adaptive OpInf stabilizes the predictions and prevents long-term energy drift. In more challenging settings (cases 2 and 3), it is capable of tracking the qualitative regime change, provided its hyperparameters are carefully tuned. Its main strengths are simplicity, low computational cost, and a remarkable tolerance to large adaptation intervals Z. However, its performance is highly sensitive to the choice of window size M, and it occasionally introduces localized numerical artifacts in the fields.

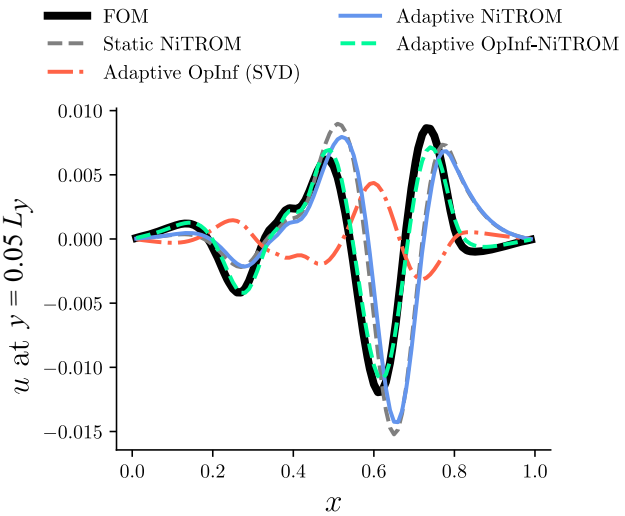


Fig. 19 u-velocity slice for adaptive ROMs ($Z = 10$, $M = 10$) at a horizontal location $y = 0.05\,L_y$ above the bottom wall in case 2 (regime change) at $t = 10$. Adaptive OpInf fails to capture the true profile, Adaptive NiTROM exhibits minimal corrections compared to its static counterpart, while the hybrid method aligns amplitude and phase with the FOM

iii. Adaptive NiTROM is limited by optimization sensitivity and cost. The joint manifold optimization in Adaptive NiTROM enables highly accurate corrections when updates are frequent and the optimization is well converged (small Z, adequate K). In such cases, the energy curve can match the full-order reference almost perfectly. However, as the adaptation interval grows, the method may fail to recover from accumulated error. In out-of-distribution cases (cases 2 and 3), the optimization landscape becomes too complex, and the method provides little benefit over static models. This sensi-

tivity highlights the difficulty of performing online trajectory optimization in dynamical systems that change rapidly. In addition, manifold optimization introduces a significant computational overhead relative to Adaptive OpInf, which limits the suitability of the method for online adaptation. Table 2 reports the average wall-clock time for a single execution of each operation.

iv. Adaptive OpInf–NiTROM combines the strengths of both. The Adaptive (hybrid) OpInf–NiTROM approach offers a practical balance between robustness and accuracy. The initial OpInf update re-learns the operators from the most recent data, effectively reinitializing the model for the new regime. The subsequent NiTROM refinement then enforces geometric consistency and removes the small numerical artifacts seen in OpInf-only predictions. Across cases 2 and 3, the hybrid configuration consistently produces the most physically meaningful and visually coherent fields. This observation suggests that combining fast regression-based updates with short manifold refinements is an effective strategy for maintaining accuracy in challenging settings, although additional work is required to make the manifold optimization more efficient.

v. Convergence behavior of online NiTROM updates. For NiTROM-based adaptive ROMs, convergence analysis differs fundamentally from the offline setting, as the optimization is re-initialized and solved repeatedly at each adaptation event during online prediction. To assess the behavior of the online optimization, we examine the convergence histories of the hybrid OpInf–NiTROM updates across all adaptation events. Figure 25 shows the normalized objective value over optimization iterations for each adaptation event in cases 1–3. While individual events exhibit different convergence rates, the objective consistently decreases within a small number of iterations across all cases. This indicates that a fixed, modest number of optimization steps is sufficient to obtain effective online updates, without requiring event-specific convergence tuning.

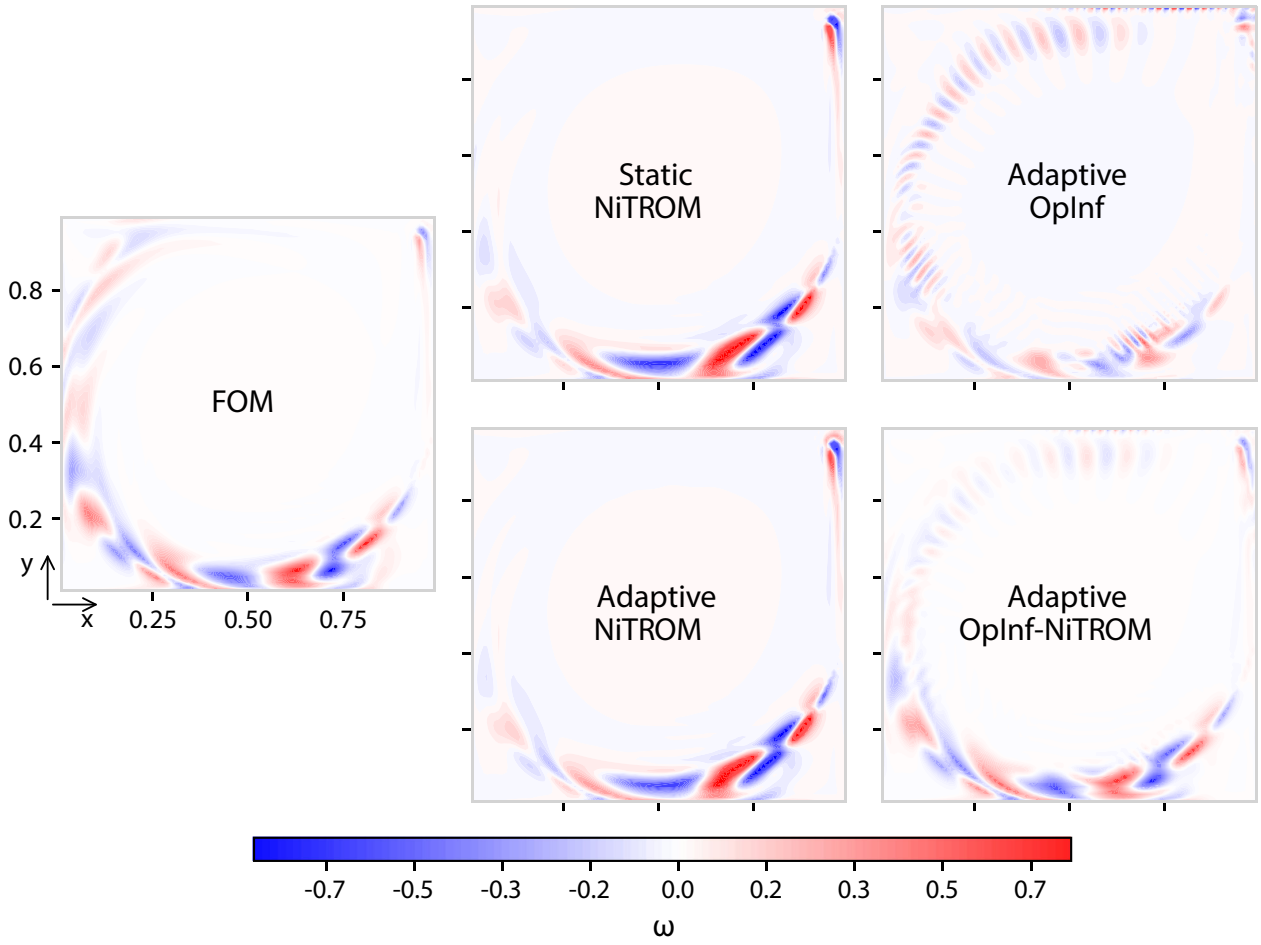


Fig. 20 Ground truth (FOM) and predicted (ROM) vorticity fields at the end of the online window ($t = 10$) in case 2. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 10$, a lookback window $M = 10$, and a per-adaptation optimization budget $K = 10$

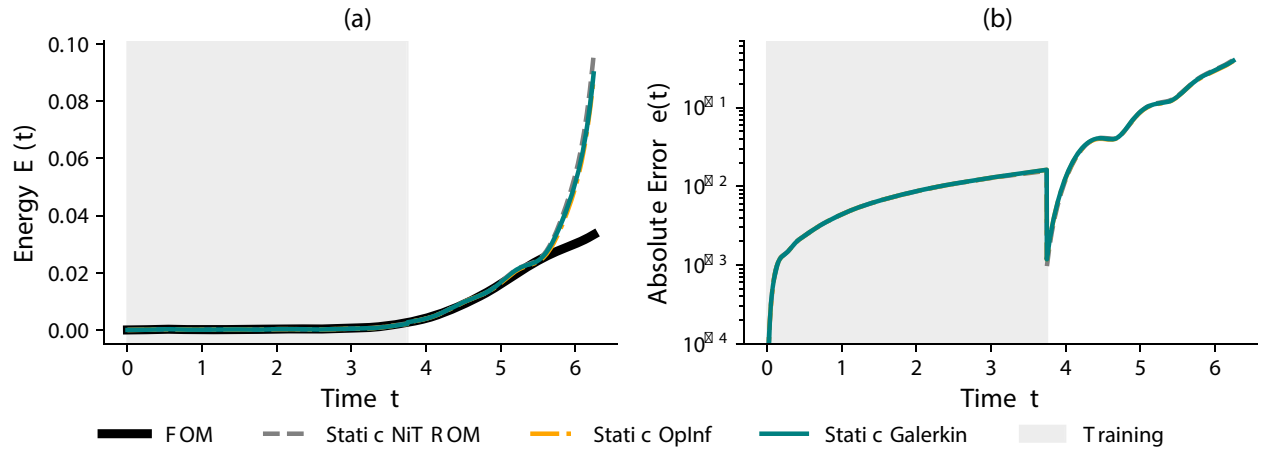


Fig. 21 Energy and field-error evolution for static ROMs in case 3 (minimal training). Despite low-energy training data, static ROMs exhibit artificial energy growth

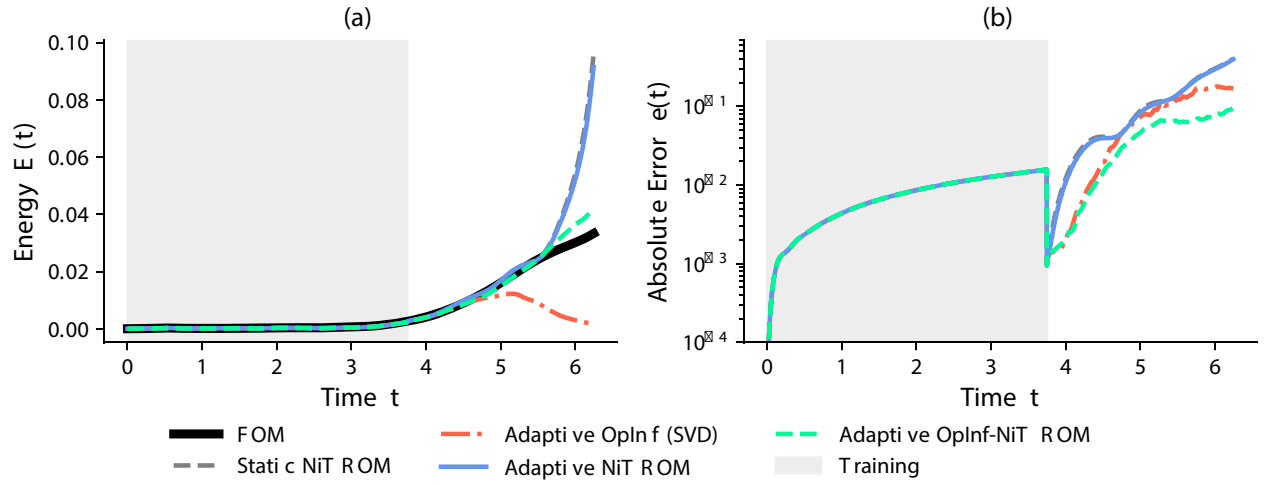


Fig. 22 Energy and field-error evolution for adaptive ROMs ($Z = 10$, $M = 10$) in case 3 (minimal training). Adaptive OpInf stabilizes but underestimates energy; Adaptive NiTROM (with $K = 10$) behaves similarly to its static version. Adaptive OpInf–NiTROM (with $K = 10$) maintains bounded energy and correctly follows the transient-growth trend

vi. Projection error analysis. To further investigate the relationship between basis quality and prediction accuracy, we report the projection error for all models across the three cases in Fig. 26. The projection error measures the distance between the high-fidelity state and its projection onto the instantaneous reduced subspace, and is defined as

$$e_{\text{proj}}(t) = \left\| \mathbf{v}_{\text{FOM}}(t) - \boldsymbol{\Phi}(t)\left(\boldsymbol{\Psi}(t)^{\top}\boldsymbol{\Phi}(t)\right)^{-1} \boldsymbol{\Psi}(t)^{\top}\mathbf{v}_{\text{FOM}}(t) \right\|_2 . \tag{13}$$

Note that the time dependence of $\boldsymbol{\Phi}(t)$ and $\boldsymbol{\Psi}(t)$ here reflects the fact that the bases are adapted online during the simulation. In cases 1 and 2, Adaptive OpInf and Adaptive OpInf–NiTROM that rebuild the basis from scratch exhibit an immediate drop in projection error after the training window. In contrast, Adaptive NiTROM evolves the basis more gradually and therefore remains closer to the static projection error. As the system evolves further away from the training regime, the projection error of Adaptive OpInf and Adaptive OpInf–NiTROM increases and can eventually exceed that of the static basis, even though the corresponding ROM predictions remain more physically accurate according to the energy evolution and flow-field visualizations. In case 3, the hybrid Adaptive OpInf–NiTROM model maintains a lower projection error than the static basis over the prediction interval, which is consistent with its improved predictive performance in this case. Overall, these results highlight an important limitation of projection error in the extrapolative setting considered here. While projection error quantifies the representability of the state within a given subspace, the predictive accuracy of an adaptive ROM also depends on the quality

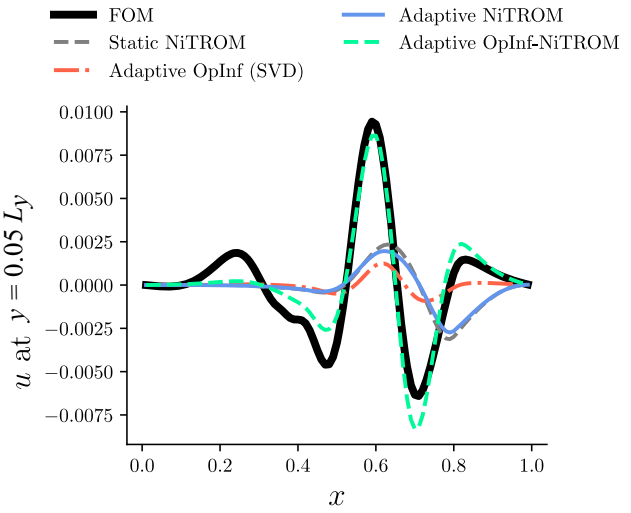


Fig. 23 u-velocity slice for adaptive ROMs ($Z = 10$, $M = 10$) at a horizontal location $y = 0.05\,L_y$ above the bottom wall in case 3 (minimal training) at $t = 6.25$. The hybrid method captures both amplitude and phase evolution, reflecting extrapolation capabilities of Adaptive OpInf–NiTROM

of the inferred latent dynamics and the interaction between basis/operator updates and finite adaptation windows. Consequently, projection error alone does not fully characterize forecasting performance in this setting, and qualitative flow-field agreement remains the most reliable indicator of model accuracy.

vii. Sensitivity of static ROMs to basis dimension. To assess whether the observed limitations of static reduced-order models could be alleviated by simply increasing the reduced dimension, we performed a sensitivity analysis of the static NiTROM model with respect to the basis size. In particular, we considered three values of the reduced dimension, $r = 5$, $r = 10$, and $r = 20$, and evaluated the resulting static models across all three cases. Figure 27 shows the corresponding energy evolution, while Fig. 28 reports the vorticity fields at the end of the prediction window.

As discussed before, in case 1, the static model with $r = 10$ already provides a reasonably accurate field prediction, although it exhibits energy growth after the training

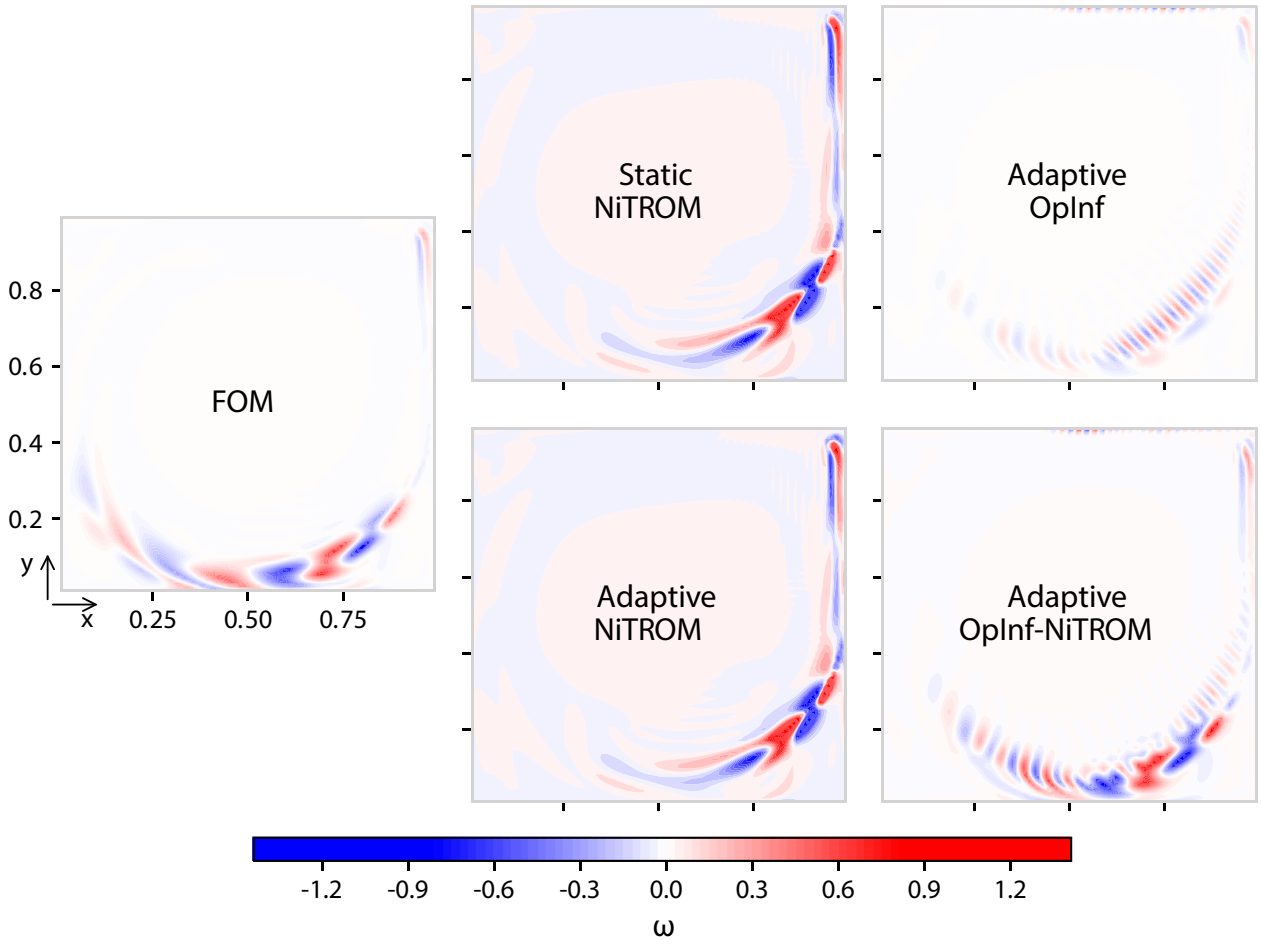


Fig. 24 Ground truth (FOM) and predicted (ROM) vorticity fields at the end of the online window ($t = 6.25$) in case 3. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 10$, a lookback window $M = 10$, and a per-adaptation optimization budget $K = 10$

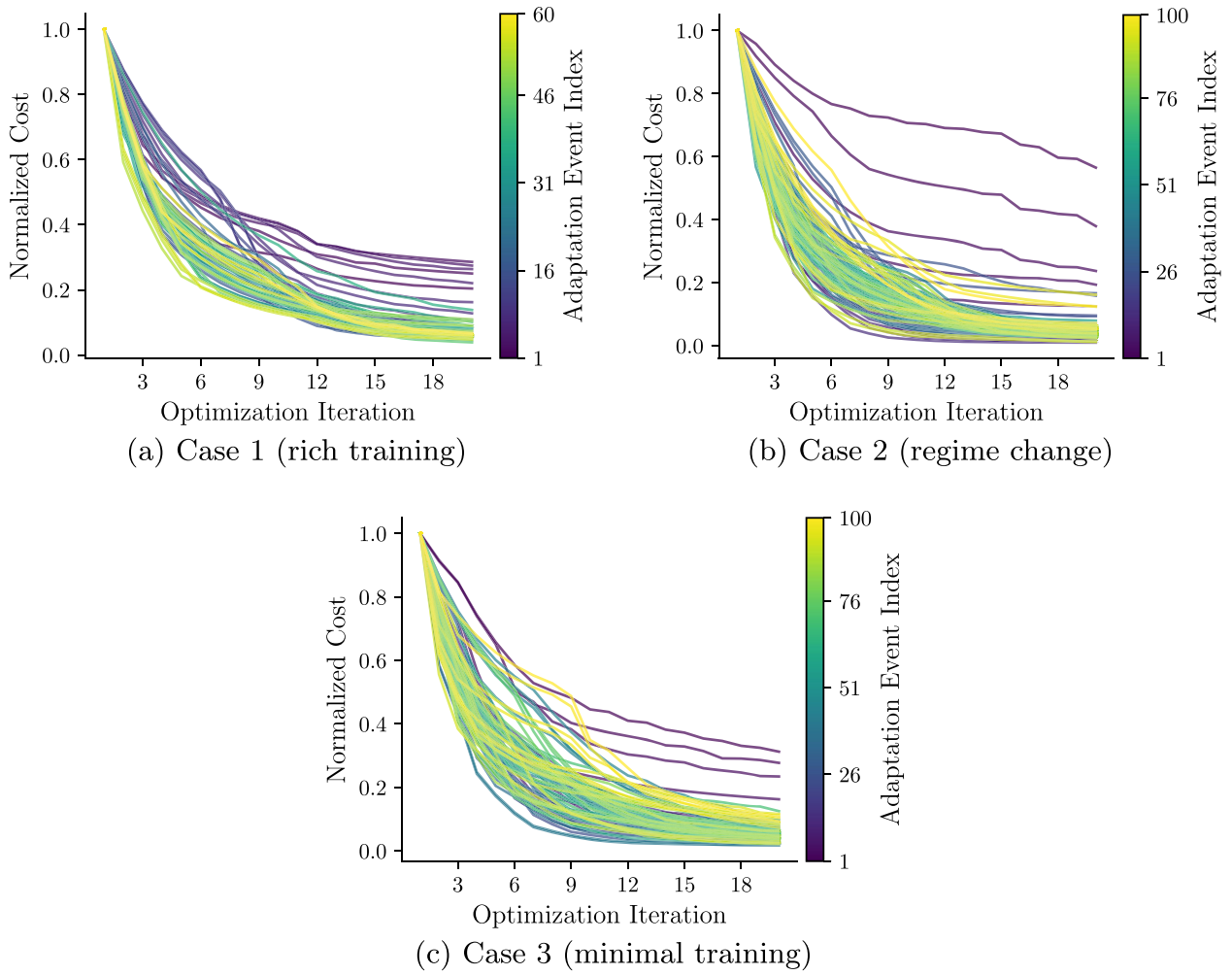


Fig. 25 Convergence histories of the online NiTROM optimization during adaptive updates for the hybrid OpInf–NiTROM model. Each curve corresponds to a single adaptation event and shows the normalized objective value over optimization iterations

phase. Increasing the basis dimension to $r = 20$ improves the energy evolution and brings it closer to the ground truth. However, inspection of the corresponding vorticity fields reveals that the larger basis does not necessarily improve the structural accuracy of the predicted flow. In particular, the $r = 20$ model produces flow patterns that deviate more visibly from the full-order solution compared to the $r = 10$ model, suggesting that increasing the reduced dimension alone does not guarantee improved qualitative agreement.

In case 2, the behavior is largely insensitive to the choice of r. All static models exhibit rapid energy growth immediately after the training window, indicating that the reduced dynamics become unstable when extrapolating beyond the training regime. Examination of the vorticity fields, particularly in the top-right region of the domain where perturbations are injected, further confirms this behavior: the predicted structures appear amplified compared to the full-order solution, suggesting the onset of instability in the reduced dynamics. Looking at the top-right region of the domain where perturbations are injected, we see that the $r = 10$ model provides a slightly closer representation of the perturbation pattern, while both smaller and larger bases fail to capture the correct phase.

In case 3, increasing the reduced dimension to $r = 20$ improves the global energy evolution and prevents the energy growth observed in the smaller models. Nevertheless, the corresponding vorticity fields reveal discrepancies with the full-order solution. In particular, the predicted flow is dominated by structures localized in the region where most of the training data were collected (near the right wall), while the evolving dynamics elsewhere in the domain are not captured accurately. This behavior illustrates a common limitation of purely offline-trained models: increasing the basis dimension can improve representability of the training manifold, but it does not necessarily improve predictive capability when the system evolves into previously unseen regimes.

Overall, these results suggest that while increasing the reduced dimension can sometimes improve accuracy, it does

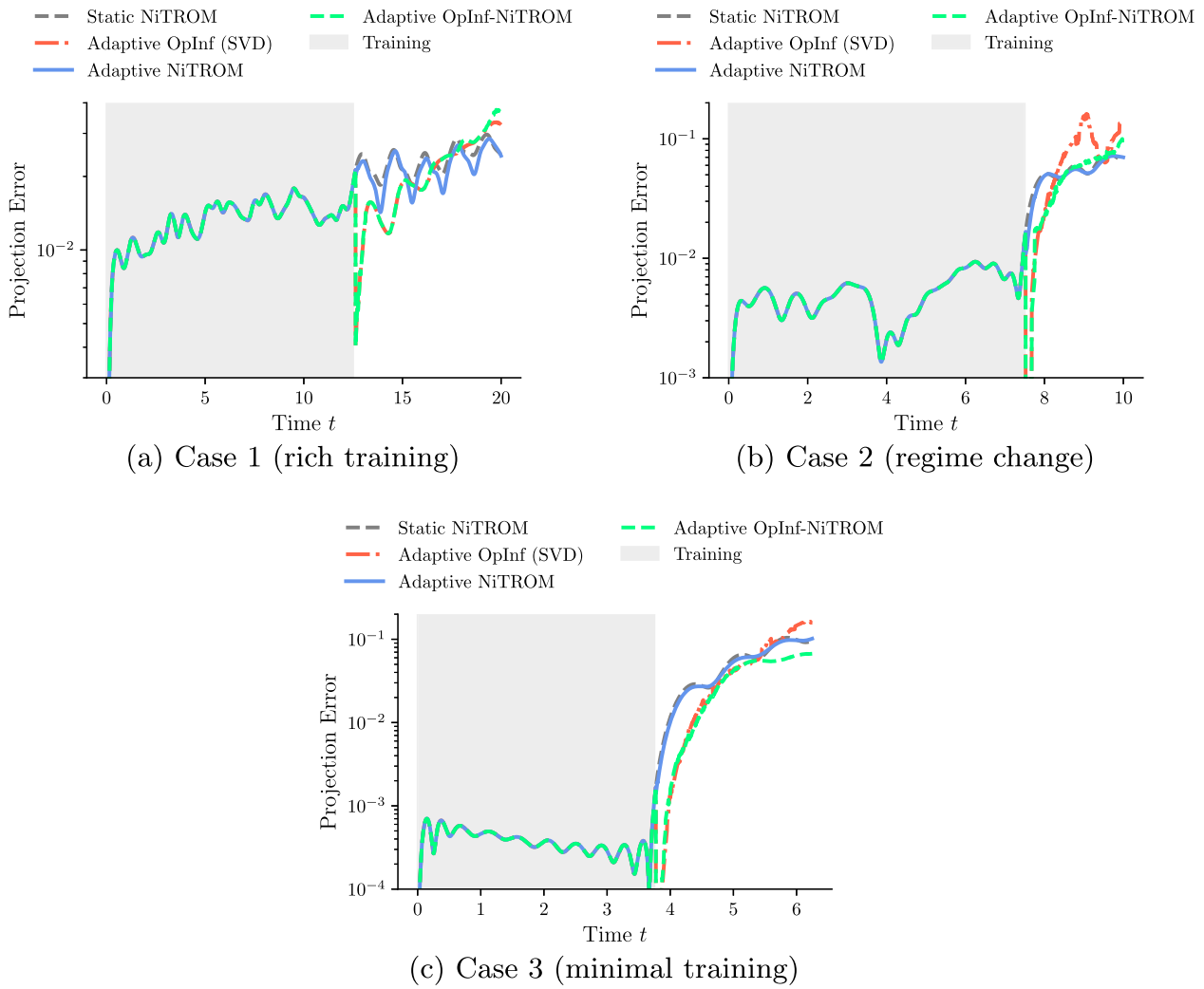


Fig. 26 Projection error over time for case 1 (upper left), case 2 (upper right), and case 3 (lower center), corresponding to the ROMs depicted in Figs. 11, 18, and 22, respectively

not consistently resolve the extrapolation challenges encountered in the present setting. Importantly, the goal of this work is not to argue against basis enrichment as a remedy for improving reduced-order models. In many applications, increasing the reduced dimension can indeed be an effective and computationally inexpensive strategy, and we fully encourage its use when it leads to improved predictions. Instead, the focus of the present work is to explore a complementary approach for situations where offline-trained models, regardless of basis dimension, are incapable of extrapolating beyond the training manifold. In such regimes, simply increasing the reduced dimension may improve representation of the training data, but does not address the underlying mismatch between the learned reduced dynamics and the evolving system behavior. Consequently, basis enrichment and online adaptation should be viewed as complementary tools: basis enrichment improves representability of the training manifold, while online adaptation enables the model to remain predictive when the dynamics evolve outside that manifold.

viii. Broader implications. Overall, the results demonstrate that online adaptation can extend the predictive range of non-intrusive ROMs relative to static models in the tested cases. When the system remains within or near the original subspace (case 1), adaptive updates fully correct amplitude drifts. When the dynamics evolve beyond that subspace (cases 2 and 3), adaptation still provides partial recovery, especially with hybrid updates. Table 3 summarizes the qualitative performance trends observed across all three cases.

ix. A note on wall-clock time in Table 2. Currently, these methods have a large overhead as we use unmodified offline versions of the SVD, OpInf and NiTROM constructions. There exist many opportunities to accelerate ROMs by a) updating algorithms to online, incremental contexts; and b) taking advantage of the fact that ROMs allow for larger time steps than FOMs. We also remark that the cavity flow

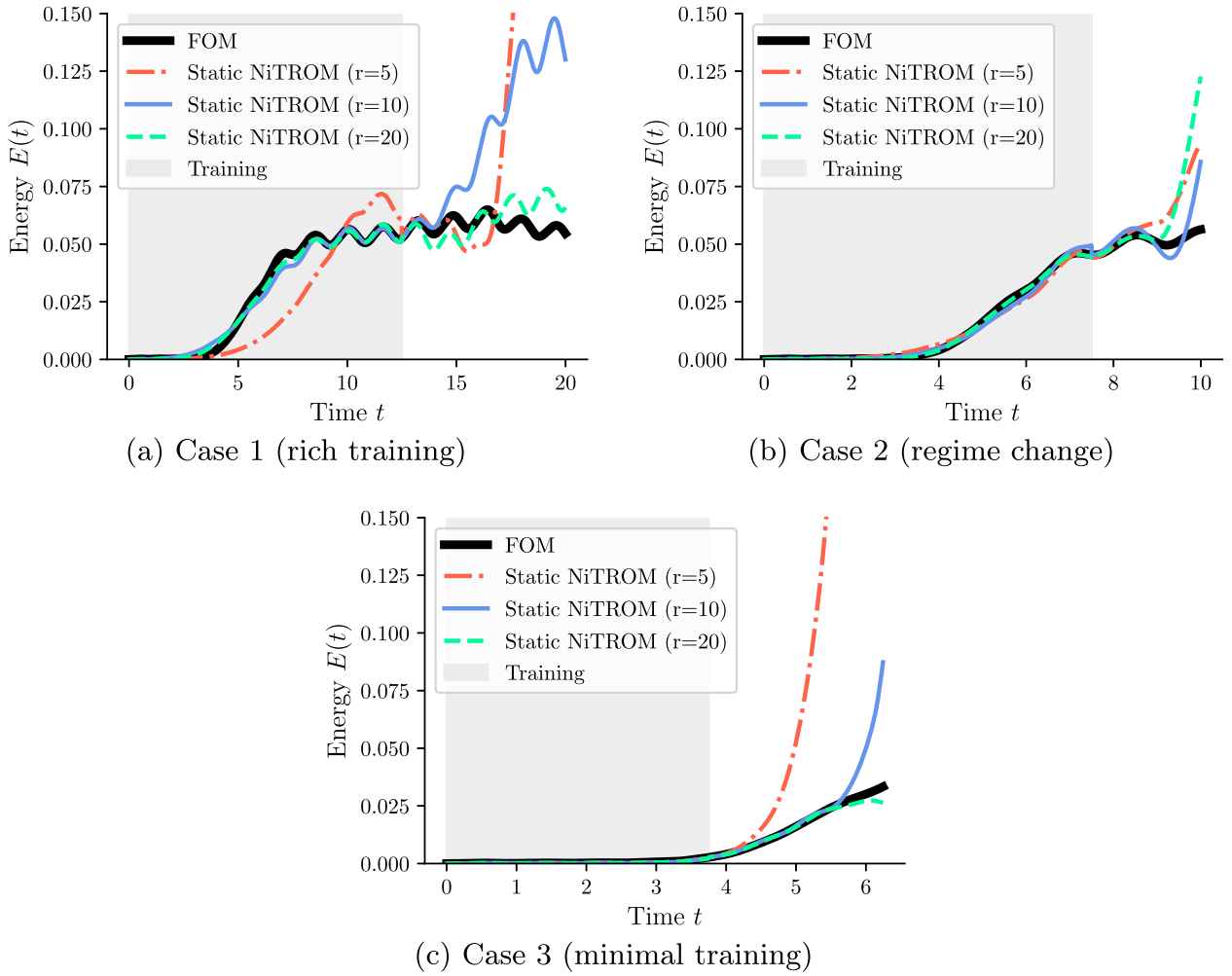


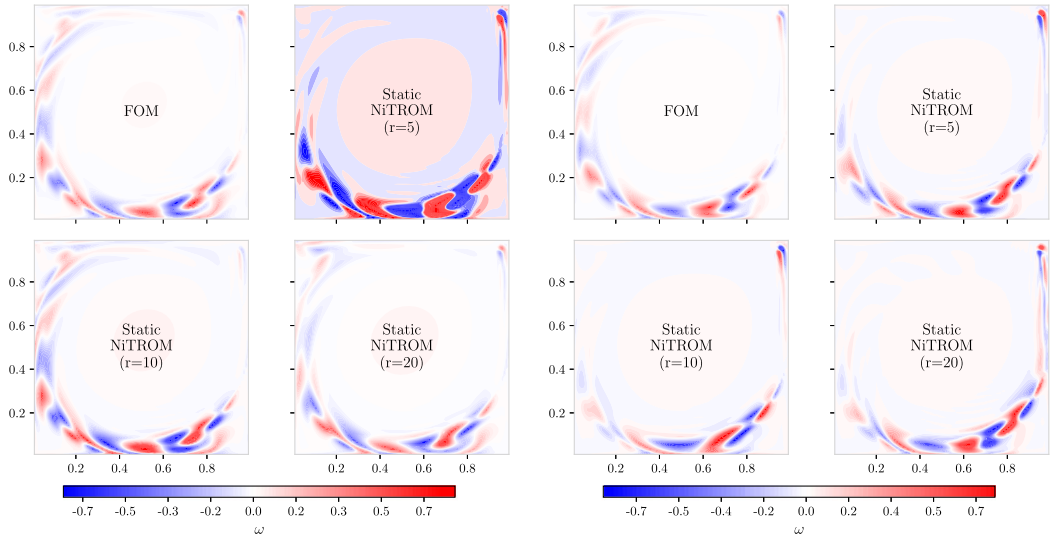
Fig. 27 Sensitivity of static NiTROM models to the reduced dimension r. Energy evolution for cases 1-3 comparing static models with $r = 5$, $r = 10$, and $r = 20$ against the full-order model are shown

under consideration is not meant to particularly highlight computational savings. In fact, the small stencil width of the spatial discretization, the explicit time stepping, the relatively small 100×100 grid, and Python vectorization, make the FOM solver extremely fast (as demonstrated in the table). More complicated problems requiring implicit time stepping, higher per-time-step cost (e.g., compressible flow solvers with shock-capturing logic), and much finer grids, would be better suited to demonstrate cost reduction. As mentioned previously, however, a major objective of this work is to quantify challenges in achieving predictive model reduction. This particular problem was chosen because the FOM solver was very efficient (allowing us to run it hundreds of times), and because *even* this flow configuration exhibits challenging dynamical features for ROMs, and allow us to identify promising research directions to ultimately enable the deployment of adaptive ROMs on transients- and advection-dominated flow problems.

The table is nonetheless useful to identify trends. In particular, it becomes apparent that NiTROM-based adaptation is much more expensive than OpInf because the optimization requires ambient-space matrix–vector and matrix–matrix operations to optimize the test and trial subspaces against the available data. Given the demonstrated benefits of the NiTROM refinement on the OpInf adaptive models, future work will focus on addressing this issue and reducing the cost associated with matrix-manifold optimization.

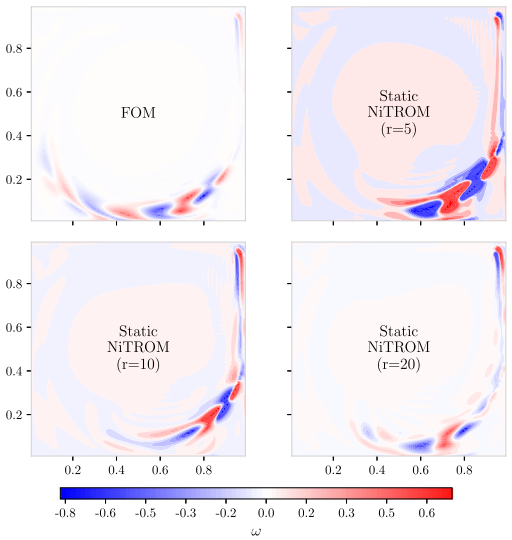
5 Conclusion and outlook

This work introduced a general framework and a set of design rules to construct *adaptive*, *non-intrusive* reduced-order models (ROM), with the ultimate objective of extending traditional, data-driven ROMs beyond the static setting and toward continuously learning, streaming surrogates. The motivation is to design ROMs that not only compress and



(a) Case 1 (rich training)

(b) Case 2 (regime change)



(c) Case 3 (minimal training)

Fig. 28 Sensitivity of static NiTROM predictions to the reduced dimension r. Vorticity fields at the end of the prediction window for cases 1–3 comparing the full-order model with static NiTROM models using $r = 5$, $r = 10$, and $r = 20$ are shown

Table 3 Qualitative summary of ROM performance across cases

Model	Case 1	Case 2	Case 3
Static Galerkin/OpInf/NiTROM	Partially succeeds	Fails	Fails
Adaptive OpInf	Excellent robustness	Partially fails	Fails
Adaptive NiTROM	Accurate for small Z	Fails	Fails
Adaptive OpInf–NiTROM	**Best overall**	**Best overall**	**Best overall**

approximate a complex dynamical system, but also *evolve with it*. To this end, we propose a comprehensive formulation of adaptive non-intrusive ROMs, together with three concrete realizations: Adaptive Operator Inference (OpInf), Adaptive Non-intrusive Trajectory-based optimization of ROM (NiTROM), and a hybrid Adaptive OpInf–NiTROM model that merges their respective advantages. These methods enable on-the-fly parameter correction using only high-quality snapshots, without any intrusive access to the full-order operators.

Systematic experiments on the two-dimensional lid-driven cavity flow demonstrated that the proposed framework can substantially improve long-term stability and accuracy compared with conventional static ROMs. When the model operates near its training regime, adaptive updates effectively eliminate energy drift and preserve amplitude and phase alignment over long horizons. Under more difficult extrapolation settings, where the system transitions into previously unseen regimes, the hybrid model consistently yields the most physically meaningful and stable results. Together, these findings demonstrate the potential of adaptive non-intrusive ROMs that incorporate periodic online corrections to maintain accuracy as the dynamics evolve beyond the training manifold, over the tested horizons. Despite these promising outcomes, several limitations remain. While the NiTROM-based models demonstrate high accuracy in moderate adaptation cases, their manifold optimization steps are computationally intensive and sensitive to hyperparameters such as the adaptation window size, lookback window size, and the number of online optimization steps. Although these methods are less expensive than full-order simulations, their current cost and sensitivity make deployment challenging in digital twin-type settings. We are also careful to acknowledge that rigorous testing and evaluations are required on a broader range of test problems and data scenarios. At present, the framework represents a *proof of concept* or solver accelerator. Addressing these limitations will require continued algorithmic development to improve robustness and efficiency.

Outlook: Several directions naturally emerge for future research. Firstly, as presented, the maximum speedup of the proposed algorithms is strictly upper bounded by the adaptation window Z, as the full high-fidelity operator is evolved every Z time steps. This may be accelerated by a sampling of the high-fidelity operator followed by interpolation, as in classical intrusive ROMs. Second, developing adaptive, error-based trigger mechanisms could replace the fixed adaptation window with intelligent decisions driven by residual indicators. Third, acceleration of the Adaptive OpInf formulation through incremental or low-rank operator updates could move the method closer to real-time feasibility, especially for high-frequency adaptation. Finally, testing the framework in a true digital twin environment, where adaptation is informed directly by sensor data rather than full-order feedback, represents an essential step toward real-world applications.

Overall, this study provides the first detailed exploration of adaptive non-intrusive reduced-order modeling and demonstrates that continuous, data-driven adaptation is necessary to train *truly* predictive models. We emphasize that the proposed framework should be viewed as a step toward adaptive non-intrusive ROMs rather than a complete solution for long-horizon predictive modeling, particularly in regimes with rapidly evolving dynamics. Nevertheless, the results establish a foundation for the next generation of self-correcting, low-dimensional surrogates capable of long-term, physics-consistent prediction in complex, non-stationary systems. We anticipate that such adaptive ROMs will play a central role in future developments in digital twins, model-based control, and scientific machine learning for high-dimensional dynamical systems.

A Additional flow-field visualizations

To complement the vorticity visualizations in Sec. 4, this appendix provides the corresponding u- and v-velocity fields for all cases (Figs. 29, 30, 31, 32, 33, 34, 35, 36). These figures illustrate the same qualitative trends discussed in Sec. 4: all adaptive models do well in suppressing the non-physical energy growth in case 1, while the hybrid model is the most effective approach in more challenging settings (cases 2 and 3).

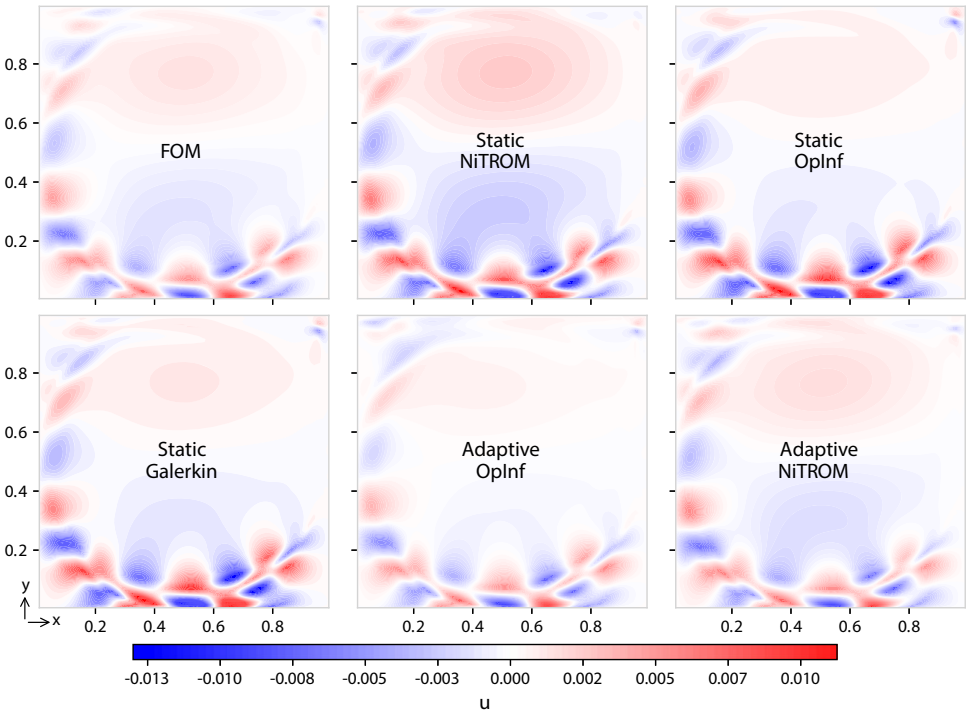


Fig. 29 Ground truth (FOM) and predicted (ROM) u-velocity fields at the end of the online window ($t = 20$) in case 1. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 10$, a lookback window $M = 100$, and a per-adaptation optimization budget $K = 10$

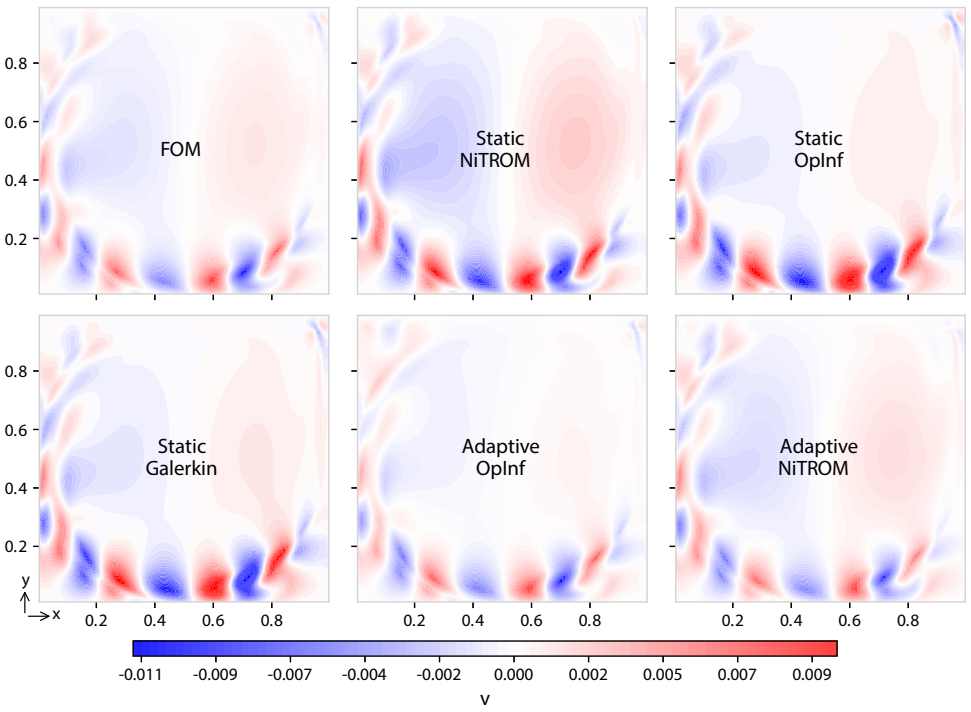


Fig. 30 Ground truth (FOM) and predicted (ROM) v-velocity fields at the end of the online window ($t = 20$) in case 1. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 10$, a lookback window $M = 100$, and a per-adaptation optimization budget $K = 10$

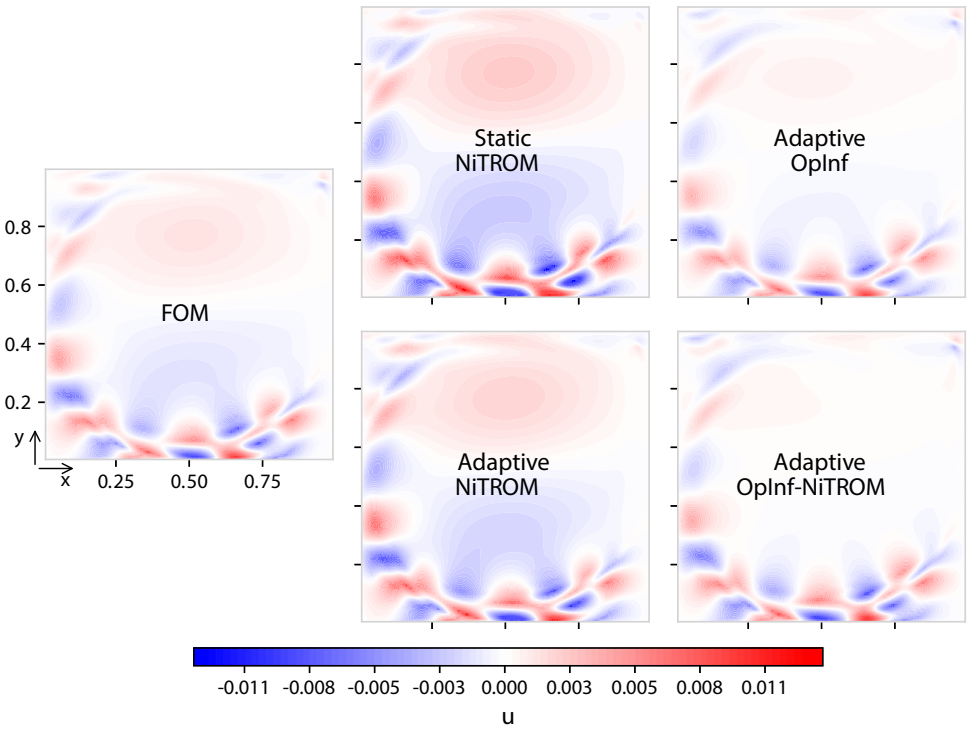


Fig. 31 Ground truth (FOM) and predicted (ROM) u-velocity fields at the end of the online window ($t = 20$) in case 1. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 50$, a lookback window $M = 20$, and a per-adaptation optimization budget $K = 10$

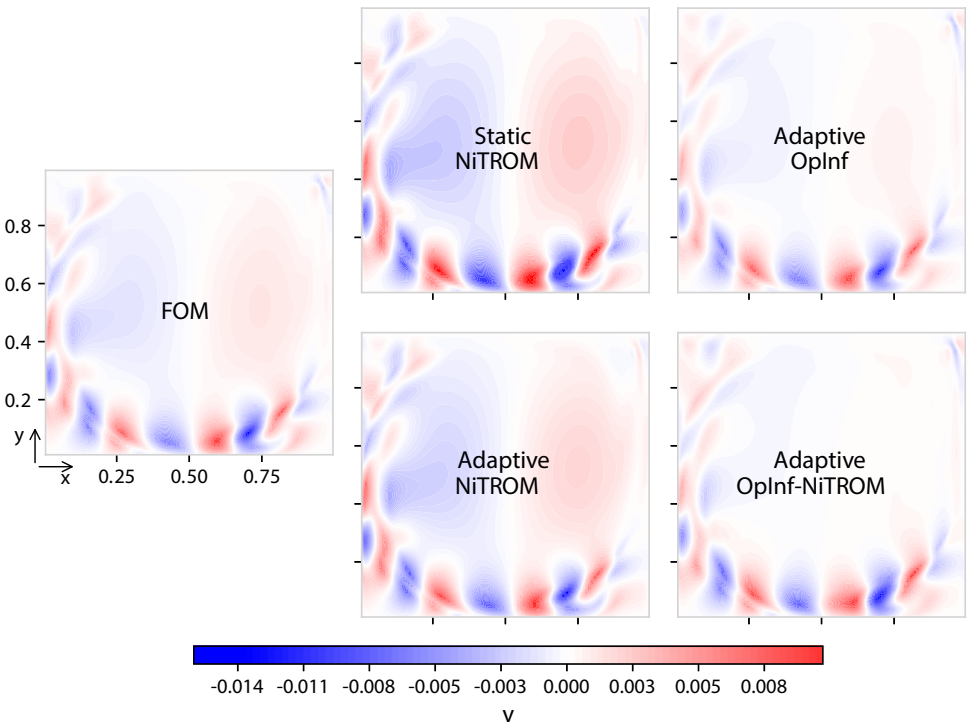


Fig. 32 Ground truth (FOM) and predicted (ROM) v-velocity fields at the end of the online window ($t = 20$) in case 1. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 50$, a lookback window $M = 20$, and a per-adaptation optimization budget $K = 10$

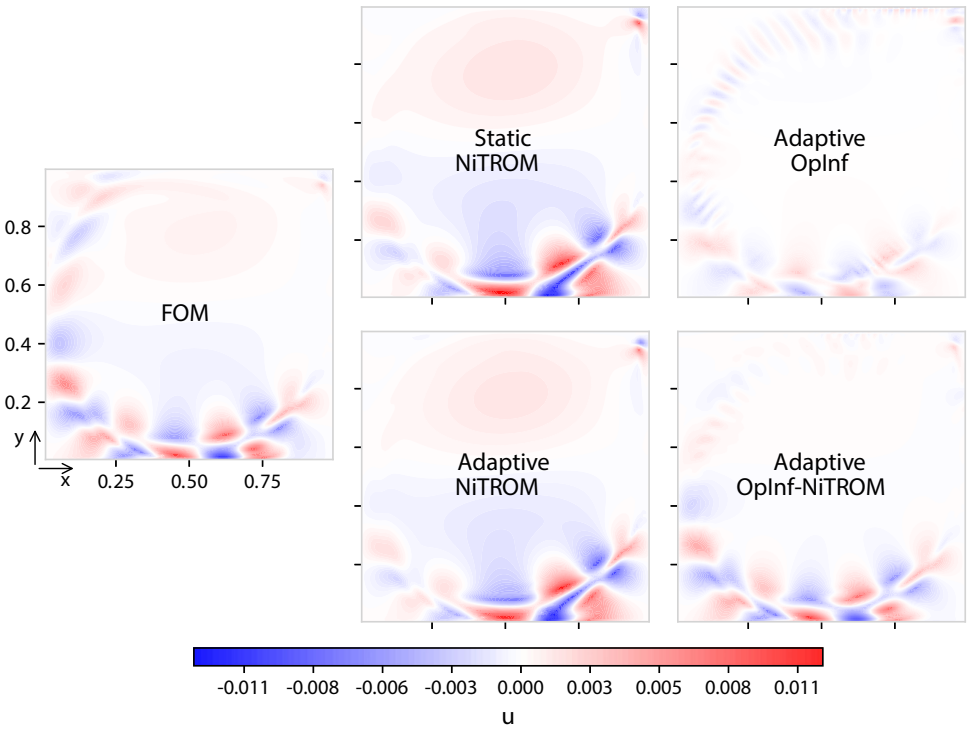


Fig. 33 Ground truth (FOM) and predicted (ROM) u-velocity fields at the end of the online window ($t = 10$) in case 2. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 10$, a lookback window $M = 10$, and a per-adaptation optimization budget $K = 10$

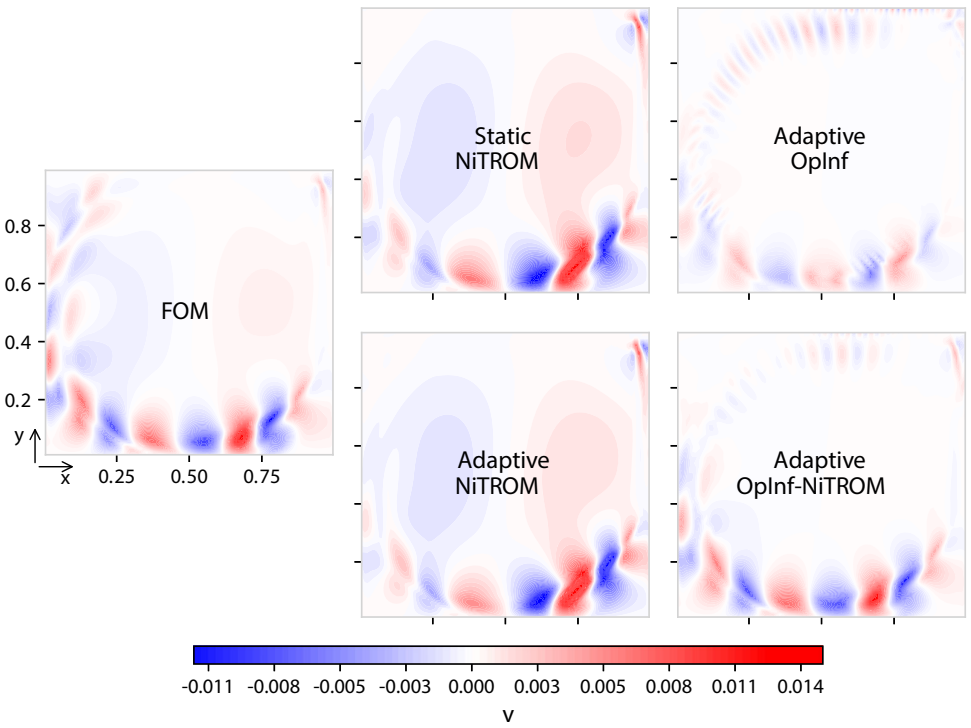


Fig. 34 Ground truth (FOM) and predicted (ROM) v-velocity fields at the end of the online window ($t = 10$) in case 2. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 10$, a lookback window $M = 10$, and a per-adaptation optimization budget $K = 10$

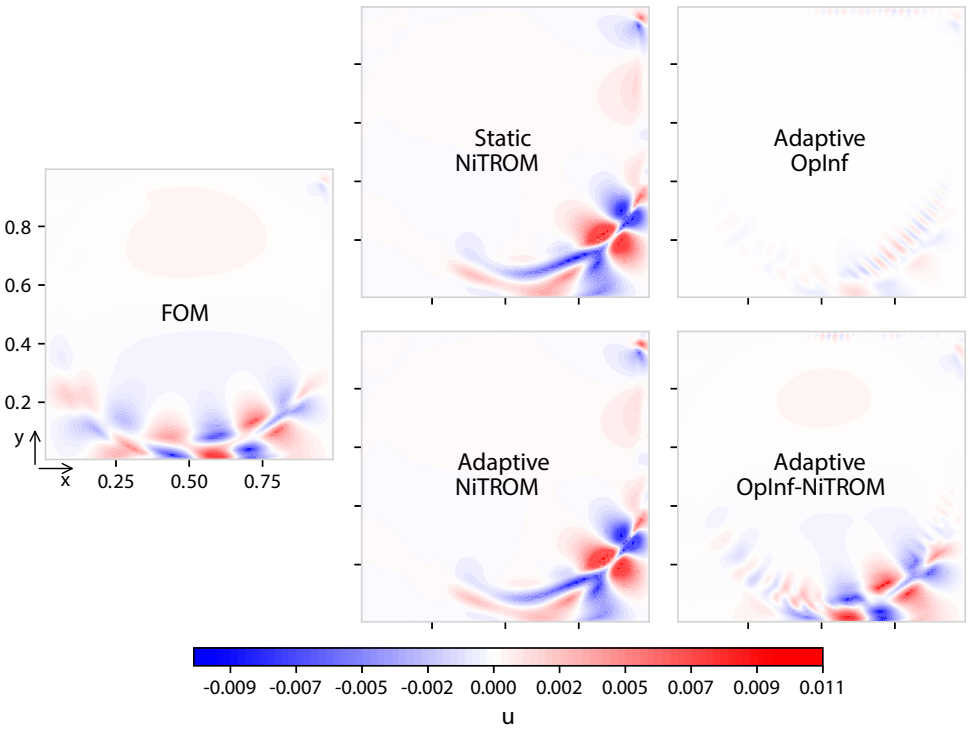


Fig. 35 Ground truth (FOM) and predicted (ROM) u-velocity fields at the end of the online window ($t = 6.25$) in case 3. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 10$, a lookback window $M = 10$, and a per-adaptation optimization budget $K = 10$

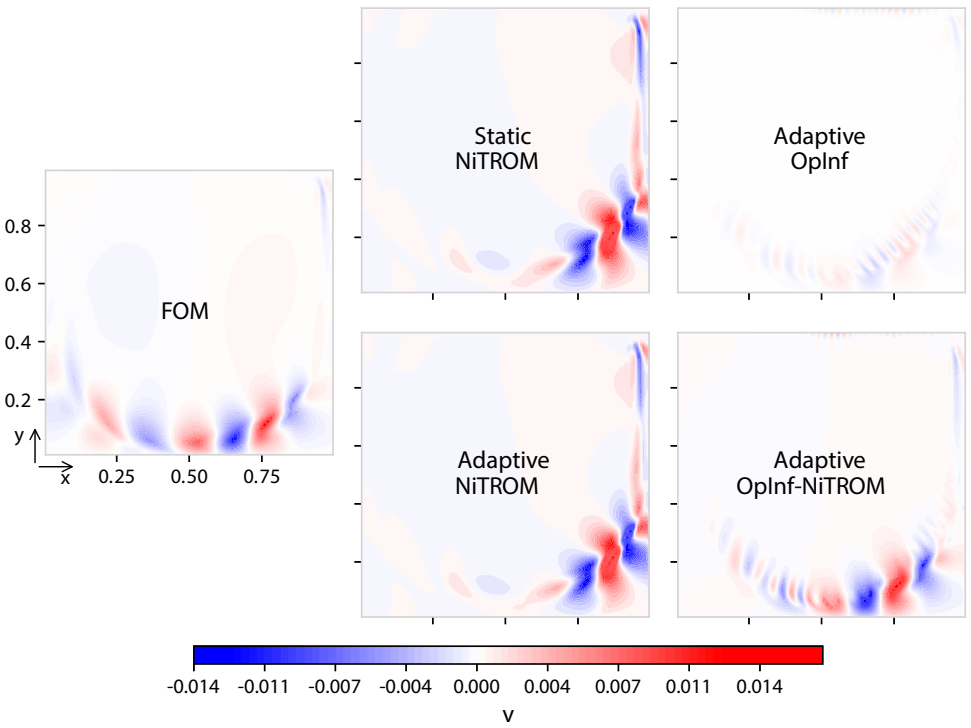


Fig. 36 Ground truth (FOM) and predicted (ROM) v-velocity fields at the end of the online window ($t = 6.25$) in case 3. All models are quadratic and $r = 10$-dimensional. For adaptive models we have an adaptation window $Z = 10$, a lookback window $M = 10$, and a per-adaptation optimization budget $K = 10$

Acknowledgements The authors acknowledge insightful discussions with Prof. Laura Balzano (University of Michigan).

Author contributions A.H. contributed to methodology, software development, numerical experiments, and writing—original draft and review and editing. A.P. contributed to methodology, software development, supervision, and writing—review and editing. K.D. contributed to conceptualization, funding acquisition, resources, supervision, and writing—review and editing.

Funding A.H. and K.D. were supported by the OUSD(RE) Grant #N00014-21-1-295. A.P. acknowledges startup support from the Newark College of Engineering at the New Jersey Institute of Technology.

Replication of Results and Data Availability The source code of the proposed framework to reproduce the results is publicly available at https://github.com/APHedayat/Adaptive-NiTROM under the MIT license.

Declarations

Conflict of interest On behalf of all authors, the corresponding author declares that there are no conflict of interest associated with this work.

References

Ahmed SE, Pawar S, San O, Rasheed A, Iliescu T, Noack BR (2021) On closures for reduced order models—a spectrum of first-principle to machine-learned avenues. Phys Fluids 10(1063/5):0061577

Amsallem D, Farhat C (2008) Interpolation method for adapting reduced-order models and application to aeroelasticity. AIAA J 46(7):1803–1813. https://doi.org/10.2514/1.35374

Amsallem D, Farhat C (2011) An online method for interpolating linear parametric reduced-order models. SIAM J Sci Comput 33(5):2169–2198. https://doi.org/10.1137/100813051

Amsallem D, Zahr MJ, Farhat C (2012) Nonlinear model order reduction based on local reduced-order bases. Int J Numer Meth Eng 92(10):891–916. https://doi.org/10.1002/nme.4371

Antoulas AC, Beattie CA, Gugercin S (2020) Interpolatory methods for model reduction. Soc Ind Appl Math. https://doi.org/10.1137/1.9781611976083

Aretz N, Willcox K (2025) Nested operator inference for adaptive data-driven learning of reduced-order models. https://doi.org/10.48550/arXiv.2508.11542

Arnold-Medabalimi N, Huang C, Duraisamy K (2022) Large-eddy simulation and challenges for projection-based reduced-order modeling of a gas turbine model combustor. Int J Spray Combust Dyn 14(1–2):153–175. https://doi.org/10.1177/175682772211006

Bai F, Wang Y (2022) A reduced order modeling method based on GNAT-embedded hybrid snapshot simulation. Math Comput Simul 199:100–132. https://doi.org/10.1016/j.matcom.2022.03.006

Balzano L, Nowak R, Recht B (2010) Online identification and tracking of subspaces from highly incomplete information. In: 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp 704–711. https://doi.org/10.1109/ALLERTON.2010.5706976

Balzano L, Nowak R, Recht B (2010) Online identification and tracking of subspaces from highly incomplete information. In: 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp 704–711. https://doi.org/10.1109/ALLERTON.2010.5706976

Barone MF, Kalashnikova I, Segalman DJ, Thornquist HK (2009) Stable Galerkin reduced order models for linearized compressible flow. J Comput Phys 228(6):1932–1946. https://doi.org/10.1016/j.jcp.2008.11.015

Benner P, Gugercin S, Willcox K (2015) A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Rev 57(4):483–531. https://doi.org/10.1137/130932715

Benner P, Goyal P, Kramer B, Peherstorfer B, Willcox K (2020) Operator inference for non-intrusive model reduction of systems with non-polynomial nonlinear terms. Comput Methods Appl Mech Eng 372:113433. https://doi.org/10.1016/j.cma.2020.113433

Benner P, Goyal P, Heiland J, Duff IP (2022) Operator inference and physics-informed learning of low-dimensional models for incompressible flows. Electron Trans Numer Anal 56:28–51. https://doi.org/10.1553/etna_vol56s28

Bhattacharya K, Hosseini B, Kovachki NB, Stuart AM (2021) Model reduction and neural networks for parametric PDES. SMAI J Comput Math 7:121–157. https://doi.org/10.5802/smai-jcm.74

Brand M (2002) Incremental singular value decomposition of uncertain data with missing values. In: Heyden A, Sparr G, Nielsen M, Johansen P (eds), Computer Vision—ECCV 2002. Springer, Berlin, Heidelberg, pp 707–720

Carlberg K (2015) Adaptive h-refinement for reduced-order models. Int J Numer Meth Eng 102(5):1192–1210. https://doi.org/10.1002/nme.4800

Carlberg K, Bou-Mosleh C, Farhat C (2011) Efficient non-linear model reduction via a least-squares Petrov-Galerkin projection and compressive tensor approximations. Int J Numer Meth Eng 86(2):155–181. https://doi.org/10.1002/nme.3050

Carlberg K, Barone M, Antil H (2017) Galerkin v. least-squares Petrov-Galerkin projection in nonlinear model reduction. J Comput Phys 330:693–734. https://doi.org/10.1016/j.jcp.2016.10.033

Chomaz J-M (2005) Global instabilities in spatially-developing flows: non-normality and nonlinearity. Annu Rev Fluid Mech 37:357–392. https://doi.org/10.1146/annurev.fluid.37.061903.175810

Conti P, Gobat G, Fresca S, Manzoni A, Frangi A (2023) Reduced order modeling of parametrized systems through autoencoders and sindy approach: continuation of periodic solutions. Comput Methods Appl Mech Eng 411:116072. https://doi.org/10.1016/j.cma.2023.116072

Etter PA, Carlberg KT (2020) Online adaptive basis refinement and compression for reduced-order models via vector-space sieving. Comput Methods Appl Mech Eng 364:112931. https://doi.org/10.1016/j.cma.2020.112931

Feng L, Fu G, Wang Z (2021) A FOM/ROM hybrid approach for accelerating numerical simulations. J Sci Comput 89(3):61. https://doi.org/10.1007/s10915-021-01668-9

Fresca S, Dede' L, Manzoni A (2021) A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs. J Sci Comput 87(2):61. https://doi.org/10.1007/s10915-021-01462-7

Geelen R, Willcox K (2022) Localized non-intrusive reduced-order modelling in the operator inference framework. Philos Trans R

Soc A Math Phys Eng Sci 380(2229):20210206. https://doi.org/10.1098/rsta.2021.0206
Geelen R, Wright S, Willcox K (2023) Operator inference for non-intrusive model reduction with quadratic manifolds. Comput Methods Appl Mech Eng 403:115717. https://doi.org/10.1016/j.cma.2022.115717
Hesthaven JS, Ubbiali S (2018) Non-intrusive reduced order modeling of nonlinear problems using neural networks. J Comput Phys 363:55–78. https://doi.org/10.1016/j.jcp.2018.02.037
Holmes P, Lumley JL, Berkooz G (1996) Turbulence, coherent structures, dynamical systems and symmetry. Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511622700
Huang C, Duraisamy K (2023) Predictive reduced order modeling of chaotic multi-scale problems using adaptively sampled projections. J Comput Phys 491:112356. https://doi.org/10.1016/j.jcp.2023.112356
Jung KS, Lacey CE, Babaee H, Chen JH (2025) Accelerating high-fidelity simulations of chemically reacting flows using reduced-order modeling with time-dependent bases. Comput Methods Appl Mech Eng 437:117758. https://doi.org/10.1016/j.cma.2025.117758
Kramer B, Peherstorfer B, Willcox KE (2024) Learning nonlinear reduced models from data with operator inference. Annu Rev Fluid Mech 56(2024):521–548. https://doi.org/10.1146/annurev-fluid-121021-025220
Lee K, Carlberg KT (2020) Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. J Comput Phys 404:108973. https://doi.org/10.1016/j.jcp.2019.108973
Lumley JL (1967) The structure of inhomogeneous turbulence. pp 166–178
McQuarrie SA, Huang C, Willcox KE (2021) Data-driven reduced-order models via regularised operator inference for a single-injector combustion process. J R Soc N Z 51(2):194–211. https://doi.org/10.1080/03036758.2020.1863237
McQuarrie SA, Khodabakhshi P, Willcox KE (2023) Nonintrusive reduced-order models for parametric partial differential equations via data-driven operator inference. SIAM J Sci Comput 45(4):1917–1946. https://doi.org/10.1137/21M1452810
Mohaghegh A, Huang C (2026) Feature-guided sampling strategy for adaptive model order reduction of convection-dominated problems. J Comput Phys 545:114468. https://doi.org/10.1016/j.jcp.2025.114468
Moore B (1981) Principal component analysis in linear systems: controllability, observability, and model reduction. IEEE Trans Autom Control 26(1):17–32. https://doi.org/10.1109/TAC.1981.1102568
Otto SE, Padovan A, Rowley CW (2022) Optimizing oblique projections for nonlinear systems using trajectories. SIAM J Sci Comput 44(3):1681–1702. https://doi.org/10.1137/21M1425815
Otto SE, Padovan A, Rowley CW (2023) Model reduction for nonlinear systems by balanced truncation of state and gradient covariance. SIAM J Sci Comput 45(5):2325–2355. https://doi.org/10.1137/22M1513228
Otto SE, Macchio GR, Rowley CW (2023) Learning nonlinear projections for reduced-order modeling of dynamical systems using constrained autoencoders. Chaos Interdiscip J Nonlinear Sci 33(11):113130. https://doi.org/10.1063/5.0169688
Padovan A, Vollmer B, Bodony DJ (2024) Data-driven model reduction via non-intrusive optimization of projection operators and reduced-order dynamics. SIAM J Appl Dyn Syst 23(4):3052–3076. https://doi.org/10.1137/24M1628414
Patil P, Babaee H (2023) Reduced-order modeling with time-dependent bases for PDES with stochastic boundary conditions. SIAM/ASA J Uncertain Quantification 11(3):727–756. https://doi.org/10.1137/21M1468097
Peherstorfer B (2020) Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. SIAM J Sci Comput 42(5):2803–2836. https://doi.org/10.1137/19M1257275
Peherstorfer B (2020) Sampling low-dimensional markovian dynamics for preasymptotically recovering reduced models from data with operator inference. SIAM J Sci Comput 42(5):3489–3515. https://doi.org/10.1137/19M1292448
Peherstorfer B, Willcox K (2015) Dynamic data-driven reduced-order models. Comput Methods Appl Mech Eng 291:21–41. https://doi.org/10.1016/j.cma.2015.03.018
Peherstorfer B, Willcox K (2015) Online adaptive model reduction for nonlinear systems via low-rank updates. SIAM J Sci Comput 37(4):2123–2150. https://doi.org/10.1137/140989169
Peherstorfer B, Willcox K (2016) Data-driven operator inference for nonintrusive projection-based model reduction. Comput Methods Appl Mech Eng 306:196–215. https://doi.org/10.1016/j.cma.2016.03.025
Peherstorfer B, Butnaru D, Willcox K, Bungartz H-J (2014) Localized discrete empirical interpolation method. SIAM J Sci Comput 36(1):168–192. https://doi.org/10.1137/130924408
Qian E, Kramer B, Peherstorfer B, Willcox K (2020) Lift & learn: physics-informed machine learning for large-scale nonlinear dynamical systems. Physica D 406:132401. https://doi.org/10.1016/j.physd.2020.132401
Qian E, Kramer B, Marques AN, Willcox KE (2019) Transform & Learn A data-driven approach to nonlinear model reduction. AIAA Aviation 2019 Forum. https://doi.org/10.2514/6.2019-3707
Qian E, Kramer B, Marques AN, Willcox KE (2019) Transform & Learn A data-driven approach to nonlinear model reduction. AIAA Aviation 2019 Forum. https://doi.org/10.2514/6.2019-3707
Ramezanian D, Nouri AG, Babaee H (2021) On-the-fly reduced order modeling of passive and reactive species via time-dependent manifolds. Comput Methods Appl Mech Eng 382:113882. https://doi.org/10.1016/j.cma.2021.113882
Rezaian E, Duraisamy K (2023) Data-driven balanced truncation for predictive model order reduction of aeroacoustic response. AIAA J 61(10):4524–4545. https://doi.org/10.2514/1.J063116
Rezaian E, Huang C, Duraisamy K (2022) Non-intrusive balancing transformation of highly stiff systems with lightly damped impulse response. Phil Trans R Soc A 380(2229):20210202. https://doi.org/10.1098/rsta.2021.0202
Rowley CW (2005) Model reduction for fluids, using balanced proper orthogonal decomposition. Int J Bifurc Chaos 15(03):997–1013. https://doi.org/10.1142/S0218127405012429
Rowley CW, Dawson STM (2017) Model reduction for flow analysis and control. Annu Rev Fluid Mech 49(2017):387–417. https://doi.org/10.1146/annurev-fluid-010816-060042
Rowley CW, Colonius T, Murray RM (2004) Model reduction for compressible flows using POD and Galerkin projection. Physica D 189(1–2):115–129. https://doi.org/10.1016/j.physd.2003.03.001
Rowley CW, Mezic I, Bagheri S, Schlatter P, Henningson DS (2009) Spectral analysis of nonlinear flows. J Fluid Mech 641:115–127. https://doi.org/10.1017/S0022112009992059
Schmid PJ (2010) Dynamic mode decomposition of numerical and experimental data. J Fluid Mech 656:5–28. https://doi.org/10.1017/S0022112010001217
Singh R, Uy WIT, Peherstorfer B (2023) Lookahead data-gathering strategies for online adaptive model reduction of transport-dominated problems. Chaos Interdiscip J Nonlinear Sci 33(11):113112. https://doi.org/10.1063/5.0169392
Sirovich L (1987) Turbulence and the dynamics of coherent structures. Part I: coherent structures. Q Appl Math 45(3):561–571. https://doi.org/10.1090/qam/910462
Swischuk R, Kramer B, Huang C, Willcox K (2020) Learning physics-based reduced-order models for a single-injector combus-

tion process. AIAA J 58(6):2658–2672. https://doi.org/10.2514/1.J058943

Willcox K, Peraire J (2002) Balanced model reduction via the proper orthogonal decomposition. AIAA J 40(11):2323–2330. https://doi.org/10.2514/2.1570

Xu J, Duraisamy K (2020) Multi-level convolutional autoencoder networks for parametric prediction of spatio-temporal dynamics. Comput Methods Appl Mech Eng 372:113379. https://doi.org/10.1016/j.cma.2020.113379

Yano M, Huang T, Zahr MJ (2021) A globally convergent method to accelerate topology optimization using on-the-fly model reduction. Comput Methods Appl Mech Eng 375:113635. https://doi.org/10.1016/j.cma.2020.113635

Zanardi I, Padovan A, Bodony DJ, Panesi M (2025) Petrov-Galerkin model reduction for thermochemical nonequilibrium gas mixtures. J Comput Phys 533:113999. https://doi.org/10.1016/j.jcp.2025.113999

Zimmermann R, Peherstorfer B, Willcox K (2018) Geometric subspace updates with applications to online adaptive nonlinear model reduction. SIAM J Matrix Anal Appl 39(1):234–261. https://doi.org/10.1137/17M1123286

Zucatti V, Zahr MJ (2024) An adaptive, training-free reduced-order model for convection-dominated problems based on hybrid snapshots. Int J Numer Meth Fluids 96(2):189–208. https://doi.org/10.1002/fld.5240