SIAM J. SCI. COMPUT.
Vol. 47, No. 2, pp. A633–A656

A ROBUST SHIFTED PROPER ORTHOGONAL DECOMPOSITION: PROXIMAL METHODS FOR DECOMPOSING FLOWS WITH MULTIPLE TRANSPORTS*

PHILIPP KRAH†, ARTHUR MARMIN†, BEATA ZORAWSKI‡, JULIUS REISS§, AND KAI SCHNEIDER†

Abstract. We present a new methodology for decomposing flows with multiple transports that further extends the shifted proper orthogonal decomposition (sPOD). The sPOD approximates transport-dominated flows by a sum of co-moving data fields. The proposed methods are derived from sPOD but optimize the co-moving fields directly and penalize their nuclear norm to promote a low rank of the individual data in the decomposition. A robustness term is added for handling interpolation errors and data noise. Using convex optimization tools, we derive three proximal algorithms to solve the decomposition problem. We report a numerical comparison with existing methods using synthetic data and then show the separation ability of our methods for incompressible and reactive flows. The resulting methodology is the basis of a new analysis paradigm that provides the same interpretability as POD for the individual co-moving fields.

Key words. forward-backward, transport phenomena, proper orthogonal decomposition, vortex shedding, reactive flows

MSC codes. 65Z05, 35Q35

DOI. 10.1137/24M164392X

1. Introduction.

Modeling of flows in time-varying geometries or of expanding reaction waves poses a major mathematical challenge due to the inherent difficulty of efficiently reducing the number of degrees of freedom (DOFs). For instance, high-fidelity simulations on massively parallel computing architectures are typically performed in multiquery applications in order to understand the flight mechanics of an insect or the spread of a fire [71, 73] for a range of parameters, e.g., Reynolds numbers or burning rates. These simulations are costly due to the tremendous number of DOFs in the system. A common approach is to reduce the DOFs using proper orthogonal decomposition (POD) and Galerkin projections, which were originally introduced in [7, 47]. For a review of the POD-Galerkin model order reduction (MOR) approach, we refer the reader to [6, 43]. However, POD-Galerkin projects the analyzed system onto a reduced linear subspace, which is often not able to capture the full dynamics

*Submitted to the journal's Numerical Algorithms for Scientific Computing section March 7, 2024; accepted for publication (in revised form) October 2, 2024; published electronically March 4, 2025.
https://doi.org/10.1137/24M164392X
Funding: Philipp Krah and Kai Schneider are supported by the project ANR-20-CE46-0010, CM2E. Philipp Krah and Julius Reiss gratefully acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) as part of GRK2433 DAEDALUS. Beata Zorawski acknowledges the funding from DAAD for the ERASMUS+ membership during her stay in Marseille and hospitality of I2M. The authors were granted access to the HPC resources of IDRIS under the Allocation A0142A14152, project 2023-91664 attributed by GENCI (Grand Équipement National de Calcul Intensif) and of Centre de Calcul Intensif d'Aix-Marseille Université.

†Aix-Marseille Université, CNRS, Institut de Mathématiques de Marseille (I2M), UMR 7373, 13013 Marseille, France (philipp.krah@univ-amu.fr, arthur.marmin@univ-amu.fr, kai.schneider@univ-amu.fr).
‡Aix-Marseille Université, CNRS, Institut de Mathématiques de Marseille (I2M), UMR 7373, 13013 Marseille, France, and Technische Universität Berlin, Institute of Mathematics, Berlin, Germany (beatazorawski@gmail.com).
§Technische Universität Berlin, Institute of Fluid Mechanics and Engineering Acoustics, 10623 Berlin, Germany (reiss@tnt.tu-berlin.de).

of the system. Therefore, it leads to large approximation errors. In particular, in the presence of moving quantities or structures with small support, the POD-Galerkin approach breaks down, which limits its applicability in the investigation of transport-dominated fluid systems.

In this work, we build on the shifted POD method (sPOD) [63], which enriches the reduced linear subspace of the POD by moving it using the transport of the system and thus achieves better approximation errors. Although multiple gradient-based optimization algorithms for the sPOD already exist in the literature [10, 62, 63], the method needs to be generalized to achieve a better separation and a more accurate description of the transport phenomena, in particular for multiple transports. We thus derive in this work three proximal algorithms that generalize the existing framework.

1.1. Model order reduction for transport-dominated systems. Model order reduction for transport-dominated systems has been widely studied in the literature (see [59] for a review), since it is one of the key challenges, for instance, in reducing combustion systems [31]. Transport-dominated systems are especially challenging because traditional MOR based on low-dimensional linear subspace approximations breaks down. This phenomenon is known as the Kolmogorov n-width barrier and was theoretically studied in [27, 55]. For linear transports, the Kolmogorov n-width decay was recently proven not to be a problem of the partial differential equation (PDE) itself; it depends on the smoothness of the initial condition and the boundary values [2]. However, transport-dominated systems occur not only in linear advected systems but also, for instance, in combustion systems [11, 17, 31, 42], flows around moving geometries [37, 38, 41], kinetic systems [8, 36], and moment models [35].

To overcome the slow decay of the approximation errors in transport-dominated systems with the approximation dimension, a description that adapts to the transport of the system can be used. One can subdivide the literature mainly into three different groups. The first group builds on the expressivity of neural networks [23, 24, 30, 33, 45, 68, 74]. More specifically, the authors in [23, 33, 36, 45] rely on autoencoder (AE) structures. Unfortunately, AEs often compromise the structural insights, such as the interpretability of the identified structures and the optimality of the resulting description. Nonetheless, physics-informed neural networks (PINNs) [33] can still provide an understanding of the internal low-dimensional dynamics. Furthermore, AE neural networks are used in [36] to identify and interpret the correspondence between the intrinsic variables and the learned structures, e.g., in Boltzmann equations. A combination of classical reduction methods like POD or kernel POD with neural networks has also been studied [24, 30, 68, 74]. It allows for a better quantification of the errors [16], which is usually not possible with classical AE.

The second group uses online-adaptive basis methods [34, 58, 60] that compute the linear approximation space locally and adaptively in time. Consequently, they fully omit the costly data sampling stage of classical MOR. However, [35] shows that the construction and update of the basis in each time step lead to a significant overhead and computational cost compared to the classical approach in which the reduced order model (ROM) is set up a priori, based on the data generated by the original PDE system.

The last group, which includes sPOD, builds on the idea of transport compensation [1, 22, 32, 40, 42, 50, 53, 63, 64, 65, 67, 70, 76], which aims at enhancing the approximation of a linear description by aligning the parameters or time-dependent structures with the help of suited transformations. A subset of the group can be further subdivided into Lagrangian approximation [1, 50, 53, 70, 76] and multiframe approximation methods [10, 63, 64]. While the former use one-to-one transforma-

tions between the snapshot and approximation spaces, the latter use a combination of these transformations that must not result in a one-to-one correspondence between approximation and snapshot spaces. Lagrangian approximations usually introduce a reference mesh, which is deformed using, for example, a space-time registration to align the mesh to local features in the flow [70], transport maps [53], transformed snapshots [75, 76], or low-rank deformations computed with neural networks [50]. Multiframe approximations have the advantage of a higher expressivity that allows handling topological changes or multiple transports. Indeed, multiframe approximations reduce to Lagrangian approximations if only one frame is assumed. On the other hand, Lagrangian approximations have the advantage that they yield less complex optimization problems, since a diffeomorphic relation is assumed. Therefore, the mappings are usually more flexible and complex. Apart from sPOD, other methods from this group include symmetry reduction, which combines symmetries, like translation invariance of the underlying PDE, with a POD reduction approach [22, 67]. This approach was proven to be a special case of the sPOD as shown in [10]. In [65], transported subspaces are used by explicitly leveraging the characteristics of the hyperbolic PDEs or by tracking the front of the reduced system [40, 42].

The sPOD enjoys a close connection to the snapshot POD [69], which is not only a data reduction method but also a tool for the analysis of fluid systems transient dynamics in vortex shedding [52], coherent structures in swirling jets [54], and stability analysis [7]. The sPOD can be viewed as a natural extension of the POD, offering similar interpretability after isolating individual co-moving structures. Such a close connection to the POD makes the sPOD an attractive method to develop further.

In addition to data analysis, sPOD has been extensively utilized in MOR [10, 11, 17, 26, 49, 37, 38, 48, 57]. This includes an intrusive MOR approach [10], specifically tailored for sPOD, which projects the original set of equations onto the nonlinear reduced manifold created by the sPOD. To handle nonlinearities in the resulting ROM, a tailored hyper-reduction strategy was developed to improve efficiency [11]. Most other methods employ nonintrusive MOR to predict unseen parameters or time instances [17, 26, 49, 37, 38, 48, 57]. These applications range from particle-laden flows [37, 38] to rotating detonation waves [49]. The manifold of presented studies demonstrates that nonintrusive approaches, in combination with sPOD, are advantageous due to the purely data-driven nature of the resulting models, which are less complex. We highlight that the decomposition approach presented in this paper has already been used to predict new states in conjunction with deep learning [17].

1.2. State of the art. The sPOD was first introduced in [63] based on a heuristic optimization of a residuum. The method builds on the idea that a single traveling wave or moving localized structure can be perfectly described by its wave profile and a time-dependent shift. Therefore, the sPOD decomposes transport fields by shifting the data field in multiple co-moving frames, in which the different waves are stationary and can be described with a few spatial basis functions determined by POD. The sPOD was then further developed in [10, 11, 12, 62]. More specifically, the sPOD was studied in its space-time continuous formulation in [10] before being discretized and solved as an optimization problem. The formulation was proven to have a solution under the assumption that the involved transformations are smooth. This formulation was later generalized in [12] to include the optimization of the shifts using initial shifts that are already close to the optimum. Nevertheless, the presented decomposition approach has not been used in the context of efficient ROMs. A first application of sPOD for efficient ROMs is given in [11], but the method relies on cutting the domain such that two distinct co-moving systems can be found and separated. Furthermore, the

decomposition relies on choosing the ranks of each co-moving field beforehand. For complicated systems, this choice is often critical for the quality of the decomposition.

In contrast, [62] proposes a discrete optimization problem based on the decay of singular values in each co-moving field. The problem shares more similarities with the discrete space implementation of the POD, which technically boils down to a singular value decomposition (SVD). Minimizing the nuclear norm of the co-moving fields results in a nonstrictly convex problem, which is easy to solve under the assumption of the convexity of the transport operators. Additionally, the ranks in each co-moving field are selected during the minimization. Unfortunately, the gradient-based optimization approach presented in [62] shows slow convergence, due to the nonsmoothness of the nuclear norm. Moreover, this method is not robust to noise since the exact ranks of the synthetic test cases cannot be estimated correctly. In this work, we propose a method to circumvent these two impediments.

1.3. Contribution and outline. Our contribution is as follows:

1. Three proximal algorithms are proposed to solve the sPOD formulation; two of them enjoy desirable theoretical properties such as descent property and convergence to a critical point, even in a nonconvex setting. These properties are important since, in contrast with [62], the convexity of the transport operators is not assumed.
2. An additional noise term can be included to capture interpolation noise or artifacts in the data to accurately predict the ranks of the system.
3. Our algorithms are compared with existing methods.
4. Applications of our methods to realistic 2D incompressible and 2D reactive flows are presented.

The main novelty of this work is to show that the new algorithms lead to a better and more efficient separation of the physical phenomena, which opens research for building surrogate models of individual systems.

The article is organized as follows: Section 2 introduces the sPOD problem in the continuous and discrete settings with our proposed generalization towards a robust decomposition. In section 3, we reformulate the discrete sPOD problem and leverage tools from convex optimization to design three algorithms that solve the latter problem. Results of the numerical experiments are presented and discussed in section 4. Conclusions are drawn in section 5.

Notation. Bold upper case letters denote matrices, bold lower case letters denote vectors, and lowercase letters denote scalars. The notation $\|.\|_*$ and $\|.\|_{\mathrm{F}}$ denotes the nuclear and the Frobenius norms of a matrix, respectively. The set $[\![1, N]\!]$ denotes the set of natural integers from 1 to N. In the following, we refer to a critical point for a function f as a point where its subdifferential contains 0.

2. Shifted POD. The sPOD is a nonlinear decomposition of a transport-dominated field $q(x,t)$ into multiple co-moving structures $\{q^k(x,t)\}_{k\in[\![1,K]\!]}$ with their respective transformations $\{\mathcal{T}^k\}_{k\in[\![1,K]\!]}$,

$$q(x,t) = \sum_{k=1}^{K} \mathcal{T}^k q^k(x,t)\,, \tag{2.1}$$

where K is the number of co-moving frames. The transformations are usually chosen such that the resulting co-moving structure can be described efficiently with the help of a dyadic decomposition

$$q^k(x,t) \approx \sum_{r=1}^{R_k} \alpha_r^k(x)\phi_r^k(t)\,, \tag{2.2}$$

where R_k is the co-moving rank. Hence, the total number of DOFs in the approximation is $R = \sum_{k=1}^{K} R_k$. The operator $\mathcal{T}^k$ transforms the co-moving coordinate frame into the reference frame, while its inverse $\mathcal{T}^{-k} := (\mathcal{T}^k)^{-1}$ transforms it back. For the sake of clarity, we present the operators as shift transformations $\Delta^k(t)$ which smoothly depend on time

$$\mathcal{T}^k q^k(x,t) = q^k(x-\Delta^k(t),t), \quad \mathcal{T}^{-k} q^k(x,t) = q^k(x+\Delta^k(t),t)\,. \tag{2.3}$$

However, as shown in [11, 17] it is straightforward to also include additional parameter dependencies in the transformations. Furthermore, the transformations can also include rotations [37, 38], low-rank shifts which depend slowly on space [17], or other diffeomorphic mappings [50]. In general, a single transformation is assumed to be at least piecewise differentiable in time and diffeomorphic. The former assumption is necessary for the differentiability of the ROM and the latter for the invertibility and uniqueness of the individual transformations. Nevertheless, the decomposition in the sense of (2.1) is not unique in general, since multiple diffeomorphic mappings can be involved.

Usually, MOR is performed on a discrete data set. Without loss of generality, we assume one spatial dimension and one temporal dimension for the purpose of the sPOD description. Thus, the data set includes M spatial grid points $\{x_m\}$ and N time grid points $\{t_n\}$. This discretization results in the construction of a snapshot matrix $\mathbf{Q}$:

$$\begin{aligned} \mathbf{Q} &= [\mathbf{q}(t_1), \dots, \mathbf{q}(t_N)] \in \mathbb{R}^{M\times N}, \\ \text{with} \quad \mathbf{q}(t) &= [q(x_1,t), \dots, q(x_M,t)]^\top \in \mathbb{R}^M\,. \end{aligned}$$

Therefore, the shift transformation (2.3) reads

$$\begin{aligned} \mathcal{T}^k\mathbf{Q} &= \left[\mathcal{T}^k\mathbf{q}(t_1), \dots, \mathcal{T}^k\mathbf{q}(t_N)\right] \in \mathbb{R}^{M\times N}, \\ \text{with} \quad \mathcal{T}^k\mathbf{q}(t) &= [q(x_1-\Delta^k(t),t), \dots, q(x_M-\Delta^k(t),t)] \in \mathbb{R}^M\,. \end{aligned}$$

Since $\tilde{x} = x_m - \Delta(t)$ may not lie on the grid, it is interpolated from neighboring grid points. In this work, the interpolation is performed with Lagrange polynomials of degree 5, which introduces an interpolation error of order $\mathcal{O}(h^6)$ [39]. Note that, with a slight abuse of notation, we use $\mathcal{T}^k$ to denote (2.3) and its approximation using Lagrange interpolation. In the remainder of the text, we assume an equidistant, periodic grid with a constant lattice spacing.

Remark 2.1. Nonperiodic domains can be handled by extending the domain Ω into $\bar{\Omega} = \Omega \cup \Omega_{\text{ext}}$ such that all the shift operations stay inside $\bar{\Omega}$. Equation (2.1) is then relaxed into

$$w(x)\left(q(x,t) - \sum_{k=1}^{K} \mathcal{T}^k q^k(x,t)\right) = 0\,, \quad \text{where} \quad (\forall x \in \bar{\Omega}) \quad w(x) = \begin{cases} 1 & \text{if } x \in \Omega, \\ 0 & \text{if } x \in \Omega_{\text{ext}}. \end{cases}$$

Details can be found in [62, section 5].

After discretization, (2.2) and (2.1) result in the following nonlinear matrix decomposition:

$$\mathbf{Q} \approx \tilde{\mathbf{Q}} \overset{\text{def}}{=} \sum_{k=1}^{K} \mathcal{T}^k \mathbf{Q}^k\,. \tag{2.4}$$

In this discrete setting, we optimize the co-moving data fields $\mathbf{Q}^k \in \mathbb{R}^{M\times N}$, which are further decomposed using SVD:

$$(\forall k \in [\![1, K]\!]) \quad \mathbf{Q}^k = \mathbf{\Psi}^k \mathbf{\Sigma}^k (\mathbf{V}^k)^\top . \tag{2.5}$$

Here, $\mathbf{\Sigma}^k = \operatorname{diag}(\sigma_1^k, \dots, \sigma_P^k)$, with $P = \min(M, N)$ is a diagonal matrix containing the singular values $\sigma_1^k \geq \sigma_2^k \geq \cdots \geq \sigma_P^k$ while $\mathbf{\Psi}^k \in \mathbb{R}^{M\times P}$ and $\mathbf{V}^k \in \mathbb{R}^{N\times P}$ are semiorthogonal matrices containing the left and right singular vectors, respectively. The POD modes are contained in the first R_k columns of $\mathbf{\Psi}^k = [\phi_p^k(x_m)]_{mp} \in \mathbb{R}^{M\times P}$. The approximation dimensions $\{R_k\}_k$ need to be estimated adequately. For maximal efficiency, we aim for a small number of modes $R = \sum_{k=1}^K R_k \ll N$. Hence, we can formulate the search of a sPOD decomposition shown in (2.4) as the following optimization problem:

$$\underset{\{\mathbf{Q}^k\}_k}{\text{minimize}} \ \sum_{k=1}^K \operatorname{rank}(\mathbf{Q}^k) \quad \text{s.t. } \mathbf{Q} = \sum_{k=1}^K \mathcal{T}^k \mathbf{Q}^k . \tag{2.6}$$

As minimizing over the rank of a matrix is NP-hard [61], we substitute the nuclear norm for the rank function, the former being the convex hull of the latter. Problem (2.6) is thus relaxed into

$$\underset{\{\mathbf{Q}^k\}_k}{\text{minimize}} \ \sum_{k=1}^K \left\|\mathbf{Q}^k\right\|_* \quad \text{s.t. } \mathbf{Q} = \sum_{k=1}^K \mathcal{T}^k \mathbf{Q}^k . \tag{2.7}$$

This relaxation of the rank function is common in robust PCA [18, 46]. However, relaxing the sum of the ranks to the sum of the nuclear norms is not a tight relaxation: indeed, the convex hull of a sum of functions is not equal to the sum of the convex hulls of each function in general.

Optimization problem (2.7) was already formulated in [62] and it was solved based on a Broyden–Fletcher–Goldfarb–Shanno method with an inexact line search designed for nonsmooth optimization problems. Nonetheless, the convergence was observed to be slow, rendering the method inefficient in practice. Furthermore, convergence to the exact ranks could not be achieved due to the interpolation noise introduced by the discrete transport operators. To circumvent the latter issue, we introduce an extra term $\mathbf{E} \in \mathbb{R}^{M\times N}$ in the sPOD decomposition

$$\mathbf{Q} = \sum_{k=1}^K \mathcal{T}^k(\mathbf{Q}^k) + \mathbf{E}, \tag{2.8}$$

in order to capture both the interpolation noise and the noise that could corrupt the data. The resulting optimization problem thus reads

$$\underset{\{\mathbf{Q}^k\}_k, \mathbf{E}}{\text{minimize}} \ \sum_{k=1}^K \lambda_k \left\|\mathbf{Q}^k\right\|_* + \lambda_{K+1} \|\mathbf{E}\|_1 \quad \text{s.t. } \mathbf{Q} = \sum_{k=1}^K \mathcal{T}^k \mathbf{Q}^k + \mathbf{E}, \tag{2.9}$$

where $\|\mathbf{E}\|_1 = \sum_{ij} |\mathbf{E}_{ij}|$ corresponds to the ℓ_1-norm of the vectorization of $\mathbf{E}$ and $\{\lambda_k\}_{k\in[\![1,K+1]\!]}$ are positive scalar parameters that can be tuned to yield different weights to the terms in the objective function. Similar to the robust PCA, solving (2.9) aims at decomposing $\mathbf{Q} = \tilde{\mathbf{Q}} + \mathbf{E}$ into a low-rank matrix $\tilde{\mathbf{Q}}$ and a sparse noisy matrix $\mathbf{E}$. A visualization of this decomposition is shown in Figure 1.

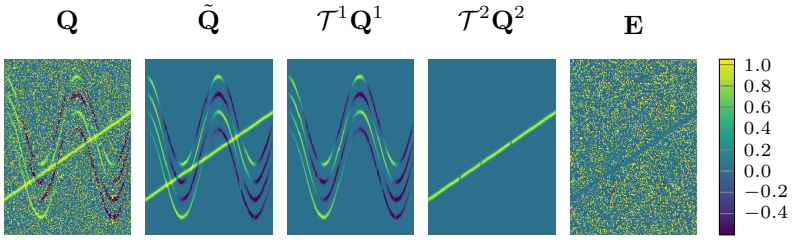


FIG. 1. *Illustration of the robust sPOD. The noise is computed by randomly setting* 12.5% *of the input entries of* $\mathbf{Q}$ *to* 1. *The input data* $\mathbf{Q}$ *and its decomposition into a low-rank part* $\tilde{\mathbf{Q}} = \mathcal{T}^1\mathbf{Q}^1 + \mathcal{T}^2\mathbf{Q}^2$, *as well as the noise matrix* $\mathbf{E}$, *are displayed from left to right.*

3. Low-rank decomposition of the snapshot matrix. This section describes two formulations of (2.9) as well as the design of three algorithms that, given a snapshot matrix $\mathbf{Q}$ and the transport operators $\{\mathcal{T}^k\}_{k\in[\![1,K]\!]}$, return low-rank estimates of the co-moving fields $\{\mathbf{Q}^k\}_{k\in[\![1,K]\!]}$ as well as the residual error $\mathbf{E}$.

3.1. Unconstrained formulation. We first write problem (2.9) as the following unconstrained optimization problem:

$$\underset{\{\mathbf{Q}^k\}_k,\mathbf{E}}{\text{minimize}} \quad \underbrace{\frac{1}{2}\left\|\mathbf{Q}-\sum_{k=1}^{K}\mathcal{T}^k\mathbf{Q}^k-\mathbf{E}\right\|_{\mathrm{F}}^2}_{\stackrel{\text{def}}{=} f(\{\mathbf{Q}^k\}_k,\mathbf{E})} + \sum_{k=1}^{K}\underbrace{\lambda_k\left\|\mathbf{Q}^k\right\|_*}_{\stackrel{\text{def}}{=} g_k(\mathbf{Q}^k)} + \underbrace{\lambda_{K+1}\|\mathbf{E}\|_1}_{\stackrel{\text{def}}{=}\tilde{g}(\mathbf{E})}, \tag{3.1}$$

where f is the data fitting term that forces the optimization variables to fit the snapshot matrix $\mathbf{Q}$, $\{g_k\}_{k\in[\![1,K]\!]}$ promotes low-rank estimates of the $\{\mathbf{Q}^k\}_k$, and $\tilde{g}$ promotes sparse residual error $\mathbf{E}$. Note that problem (3.1) has a solution since its objective function is lower semicontinuous and coercive. Therefore, the set of its minimizers is a nonempty compact set.

We define the regularization term g such that $g(\{\mathbf{Q}^k\}_k,\mathbf{E}) = \sum_{k=1}^{K} g_k(\mathbf{Q}^k) + \tilde{g}(\mathbf{E})$. Hence, by denoting $\mathbf{x} = (\mathbf{Q}^1,\dots,\mathbf{Q}^K,\mathbf{E})^\top$ the vector of optimization variables, problem (3.1) reads

$$\min_{\mathbf{x}} \quad F(\mathbf{x}) \stackrel{\text{def}}{=} f(\mathbf{x}) + g(\mathbf{x})\,. \tag{3.2}$$

The function f is a $\mathcal{C}^1$ nonconvex function with β-Lipschitz gradient while g is a proper lower semicontinuous, convex, nonsmooth, and separable function. The objective function F is bounded from below by 0 since it is the sum of two nonnegative functions, f and g. Splitting algorithms are well suited to solve problems in the form of (3.2) [20].

3.2. Joint proximal gradient method. Splitting problems such as (3.2) have been extensively studied in the convex optimization literature (see [4] and references therein), and an efficient algorithm to solve them is the forward-backward (FB) algorithm, also known as the proximal gradient method [5]. The FB method is an iterative algorithm whose iterations are composed by a gradient step (or forward step) on the smooth term, here f, and a proximal step (or backward step) on the nonsmooth term, here g. A single step at iteration t can be summarized as follows:

$$\mathbf{x}^{(t+1)} \quad \longleftarrow \quad \operatorname{prox}_{\alpha g}\left(\mathbf{x}^{(t)} - \alpha\nabla f(\mathbf{x}^{(t)})\right), \tag{3.3}$$

where the superscript t refers to the current iteration, α is the stepsize, and prox is the proximal operator which is uniquely defined for a proper lower semicontinuous convex function $h : \mathbb{R}^N \to \mathbb{R}^N$ as

$$\mathrm{prox}_h : \mathbf{x} \mapsto \arg\min_{\mathbf{y} \in \mathbb{R}^N} h(\mathbf{y}) + \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 .$$

The proximity operator was introduced in the early work [51] and can be viewed as a generalization of the projection onto a convex set. Indeed, the proximity operator of an indicator function on a convex set is equal to the projection on this set. Moreover, the proximity operator enjoys many projection-like properties such as nonexpansiveness. See [4] for an exhaustive presentation of the proximity operator.

In order to apply the FB algorithm to solve (3.1), we need to perform the iteration (3.3), i.e., to compute the gradient of f as well as the proximity operator of g. Using the definition of the gradient, we have that $\nabla f(\mathbf{x}) = (\nabla_{\mathbf{Q}^1} f(\mathbf{x}), \dots, \nabla_{\mathbf{E}} f(\mathbf{x}))^\top$. The computation of the partial gradients is performed in [62] and yields

$$(\forall k \in [\![1, K]\!]) \quad \nabla_{\mathbf{Q}^k} f(\mathbf{x}) = -\mathcal{T}^{-k}\mathbf{R} \quad \text{and} \qquad \nabla_{\mathbf{E}} f(\mathbf{x}) = -\mathbf{R},$$

where $\mathbf{R}$ is the residual of the approximation $\mathbf{R} = \mathbf{Q} - \sum_{k=1}^K \mathcal{T}^k \mathbf{Q}^k - \mathbf{E}$. Moreover, since g is separable, we have that [4]

$$\mathrm{prox}_g(\mathbf{x}) = \left((\mathrm{prox}_{\lambda_k g_k}(\mathbf{Q}^k))_{k \in [\![1,K]\!]}, \mathrm{prox}_{\lambda_{K+1}\tilde{g}}(\mathbf{E}) \right)^\top .$$

The proximal operator of the ℓ_1-norm is simply the soft thresholding operator applied elementwise [4]. Hence, $\mathrm{prox}_{\alpha\|.\|_1}(\mathbf{E}) = \mathrm{Soft}_{[-\alpha,\alpha]}(\mathbf{E})$ where $\mathrm{Soft}_{[-\alpha,\alpha]}(x) = \mathrm{sgn}(x)\max(0, |x| - \alpha)$. The proximity operator of the nuclear norm of a matrix also has a closed-form expression which is simply the singular value thresholding (SVT) of the matrix [4]

$$\mathrm{prox}_{\alpha\|.\|_*}(\mathbf{Q}^k) = \mathbf{Q}^k = \mathbf{\Psi}^k \,\mathrm{Diag}(\mathbf{d}^k)(\mathbf{V}^k)^\top , \tag{3.4}$$

where $\mathbf{d}^k = \mathrm{Soft}_{[-\alpha,\alpha]}(\boldsymbol{\sigma}^k)$ and $\mathbf{\Psi}^k$, $\boldsymbol{\sigma}^k$, and $(\mathbf{V}^k)^\top$ are the components of the SVD of $\mathbf{Q}^k$ given in (2.5). Tying everything together, (3.3) becomes

$$\begin{pmatrix} \mathbf{Q}^{1,(t+1)} \\ \vdots \\ \mathbf{Q}^{K,(t+1)} \\ \mathbf{E}^{(t+1)} \end{pmatrix} = \begin{pmatrix} \mathrm{SVT}_{\alpha\lambda_1}(\mathbf{Q}^{1,(t)} + \alpha\mathcal{T}^{-1}\mathbf{R}^{(t)}) \\ \vdots \\ \mathrm{SVT}_{\alpha\lambda_K}(\mathbf{Q}^{K,(t)} + \alpha\mathcal{T}^{-K}\mathbf{R}^{(t)}) \\ \mathrm{Soft}_{[-\alpha\lambda_{K+1},\alpha\lambda_{K+1}]}(\mathbf{E}^{(t)} + \alpha\mathbf{R}^{(t)}) \end{pmatrix}, \tag{3.5}$$

which leads to Algorithm 3.1. Note that, for the sake of clarity, we write (3.5) as a for-loop in the pseudo-code of Algorithm 3.1, but it can be implemented with vectorization to speed up the computation.

Convergence of JFB. The convergence of Algorithm 3.1 has been studied extensively in the convex setting in [4]. Nonetheless, f (and thus F) is nonconvex due to the nonconvexity of the transport operators $\{\mathcal{T}^k\}_k$. In this case, convergence to a critical point of problem (3.1) by a finite sequence of iterates has been proved to occur in [3] if

(i) the function F satisfies the Kurdyka–Łojasiewicz (KL) inequality [44],
(ii) and the generated sequence of iterates is bounded.

Algorithm 3.1. Pseudo-code of the joint FB to solve (3.1).

Input: Snapshot matrix $\mathbf{Q} \in \mathbb{R}^{m \times n}$, Transformations $\{\mathcal{T}^k\}_k$.
Input: Initial values $\mathbf{Q}^{1,(0)}, \ldots, \mathbf{Q}^{K,(0)}, \mathbf{E}^{(0)}$, Stepsize $\alpha \in]0, 2/\beta[$.
Output: Estimates of $\mathbf{Q}^1, \ldots, \mathbf{Q}^K, \mathbf{E}$.

Initialize t to 0.
repeat
  Compute the residual of the approximation $\mathbf{R}^{(t)}$ $\mathbf{R}^{(t)} \leftarrow \mathbf{Q} - \sum_{k=1}^{K} \mathcal{T}^k \mathbf{Q}^{k,(t)} - \mathbf{E}^{(t)}$.
  Perform the joint FB step from Equation (3.3):
  for Variable block $\mathbf{Q}^{1,(t)}$ **to** Variable block $\mathbf{Q}^{K,(t)}$ **do**
    Update the block $\mathbf{Q}^k$ in the optimization vector

$$\mathbf{Q}^{k,(t+1)} = \mathrm{SVT}_{\alpha\lambda_k}(\mathbf{Q}^{k,(t)} + \alpha \mathcal{T}^{-k} \mathbf{R}^{(t)}) .$$

  end for
  Update the last block $\mathbf{E}^{(t)}$

$$\mathbf{E}^{(t+1)} = \mathrm{Soft}_{[-\alpha\lambda_{K+1}, \alpha\lambda_{K+1}]}(\mathbf{E}^{(t)} + \alpha \mathbf{R}^{(t)}) .$$

  Increment t.
until stopping criterion is met.
return $\{\mathbf{Q}^{k,(t)}\}_k, \mathbf{E}^{(t)}$

Functions that satisfy the KL inequality form a wide class of functions, which encompasses semialgebraic and real analytic functions [13, 44]. The transport operators which are useful in applications satisfy the KL inequality, as we shall see in section 4. Moreover, the FB algorithm satisfies the descent lemma [3]. As a consequence, the objective function F is guaranteed to decrease at each iteration. Algorithm 3.1 thus enjoys interesting theoretical properties.

3.3. Block-coordinate descent proximal gradient method. In section 3.2, we applied a direct approach to solve (3.1). In contrast, we propose here a second approach that consists in using a block coordinate descent (BCD) approach where we update one matrix amongst the optimization variables $\mathbf{x} = (\{\mathbf{Q}^k\}_k, \mathbf{E})$, the other ones being fixed. We perform a cyclic BCD, i.e., we start by solving (3.1) in $\mathbf{Q}^1$, then in $\mathbf{Q}^2$, and continue to solve for each block in $\mathbf{x}$ until we reach the block $\mathbf{E}$; at that point, we repeat the scheme [5]. However, in BCD FB, each subproblem is not fully minimized; only a single step of FB is performed [5, 14, 19]. The corresponding algorithm is shown in Algorithm 3.2. For the sake of clarity, the gradient and the proximal steps in the update of $\mathbf{Q}^k$ are separated. Note that this algorithm is exactly the proximal alternating linearized minimization (PALM) algorithm constructed in [14]. The involved gradients and proximal operators are the same as the ones in section 3.2, where we compute a closed-form expression for each of them.

Convergence of BCD FB. The study of the cyclic BCD FB algorithm in a nonconvex setting has been conducted in [14]. Similar to the joint case, the convergence to a critical point is theoretically guaranteed when F satisfies the KL inequality. Moreover, in the BCD scheme, the assumption that f is gradient β-Lipschitz is relaxed: only the partial gradients of f need to be β_k-Lipschitz. However, these Lipschitz constants need to be upper- and lower-bounded for each step in the sequence of iterates and

Algorithm 3.2. Pseudo-code of the BCD FB to solve (3.1).

Input: Snapshot matrix $\mathbf{Q} \in \mathbb{R}^{m\times n}$, Transformations $\{\mathcal{T}^k\}_k$.
Input: Initial values $\mathbf{Q}^{1,(0)},\dots,\mathbf{Q}^{K,(0)},\mathbf{E}^{(0)}$, Stepsizes: $(\forall k \in [\![1, K+1]\!])\, \alpha_k \in]0, 2/\beta_k[$.
Output: Estimates of $\mathbf{Q}^1,\dots,\mathbf{Q}^K,\mathbf{E}$.

1: Initialize t to 0.
2: **repeat**
3: Perform the FB step for each block of problem (3.1):
4: **for** Variable block $\mathbf{Q}^{1,(t)}$ **to** Variable block $\mathbf{Q}^{K,(t)}$ **do**
5: Update the block $\mathbf{Q}^k$ in the optimization vector

$$\mathbf{G}^{k,(t)} \leftarrow \mathbf{Q}^{k,(t)} - \alpha_k \nabla_{\mathbf{Q}^k} f(\mathbf{Q}^{1,(t+1)},\dots,\mathbf{Q}^{k-1,(t+1)},\mathbf{Q}^{k,(t)},\dots,\mathbf{Q}^{K,(t)})$$
$$\mathbf{Q}^{k,(t+1)} \leftarrow \operatorname{prox}_{\alpha_k\lambda_k\|\cdot\|_*}(\mathbf{G}^{k,(t)})\,.$$

6: **end for**
7: Update the last block $\mathbf{E}^{(t)}$

$$\mathbf{E}^{(t+1)} \leftarrow \operatorname{prox}_{\alpha_{K+1}\lambda_{K+1}\|\cdot\|_1}(\mathbf{E}^{(t)} - \alpha_{K+1}\nabla_{\mathbf{E}} f(\mathbf{Q}^{1,(t+1)},\dots,\mathbf{Q}^{K,t+1},\mathbf{E}^{(t)}))\,.$$

8: Increment t.
9: **until** stopping criterion is met.
10: **return** $\{\mathbf{Q}^{k,(t)}\}_k, \mathbf{E}^{(t)}$

for each block (see [14, Assumption 2]). Similarly to its joint version, the BCD FB algorithm satisfies to a descent lemma, and thus, the objective function is guaranteed to decrease after each iteration.

3.4. Constrained formulation. Inspired by the work [62], we formulate (2.9) as the following constrained optimization:

$$\begin{cases} \underset{\{\mathbf{Q}^k\}_k,\mathbf{E}}{\text{minimize}} & \mathcal{J}_1\left(\{\mathbf{Q}^k\}_k,\mathbf{E}\right) \overset{\text{def}}{=} \displaystyle\sum_{k=1}^{K} \lambda_k \left\|\mathbf{Q}^k\right\|_* + \lambda_{K+1}\|\mathbf{E}\|_1 \\ \text{s.t.} & \mathbf{Q} = \displaystyle\sum_{k=1}^{K} \mathcal{T}^k(\mathbf{Q}^k) + \mathbf{E}\,, \end{cases} \tag{3.6}$$

where the minimization of the objection function $\mathcal{J}_1$ promotes low-rank co-moving fields while the constraint ensures that the latter generates a good approximation of $\tilde{\mathbf{Q}}$. A standard method to solve problem (3.6) is the augmented Lagrangian method (ALM) [9]. It consists in the unconstrained minimization of the augmented Lagrangian $\mathcal{L}_\mu$ related to problem (3.6),

$$\begin{aligned} &\mathcal{L}_\mu(\{\mathbf{Q}^k\}_k,\mathbf{E},\mathbf{Y}) \\ &\quad = \mathcal{J}_1\left(\{\mathbf{Q}^k\}_k,\mathbf{E}\right) + \left\langle \mathbf{Y} \,\middle|\, \mathbf{Q} - \sum_{k=1}^{K} \mathcal{T}^k(\mathbf{Q}^k) - \mathbf{E} \right\rangle \\ &\qquad + \frac{\mu}{2}\left\| \mathbf{Q} - \sum_{k=1}^{K} \mathcal{T}^k(\mathbf{Q}^k) - \mathbf{E} \right\|_{\mathrm{F}}^2, \end{aligned} \tag{3.7}$$

Algorithm 3.3. Pseudo-code of the ALM to solve (3.6).

Input: Snapshot matrix $\mathbf{Q} \in \mathbb{R}^{m\times n}$, Transformations $\{\mathcal{T}^k\}_k$.
Input: Initial values $\mathbf{Q}^{1,(0)}, \dots, \mathbf{Q}^{K,(0)}, \mathbf{E}^{(0)}$, Parameter $\mu > 0$.
Output: Estimates of $\mathbf{Q}^1, \dots, \mathbf{Q}^K, \mathbf{E}$.

Initialize t to 0.
Initialize the dual variable $\mathbf{Y}^{(t)}$ to $\mathbf{0}$.
repeat
  for Frame $k = 1, \dots, K$ **do**
    Compute the residual: $\mathbf{R} \leftarrow \mathbf{Q} - \sum_{l=1}^{k-1} \mathcal{T}^l(\mathbf{Q}^{l,(t+1)}) - \sum_{l=k+1}^{K} \mathcal{T}^l(\mathbf{Q}^{l,(t)}) - \mathbf{E}^{(t)}$
    Perform gradient step: $\mathbf{Q}^{k,(t+1)} \leftarrow \mathcal{T}^{-k}(\mathbf{R} + \mu^{-1}\mathbf{Y}^{(t)})$
    Perform proximal step: $\mathbf{Q}^{k,(t+1)} \leftarrow \mathrm{SVT}(\mathbf{Q}^{k,(t+1)}, \mu^{-1}\lambda_k)$
  end for
  Perform gradient step: $\mathbf{E}^{(t+1)} \leftarrow \mathbf{E}^{(t)} + \mu^{-1}(\mathbf{Q} - \sum_{k=1}^{K} \mathcal{T}^k(\mathbf{Q}^{k,(t+1)}) - \mathbf{E}^{(t)}) + \mathbf{Y}^{(t)}$
  Perform proximal step: $\mathbf{E}^{(t+1)} \leftarrow \mathrm{Soft}_{[-\mu^{-1}\lambda_{K+1}, \mu^{-1}\lambda_{K+1}]}(\mathbf{E}^{(t+1)})$
  Perform gradient ascent: $\mathbf{Y}^{(t+1)} \leftarrow \mathbf{Y}^{(t)} + \mu(\mathbf{Q} - \sum_k \mathcal{T}^k(\mathbf{Q}^{k,(t+1)}) - \mathbf{E}^{(t+1)})$
until stopping criterion is met.
return $\{\mathbf{Q}^{k,(t+1)}\}_k, \mathbf{E}^{(t)}$

where $\mathbf{Y}$ is an $M \times N$ real matrix corresponding to the Lagrange multipliers and μ is a strictly positive real parameter.

Note that (3.7) has a form similar to (3.2): it is the sum of a convex lower semicontinuous separable term $\mathcal{J}_1$ and a smooth term. Therefore, the minimization of $\mathcal{L}_\mu$ can be performed like in section 3.3, with a cyclic BCD where an FB step is performed for each block, a.k.a. PALM algorithm [14]. The step sizes are set to μ^{-1} following [39]. Then, the Lagrangian multiplier is updated using a gradient ascent. The obtained algorithm is displayed in Algorithm 3.3.

Convergence of ALM. Although Algorithm 3.3 looks like the alternative direction method of multipliers (ADMM) [15], it is not an ADMM because of the nonlinearity of the transport operators. To the best of our knowledge, there are no theoretical guarantees about the convergence of Algorithm 3.3, even if some recent works extended ADMM for some nonconvex settings [25, 66] and for substituting a linear operator with a multilinear one in the constraint [56].

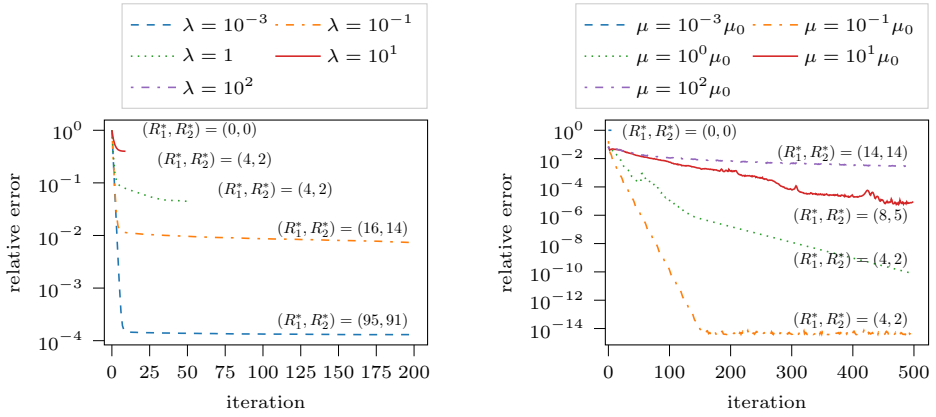
4. Experimental results. In this section, we refer to Algorithm 3.1 as JFB, to Algorithm 3.2 as BFB, and to Algorithm 3.3 as ALM. All the simulations presented in this section have been conducted with implementations in Python.[1] We compare them with the $\mathcal{J}_2$ method derived in [62] and the multishift and reduce method $\mathcal{J}_3$ used in [10, 63]. For every experiment, we use the same initialization for all the different algorithms: we set matrices $\{\mathbf{Q}^k\}$ and $\mathbf{E}$ to $\mathbf{0}$, the matrix composed solely of 0.

Stopping criterion. The following stopping criterion is used for JFB and BFB:

$$F(\mathbf{x}^{(t)}) - F(\mathbf{x}^{(t+1)}) \leq \delta F(\mathbf{x}^{(t)}), \tag{4.1}$$

where δ is a tolerance set to 10^{-5}, $\mathbf{x}^{(t)}$ is the previous iterate, and $\mathbf{x}^{(t+1)}$ is the current one. If the convergence is not reached after $5,000$ iterations, we stop the algorithm

[1] Source code is available at https://zenodo.org/doi/10.5281/zenodo.13366119.



(a) Influence of λ on BFB. (b) Influence of μ on ALM.

FIG. 2. *Impact of the hyperparameters on the relative reconstruction error at each iteration on the multilinear transport test case. The co-moving ranks* $R_k^* = \text{rank}(\mathbf{Q}^k)$ $k = 1, 2$ *at iteration* 500 *are stated for each hyperparameter at the end of each line.*

and return the current estimated co-moving fields and residual error. The stopping criterion for ALM is similar to (4.1), but we use $\mathcal{E}(\mathbf{x}) = \frac{\|\mathbf{Q} - \sum_k \mathcal{T}^k(\mathbf{Q}^{k,(t)}) - \mathbf{E}^{(t)}\|_F}{\|\mathbf{Q}\|_F}$ instead of F and set the maximum number of iterations to 500.

Implementation. For FB methods, it is difficult to find an analytic expression for the Lipschitz constants β and $\{\beta_k\}$, which depend on $\{\mathcal{T}^k\}$. Hence, we use a stepsize $\alpha = 1/K$ in all test cases, which is small enough to obtain convergence. Note that the number of frames K is a priori known since we assume we know the transformations $\mathcal{T}^k$. The parameters $\{\lambda_k\}_{k \in [\![1,K]\!]}$ are all set to the same value λ: we have no a priori information to promote more low-rank estimates for some of the co-moving frames $\{\mathbf{Q}^k\}_k$ than the others in our test cases. Choosing a higher value of λ favors lower-rank factors $\mathbf{Q}^k$, whereas a lower value promotes a low reconstruction error. The value used in our test case is determined empirically: we test several values and select the one that yields the best results. The parameter μ in ALM has the opposite behavior, and its value is set around $\mu_0 = MK/(4\|\mathbf{Q}\|_1)$, similar to [18]. The impact of these parameters is displayed in Figure 2 for the test case from section 4.1.1. The complexity per iteration of all algorithms scales the same and is dominated by the SVD of the co-moving fields performed in the proximal operator. It can be decreased using randomization [28] or wavelet techniques [41]. In the supplementary material section SM2, we present additional performance tests and a complexity study of our algorithms. The scaling behavior with respect to M and N is investigated: we observe that for representative examples, the complexity scales linearly as $\mathcal{O}(M)$ in the space dimension M and as $\mathcal{O}(N^{1.4})$ with the number of snapshots N. The complexity in N can be further reduced to $\mathcal{O}(N)$ using a randomized SVD.

Performance evaluation. We compare the different algorithms with the following criteria: their computational efficiency (CPU time) along with the number of iterations before convergence, the ranks R_k of the estimated co-moving fields $\{\mathbf{Q}^k\}_{k \in [\![1,K]\!]}$, and the relative reconstruction error defined as $\|\mathbf{Q} - \sum_{k=1}^K \mathcal{T}^k(\mathbf{Q}^k) - \mathbf{E}\|_F / \|\mathbf{Q}\|_F$. Table 1 summarizes the performance comparison of the three proposed methods for the test cases described below.

TABLE 1

Performance comparison of the algorithms designed to solve problem (2.9) *on our test cases. The lowest relative error and CPU times are highlighted in bold. $\mathcal{J}_2$ and $\mathcal{J}_3$ cannot estimate the ranks, the values indicated the input ranks. The cross means the original code cannot handle the data.*

	JFB	BFB	ALM	$\mathcal{J}_2$	$\mathcal{J}_3$
Relative error					
Multilinear transport	1.42e-02	1.41e-02	**1.9e-05**	1.37e-03	6.4e-02
Sine waves	1.43e-02	7.96e-01	**1.3e-04**	1.45e-02	8.1e-01
Wildland fire (temperature)	9.1e-02	9.5e-02	**2.1e-02**	4.75e-02	×
Two cylinders wake flow u_1	1.58e-03	1.58e-03	1.02e-02	**8.43e-03**	×
Two cylinders wake flow u_2	1.1e-02	1.1e-02	1.62e-02	**1.41e-02**	×
Estimated ranks					
Multilinear transport	(4,2)	(4,2)	(4,2)	(4,2)	(4,2)
Sine waves	(4,1)	(4,1)	(4,1)	(4,1)	(4,1)
Wildland fire (temperature)	(4,10)	(5,9)	(10,8)	(10,8)	×
Two cylinders wake flow u_1	(203,228)	(209,221)	(40,31)	(40,31)	×
Two cylinders wake flow u_2	(221,251)	(221,251)	(119,128)	(119,128)	×
CPU time					
Multilinear transport	27s	**16s**	9s	28s	2s
Sine waves	75s	89s	14s	**3s**	1s
Wildland fire (temperature)	243s	331s	201s	**70s**	×
Two cylinders wake flow u_1	12h	14h	34h	**7h**	×
Two cylinders wake flow u_2	29h	29h	34h	**7h**	×
Number of iterations					
Multilinear transport	221	174	104	500	1000
Sine waves	961	777	145	71	1000
Wildland fire (temperature)	15	18	7	6	×
Two cylinders wake flow u_1	612	553	500	500	×
Two cylinders wake flow u_2	1500	1500	500	500	×

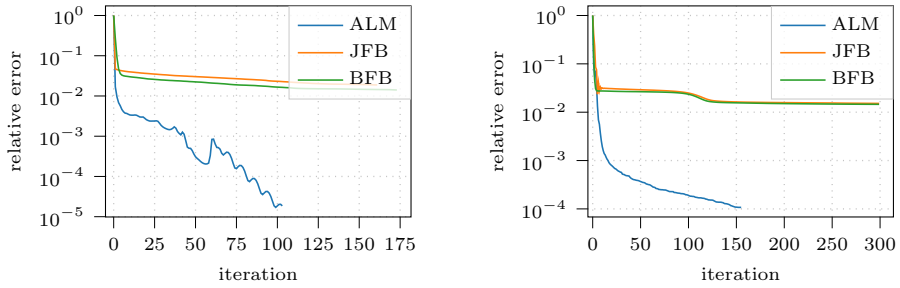
4.1. Validation on synthetic data. We first test our algorithms on synthetic data for which an exact decomposition of the snapshot matrix $\mathbf{Q}$ is known in order to numerically validate our approach.

4.1.1. Multilinear transport. This example illustrates that, in a noiseless context, our algorithms are able to retrieve a low-rank decomposition with a low reconstruction error and the correct ranks. To this end, we generate a snapshot matrix $\mathbf{Q}$ of dimensions 400×200 by discretizing the following transport-dominated field q composed of two co-moving structures of ranks $(R_1, R_2) = (4, 2)$:

$$q(x,t) = \sum_{r=1}^{R_1} \sin(rt\pi) h(x + \Delta_1(t) - 0.1r) + \sum_{r=1}^{R_2} \cos(rt\pi) h(x + \Delta_2(t) - 0.1r)\,.$$

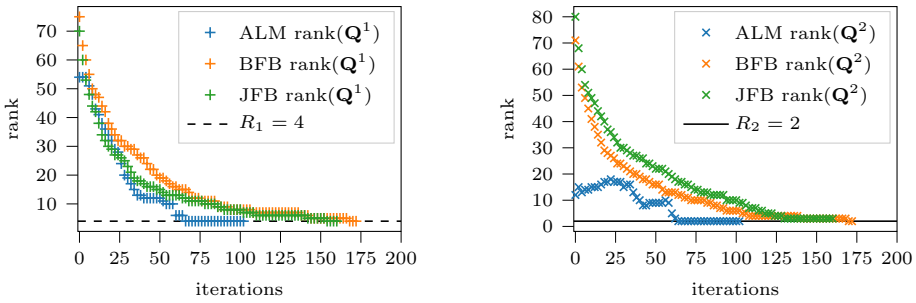
The initial spatial profile of the waves in each co-moving frame is given by $h(x) = \exp(-x^2/\delta^2)$, where δ is set to 0.0125 and the shift (Δ_1, Δ_2) to $(t, -t)$. The discretization grid is obtained by uniformly discretizing the set $[-0.5, 0.5] \times [0, 0.5]$. Hence, after the shift transformations, the data fit on the grid and do not cause any interpolation error. Moreover, the data matrix $\mathbf{Q}$ is also free from any noise in the data. Consequently, $\mathbf{E} = \mathbf{0}$ in model (2.8).

Now, we apply the FB algorithms to decompose $\mathbf{Q}$ with $\lambda = 0.3$ and $\lambda_{K+1} = 0$, as well as ALM with $\lambda = 1$, $\lambda_{K+1} = 0$, and $\mu = MN_t/(4||\mathbf{Q}||_1)$. In Figure 3(a), we plot the relative error at each iteration of the algorithms, while in Figure 4, we plot the evolution of the estimated co-moving ranks (R_1^*, R_2^*). We first observe the descent



(a) Multilinear transport test case. (b) Sine waves with noise test case.

FIG. 3. *Decay of the relative approximation error in the Frobenius norm.*



(a) Rank evolution of $\mathbf{Q}^1$. (b) Rank evolution of $\mathbf{Q}^2$.

FIG. 4. *Ranks of the estimated co-moving fields at each iteration for the multilinear transport test case.*

property of FB methods, while ALM misses such a property. We then remark that all the methods retrieve the correct ranks. We also note that, although ALM reaches the maximum number of iterations, it has a better accuracy than the two FB methods. Moreover, Table 1 shows that $\mathcal{J}_2$ reaches a machine precision relative error for the decomposition. However, in contrast with our proposed methods, $\mathcal{J}_2$ suffers from two drawbacks:

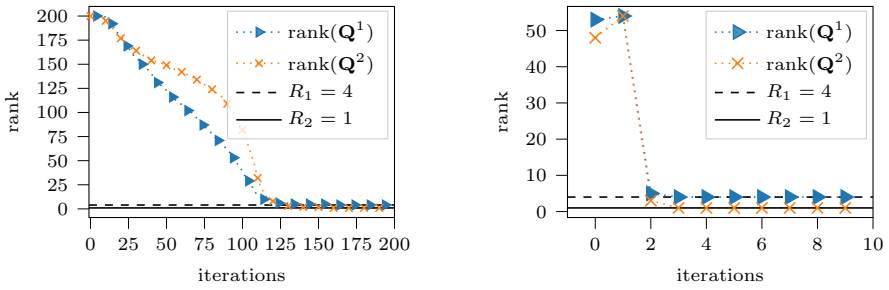
(i) it requires to know explicitly the correct ranks, which is a challenging impediment on real data;

(ii) it performs worse in the presence of noise, as we will see in section 4.1.2.

Another advantage of the new formulation is that the nuclear norm removes any frame that does not lead to a low-rank description. In the supplementary material section SM1, we study the behavior of ALM when an additional frame with the shift $\Delta_3(t) = t^2$ is added to the decomposition. As this shift does not describe any transport present in the system, the additional co-moving frame $\mathbf{Q}^3$ converges to $\mathbf{0}$ along the iterations.

4.1.2. Sine waves with noise. We now evaluate our methods in a noisy context. Similarly to the previous section, we generate a snapshot matrix $\mathbf{Q}$ with dimensions 400×200 by discretizing the following field q composed of two co-moving structures with ranks $(R_1, R_2) = (4, 1)$:

$$q(x,t) = \sum_{r=1}^{R_1} \sin(4\pi r t)\, h(x - 0.1 - 0.25 + \Delta_1(t)) + h(x - 0.2 - \Delta_2(t))\,,$$



(a) BFB ($\lambda = 0.3$, $\lambda_{K+1} = 0.0135$). (b) ALM.

FIG. 5. *Ranks of the estimated co-moving fields at each iteration for the sine waves test case with noise.*

where $h(x) = \exp(-x^2/\delta^2)$, $\delta = 0.0125$, $(\Delta_1, \Delta_2) = (0.25\cos(7\pi t), -t)$, and the discretization grid is a uniform lattice on $[0, 0.5] \times [0, 1]$. In contrast to the previous example, the transformations $\mathcal{T}^1$ and $\mathcal{T}^2$ now introduce interpolation errors to the data stored in $\mathbf{Q}$. Furthermore, we add a salt-and-pepper noise on the data $\mathbf{Q}$ by setting 12.5% of its elements to 1. The indices of the noisy data are drawn randomly from a discrete uniform distribution. An illustration of the snapshot matrices and their sPOD decomposition was given in Figure 1.

We run our three algorithms with the parameters $\lambda = 0.3$ and $\lambda_{K+1} = 0.0135$ for FB methods and $\lambda = 1$, $\lambda_{K+1} = 1/\sqrt{\min(M, K)}$ and $\mu = \mu_0/10$ for ALM. Since $\mathcal{J}_2$ and $\mathcal{J}_3$ are not able to estimate the ranks for noisy data, we give them the correct ones to be able to conduct a comparison. Nonetheless, this is a severe restriction compared to our proximal methods. Figures 3(b) and 5 respectively show the relative error and the estimated ranks as a function of the iterations. We observe that even in the presence of noise, our three methods estimate the correct ranks. Furthermore, ALM shows the lowest relative error as well as the lowest running time.

4.2. 2D wildland-fire model. We now test our algorithms on the 2D wildland fire model given in [17], where authors use this model to assess the validity and the performance of their neural network-based sPOD. The model consists of two coupled reaction-diffusion equations: one describes the evolution of the temperature, and the other one describes the evolution of the fuel supply mass fraction. We use similar model parameters as in [17] and showcase only the results with respect to the temperature. However, similar statements hold when the supply mass fraction is included (see [17]). The differential equations are discretized using a 500×500 equally spaced grid on the domain $[0, 500]^2$ and integrated up to time $T_{\text{end}} = 900$ for a reaction rate $\mu = 558.49$ and wind velocity $v = 0.2$. We then generate the snapshot matrix of the temperature with 100 equally spaced snapshots.[2]

A selected snapshot of the temperature profile is shown in Figure 6. In this simulation, a fire starts as an initial ignition with a Gaussian distribution in the center of the domain. Thereafter, a reaction wave spreads from the middle of the domain to the right, induced by a wind force. The ignition and traveling wave can be decomposed into a stationary frame and a frame that captures the traveling reaction wave.

To separate the frames, we apply the transformations outlined in [17]. In this example, the shift now depends not only on time but also on space. This spatial

[2]Data is available for download at https://doi.org/10.5281/zenodo.13355796.

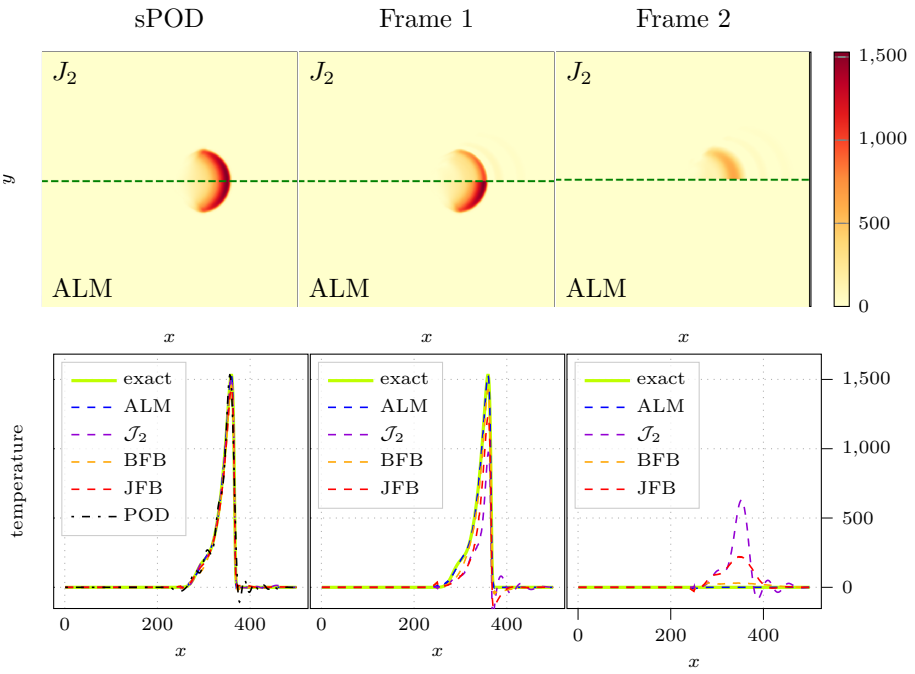


FIG. 6. *Decomposition results for the 2D wildland-fire test case with wind at $t = 50$. The first row shows the sPOD approximation of the temperature and its decomposition into the individual frames using the J_2 (upper half) and ALM (lower half) algorithm. The second row shows a profile plot of all algorithms along the horizontal line at $y = 250$ and the results of the POD with rank $R = 18$. The ranks and approximation quality of all other algorithms are listed in Table 1.*

dependency is modeled using a low-rank parameterization, as detailed in [17]. After configuring the shift transformations, we execute the proximal sPOD algorithms and assess their performance. We set $\lambda = 2200$ and $\lambda_{K+1} = 0$ for decomposing the temperature snapshot matrix using the FB algorithms. The results are visualized for one snapshot in Figure 6 and quantified in Table 1. First, we observe that all sPOD algorithms approximate the data without the typical oscillatory effects induced by the POD (lower left profile picture). Furthermore, the direct comparison in Figure 6 shows that the J_2 algorithm is not able to separate the traveling reaction wave from the initial ignition impulse. In contrast, the proximal methods provide a better separation, whereas the ALM algorithm shows the best results. The noise part captured in E is not shown in our examples as it only contains small interpolation errors.

4.3. Two cylinder wake flow. Lastly, we study the incompressible flow around two cylinders simulated with the open source software `WABBIT` [21]. The setup is visualized in Figure 7 and is inspired by biolocomotion, where the leader is followed by a chaser in a free-stream flow of uniform velocity u_∞. In biolocomotion, the interaction between animals in close proximity, like fish or birds, is studied to understand their swarm behavior [29, 72]. In particular, one tries to explain swarm behavior with potential physics reasons, like energy minimization, or biological reasons, such as breeding or defense. To study a swarm from an idealist fluid dynamic perspective, a leading cylinder with a diameter l is placed at a fixed position $(x_1, y_1) = (L/4, L/2)$ in a uniform flow at Reynolds number $\mathrm{Re} = u_\infty l/\nu = 200$, and the chaser further downstream $(x_2, y_2) = (L/2, L/2 + \Delta_{\mathrm{cyl}}(t))$ is shifted along a vertical path $\Delta_{\mathrm{cyl}}(t)$

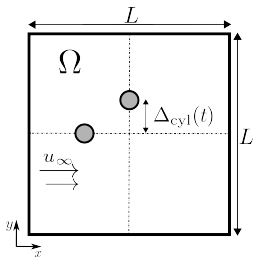


FIG. 7. *Illustration of the computational set-up of the two-cylinder simulation, one fixed, one vertically moving. The flow is driven by the inflow u_∞, which is indicated by the arrows. Within the computational domain $\Omega = [0, L]^2$, the flow passes the two cylinders colored gray and generates a vortex shedding. The fixed cylinder is located at $(x, y) = (L/4, L/2)$, and the moving cylinder is at $(x_2, y_2) = (L/2, L/2 + \Delta_{\text{cyl}}(t))$.*

TABLE 2
Physical parameters of the two-cylinder simulation.

Name	Value	Name	Value
Simulation time	$T_{\text{end}} = 2000$	Domain size	$[0, 64] \times [0, 64]$
Inflow velocity	$u_\infty = 1$	Reynolds number	$\text{Re} = 200$
Cylinder diameter	$l = 2$	Viscosity	$\nu = 10^{-2}$

that is time-dependent. The vortex shedding generated by the first cylinder impacts the drag and lift forces of the second cylinder. To study the interactions between the two systems, they need to be separated, for example, using the proposed proximal algorithms.

The snapshot set for the example is built from the trajectory corresponding to the path $\Delta_{\text{cyl}}(t) = 16 \sin(2\pi f_1 t)$ with $f_1 = 10 f_{\text{wake}} = 0.2 \times 10^{-2} s^{-1}$. Here, f_{wake} was calculated from the Strouhal number $\text{St} = l f_{\text{wake}}/u_\infty$ of the leading cylinder. The data[3] are sampled with $\Delta(t) = 1$ in time, resulting in 500 snapshots. We sample a full period $T = 500 = 1/f_1$ of the movement in an interval $[3T, 4T]$. After the simulation, we upsample the adaptive grid to a uniform 512×512 grid. All physically relevant parameters like properties of the fluid are listed in Table 2. Further information about the simulation can be found in [39].

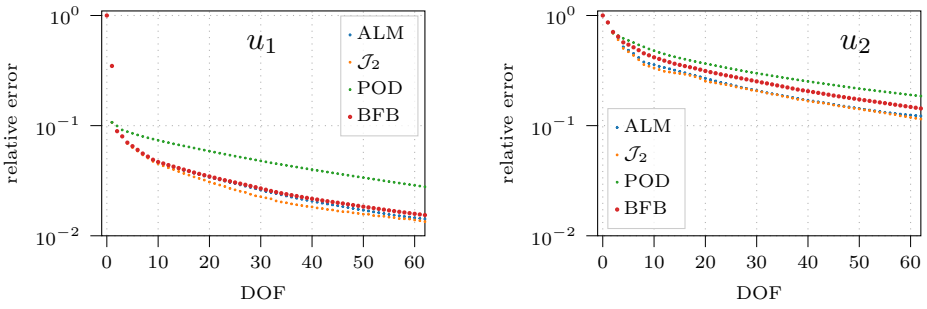
To reduce the data using the sPOD, we introduce the shift transformations. For the leading cylinder, a shift transformation is not needed since the cylinder is stationary ($\mathcal{T}^1 = \text{Id}$). For the second one, we introduce the shift transformation

$$\mathcal{T}^2(q)(x, y, t) = q(x, y + \Delta_{\text{cyl}}(t), t)\,, \tag{4.2}$$

which accounts for the movement of the cylinder and its vortex shedding.

Note that with the utilized mappings both cylinders are stationary in their corresponding frames and similarly in their vortex shedding. Hence, the structures are no longer transport-dominated, and therefore a better decomposition can be achieved. This is in contrast with purely Lagrangian methods like [1, 50, 53, 70, 76]. Here, a single one-to-one mapping of the domain onto a reference mesh is used to compensate for the transport. However, even if the two cylinders are stationary in this reference mesh, a strong oscillation of the vortex shedding could not be avoided, since the two vortex sheddings cross. This explains the necessity of a multiframe approximation for a separation of the two phenomena.

[3]Data is available for download at https://doi.org/10.5281/zenodo.13355796.



(a) Horizontal velocity component.

(b) Vertical velocity component.

FIG. 8. *Relative reconstruction error of the velocity fields (u_1, u_2) in the Frobenius norm versus the DOFs. The DOFs are determined from the truncation rank R of the POD and as the sum of the co-moving ranks $R = R_1 + R_2$ in the case of $\mathcal{J}_2$ and the proximal algorithms, respectively.*

With the imposed shifts, we apply the proximal algorithms separately to the individual velocity components $q = u_1, u_2$ of the PDE solution. We apply the following strategies to compare the algorithms:

(i) We run the proximal algorithms until they reach the stopping criterion to obtain two co-moving fields $\{q^k\}_{k=1,2}$ with their corresponding truncation ranks $\{R_k^*\}_{k=1,2}$.

(ii) We truncate the $\{q^k\}_{k=1,2}$ for all possible rank combinations $(R_1, R_2) \in [\![1, R_1^*]\!] \times [\![1, R_2^*]\!]$ and select the pairs (R_1, R_2) for which the ROM with R DOFs has the smallest truncation error.

(iii) We run the $\mathcal{J}_2$ algorithm on the exact same pairs (R_1, R_2) determined from the previous step.

The comparison of the resulting approximation errors for all algorithms can be seen in Figure 8. In the implementation of ALM, we set the parameters as follows: the initial value μ_0 is defined as $MN_t/(4||\mathbf{Q}||_1)$, and μ is set to $5.0 \times 10^{-3}\mu_0$ for u_1 and to $4.0 \times 10^{-4}\mu_0$ for u_2, respectively. For both ALM and FB, we configure the parameters with $\lambda = 1$ and $\lambda_{K+1} = 0$.

As shown in Figure 8, the approximation errors are similar for all algorithms. The results of sPOD algorithms are superior to the results of the POD. Additionally, it should be pointed out that, in contrast to the proximal algorithms, the $\mathcal{J}_2$ algorithm requires a separate run of the algorithm for every data point shown in Figure 8. Indeed, $\mathcal{J}_2$ optimizes q^1, q^2 only for a fixed rank. However, this does not imply that the optimized co-moving fields have a fast singular value decay. As a consequence, $\mathcal{J}_2$ is not able to separate the two cylinders well. This is shown in Figure 9, which displays the two dimensional vorticity field $\omega(x, y) = \partial_x u_2(x, y) - \partial_y u_1(x, y)$ resulting from the sPOD approximation of the velocity field (u_1, u_2). A video of the decomposed flow field is presented in the supplementary material section SM3.

Besides the better approximation quality compared to the POD, we highlight the two important implications of this example, which could not have been achieved previously.

(i) Since proximal algorithms are capable of separating the flow of the two systems, one can build a surrogate model for the individual systems that includes the path as a reduced variable. Therefore, it may be used to optimize the path of the second cylinder regarding drag or lift.

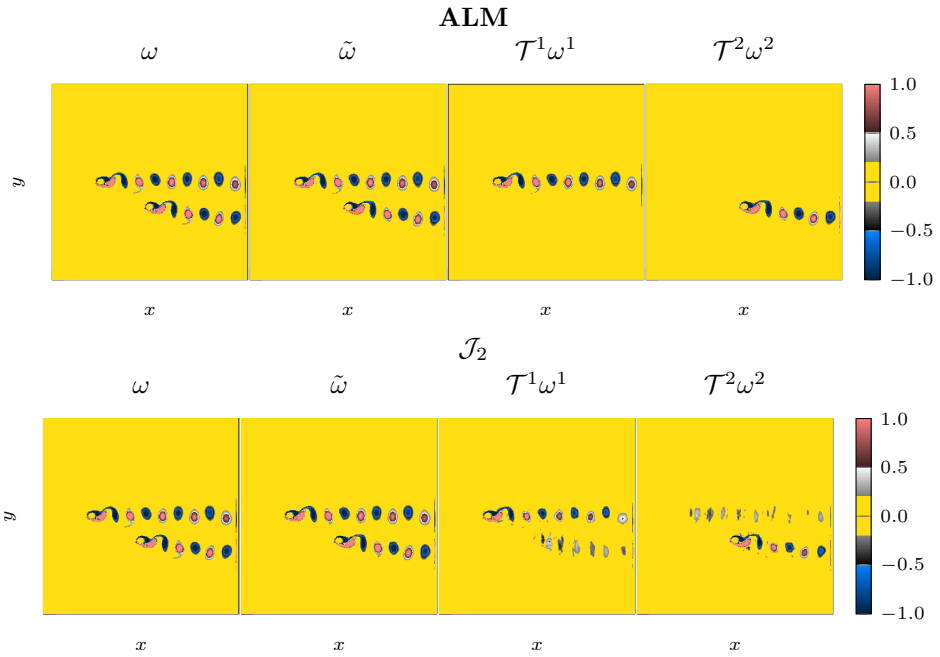


FIG. 9. *Separation of the two moving cylinders. The vorticity field* $\omega = \partial_x u_2 - \partial_y u_1$ *is displayed. It has been computed from the coupled cylinder pair and the reconstructed vorticity field* $\tilde{\omega} = \omega^1 + \omega^2$, *with* $\omega^i = \partial_x \mathcal{T}^i u_2^i - \partial_y \mathcal{T}^i u_1^i$, $i = 1, 2$. *The co-moving ranks that are estimated by ALM and used as an input in* $\mathcal{J}_2$ *are* $(R_1, R_2) = (40, 31)$ *for* (u_1^1, u_1^2) *and* $(R_1, R_2) = (119, 128)$ *for* (u_2^1, u_2^2). *Separation similar to ALM is obtained with both FB algorithms (not shown here).*

(ii) The example allows us to identify structures that can be attributed to the free-stream flow and structures that are responsible for the interaction. A first study in this direction can be found in [39].

4.4. Discussion. Table 3 compares and summarizes the advantages of the three methods we propose. In particular, we note that FB algorithms have the desirable descent property, which ensures that each iteration yields a better minimizer. Moreover, they also have theoretical guarantees that they converge to a critical point. In contrast, ALM performs best experimentally but does not have these two important theoretical properties. We observe in Table 1 that, although ALM incurs higher costs per iteration, it generally requires fewer iterations to achieve convergence, making it more cost-effective overall. However, the algorithm is more computationally expensive than the POD method, as each iteration requires performing multiple SVDs. Consequently, the offline costs for the decomposition are relatively high. For instance, according to [17], the CPU time $t_{\text{FOM}} = 35.7\,\text{s}$ is required for the creation/simulation of the 2D wildland fire data. This is approximately six times less CPU time compared to the ALM decomposition performed on a similar architecture.

5. Conclusion. We have presented three proximal algorithms to extend sPOD for transport-dominated flows with multiple transports using a decomposition into co-moving linear subspaces. FB methods own enjoyable theoretical properties such as a descent lemma and convergence even in a nonconvex setting, such as sPOD, while ALM demonstrates the best numerical results. Furthermore, we have shown that our methods can estimate the correct ranks of the different components of the sPOD.

Table 3
Benefits of the three proposed proximal methods.

	JFB	BFB	ALM
Guarantee of convergence	Yes	Yes	No
Descent lemma	Yes	Yes	No
Experimental accuracy	Low	Low	High

In contrast to existing approaches, our methods are robust to noise. The numerical results show an accurate and strict separation of the involved transport phenomena. The close connection of our algorithms to the POD in combination with the strict separation opens a new paradigm for the optimization, control, and analysis of flows. A promising topic for future research would be the development of methods that can estimate both the co-moving structures and the associated transformation operators during the optimization phase. A first step in this direction can be found in [77].

Credit authorship contribution statement. In Table 4, we declare the authors' contributions to this work.

Table 4

P. Krah:	Conceptualization, Methodology, Software, Writing - original draft.
A. Marmin:	Methodology, Formal Analysis, Software, Writing - original draft.
B. Zorawski:	Software, Visualization, Writing - review and editing.
J. Reiss:	Writing - review & editing, Funding acquisition.
K. Schneider:	Writing - review & editing, Funding acquisition, Project administration.

Acknowledgment. The authors acknowledge Shubhaditya Burela and Thomas Engels for providing the wildland fire and the two-cylinder data, respectively. Furthermore, we thank Martin Isoz and Anna Kovarnova for discussion related to this work.

REFERENCES

[1] M. Alireza Mirhoseini and M. J. Zahr, *Model reduction of convection-dominated partial differential equations via optimization-based implicit feature tracking*, J. Comput. Phys., 473 (2023), 111739, https://doi.org/10.1016/j.jcp.2022.111739.

[2] F. Arbes, C. Greif, and K. Urban, *The Kolmogorov N-Width for Linear Transport: Exact Representation and the Influence of the Data*, preprint, https://arxiv.org/abs/2305.00066, 2023.

[3] H. Attouch, J. Bolte, and B. F. Svaiter, *Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods*, Math. Program., 137 (2011), pp. 91–129, https://doi.org/10.1007/s10107-011-0484-9.

[4] H. H. Bauschke and P. L. Combettes, *Fenchel–Rockafellar duality,* in Convex Analysis and Monotone Operator Theory, CMS Books in Mathematics, Springer New York, 2011, pp. 207–222, https://doi.org/10.1007/978-1-4419-9467-7_15.

[5] A. Beck, *First-Order Methods in Optimization*, MOS-SIAM Ser. Optim. 25, SIAM, Philadelphia, 2017, https://doi.org/10.1137/1.9781611974997.

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

[7] G. Berkooz, P. Holmes, and J. Lumley, *The proper orthogonal decomposition in the analysis of turbulent flows*, Annu. Rev. Fluid Mech., 25 (1993), pp. 539–575, https://doi.org/10.1146/annurev.fl.25.010193.002543.

[8] F. Bernard, A. Iollo, and S. Riffaud, *Reduced-order model for the BGK equation based on POD and optimal transport*, J. Comput. Phys., 373 (2018), pp. 545–570, https://doi.org/10.1016/j.jcp.2018.07.001.
[9] D. P. Bertsekas, *Nonlinear Programming*, 3rd ed., Athena Scientific, Belmont, MA, 2016.
[10] F. Black, P. Schulze, and B. Unger, *Projection-based model reduction with dynamically transformed modes*, ESAIM Math. Model. Numer. Anal., 54 (2020), pp. 2011–2043, https://doi.org/10.1051/m2an/2020046.
[11] F. Black, P. Schulze, and B. Unger, *Efficient wildland fire simulation via nonlinear model order reduction*, Fluids, 6 (2021), 280, https://doi.org/10.3390/fluids6080280.
[12] F. Black, P. Schulze, and B. Unger, *Modal decomposition of flow data via gradient-based transport optimization*, in Active Flow and Combustion Control 2021, Springer, 2021, pp. 203–224, https://doi.org/10.1007/978-3-030-90727-3_13.
[13] J. Bolte, A. Daniilidis, A. Lewis, and M. Shiota, *Clarke subgradients of stratifiable functions*, SIAM J. Optim., 18 (2007), pp. 556–572, https://doi.org/10.1137/060670080.
[14] J. Bolte, S. Sabach, and M. Teboulle, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Math. Program., 146 (2014), pp. 459–494, https://doi.org/10.1007/s10107-013-0701-9.
[15] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2010), pp. 1–122, https://doi.org/10.1561/2200000016.
[16] S. Brivio, S. Fresca, N. R. Franco, and A. Manzoni, *Error estimates for POD-DL-ROMs: A deep learning framework for reduced order modeling of nonlinear parametrized PDEs enhanced by proper orthogonal decomposition*, Adv. Comput. Math., 50 (2024), 33, https://doi.org/10.1007/s10444-024-10110-1.
[17] S. Burela, P. Krah, and J. Reiss, *Parametric Model Order Reduction for a Wildland Fire Model via the Shifted Pod Based Deep Learning Method*, preprint, https://arxiv.org/abs/2304.14872, 2023.
[18] E. J. Candès, X. Li, Y. Ma, and J. Wright, *Robust principal component analysis?*, J. ACM, 58 (2011), pp. 1–37, https://doi.org/10.1145/1970392.1970395.
[19] E. Chouzenoux, J.-C. Pesquet, and A. Repetti, *A block coordinate variable metric forward-backward algorithm*, J. Global Optim., 66 (2016), pp. 457–485, https://doi.org/10.1007/s10898-016-0405-9.
[20] P. L. Combettes and J.-C. Pesquet, *Proximal splitting methods in signal processing*, in Fixed-Point Algorithms for Inverse Problems in Science and Engineering, H. H. Bauschke, R. S. Burachik, P. L. Combettes, V. Elser, D. R. Luke, and H. Wolkowicz, eds., Springer, 2011, pp. 185–212.
[21] T. Engels, K. Schneider, J. Reiss, and M. Farge, *A wavelet-adaptive method for multiscale simulation of turbulent flows in flying insects*, Commun. Comput. Phys., 30 (2021), pp. 1118–1149, https://doi.org/10.4208/cicp.OA-2020-0246.
[22] F. Fedele, O. Abessi, and P. J. Roberts, *Symmetry reduction of turbulent pipe flows*, J. Fluid. Mech., 779 (2015), pp. 390–410, https://doi.org/10.1017/jfm.2015.423.
[23] S. Fresca, L. Dedè, and A. Manzoni, *A comprehensive deep learning-cased approach to reduced order modeling of nonlinear time-dependent parametrized PDEs*, J. Sci. Comput., 87 (2021), 6, https://doi.org/10.1007/s10915-021-01462-7.
[24] S. Fresca and A. Manzoni, *POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition*, Comput. Methods Appl. Mech. Engrg., 388 (2022), 114181, https://doi.org/10.1016/j.cma.2021.114181.
[25] W. Gao, D. Goldfarb, and F. E. Curtis, *ADMM for multiaffine constrained optimization*, Optim. Methods Softw., 35 (2019), pp. 257–303, https://doi.org/10.1080/10556788.2019.1683553.
[26] H. Gowrachari, N. Demo, G. Stabile, and G. Rozza, *Non-intrusive Model Reduction of Advection-Dominated Hyperbolic Problems Using Neural Network Shift Augmented Manifold Transformations*, preprint, https://doi.org/10.48550/ARXIV.2407.18419, 2024.
[27] C. Greif and K. Urban, *Decay of the Kolmogorov N-width for wave problems*, Appl. Math. Lett., 96 (2019), pp. 216–222, https://doi.org/10.1016/j.aml.2019.05.013.
[28] N. Halko, P. G. Martinsson, and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288, https://doi.org/10.1137/090771806.
[29] C. K. Hemelrijk and H. Hildenbrandt, *Schools of fish and flocks of birds: Their shape and internal structure by self-organization*, Interface Focus, 2 (2012), pp. 726–737, https://doi.org/10.1098/rsfs.2012.0025.

[30] J. S. Hesthaven and S. Ubbiali, *Non-intrusive reduced order modeling of nonlinear problems using neural networks*, J. Comput. Phys., 363 (2018), pp. 55–78, https://doi.org/10.1016/j.jcp.2018.02.037.

[31] C. Huang, K. Duraisamy, and C. Merkle, *Challenges in reduced order modeling of reacting flows*, in 2018 Joint Propulsion Conference, American Institute of Aeronautics and Astronautics, 2018, https://doi.org/10.2514/6.2018-4675.

[32] E. N. Karatzas, F. Ballarin, and G. Rozza, *Projection-based reduced order models for a cut finite element method in parametrized domains*, Comput. Math. Appl., 79 (2020), pp. 833–851, https://doi.org/10.1016/j.camwa.2019.08.003.

[33] Y. Kim, Y. Choi, D. Widemann, and T. Zohdi, *A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder*, J. Comput. Phys., 451 (2022), 110841, https://doi.org/10.1016/j.jcp.2021.110841.

[34] O. Koch and C. Lubich, *Dynamical low-rank approximation*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 434–454, https://doi.org/10.1137/050639703.

[35] J. Koellermeier, P. Krah, and J. Kusch, *Macro-micro decomposition for consistent and conservative model order reduction of hyperbolic shallow water moment equations: A study using POD-Galerkin and dynamical low-rank approximation*, Adv. Comput. Math., 50 (2024), 76, https://doi.org/10.1007/s10444-024-10175-y.

[36] J. Koellermeier, P. Krah, J. Reiss, and Z. Schellin, *Model order reduction for the 1D Boltzmann-BGK equation: Identifying intrinsic variables using neural networks*, Microfluid. Nanofluid., 28 (2024), 16, https://doi.org/10.1007/s10404-024-02711-5.

[37] A. Kovárnová and M. Isoz, *Model order reduction for particle-laden flows: Systems with rotations and discrete transport operators*, in Topical Problems of Fluid Mechanics 2023, TPFM, Institute of Thermomechanics of the Czech Academy of Sciences; CTU in Prague Faculty of Mech. Engineering Dept. Tech. Mathematics, 2023, https://doi.org/10.14311/tpfm.2023.014.

[38] A. Kovárnová, P. Krah, J. Reiss, and M. Isoz, *Shifted proper orthogonal decomposition and artificial neural networks for time-continuous reduced order models of transport-dominated systems*, in Topical Problems of Fluid Mechanics 2022, TPFM, Institute of Thermomechanics of the Czech Academy of Sciences, 2022, https://doi.org/10.14311/tpfm.2022.016.

[39] P. Krah, *Non-linear Reduced Order Modeling for Transport Dominated Fluid Systems*, Ph.D thesis, Technische Universität Berlin, 2023, https://doi.org/10.14279/depositonce-16974.

[40] P. Krah, S. Büchholz, M. Häringer, and J. Reiss, *Front transport reduction for complex moving fronts: Nonlinear model reduction for an advection-reaction-diffusion equation with a Kolmogorov-Petrovsky-Piskunov reaction term*, J. Sci. Comput., 96 (2023), 28, https://doi.org/10.1007/s10915-023-02210-9.

[41] P. Krah, T. Engels, K. Schneider, and J. Reiss, *Wavelet adaptive proper orthogonal decomposition for large-scale flow data*, Adv. Comput. Math., 48 (2022), 10, https://doi.org/10.1007/s10444-021-09922-2.

[42] P. Krah, M. Sroka, and J. Reiss, *Model order reduction of combustion processes with complex front dynamics*, in Numerical Mathematics and Advanced Applications ENUMATH 2019, Springer, 2020, pp. 803–811, https://doi.org/10.1007/978-3-030-55874-1_79.

[43] K. Kunisch and S. Volkwein, *Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics*, SIAM J. Numer. Anal., 40 (2002), pp. 492–515, https://doi.org/10.1137/S0036142900382612.

[44] K. Kurdyka, *On gradients of functions definable in o-minimal structures*, Ann. de l'Institut Fourier, 48 (1998), pp. 769–783, https://doi.org/10.5802/aif.1638.

[45] K. Lee and K. T. Carlberg, *Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders*, J. Comput. Phys., 404 (2020), 108973, https://doi.org/10.1016/j.jcp.2019.108973.

[46] Z. Lin, R. Liu, and Z. Su, *Linearized alternating direction method with adaptive penalty for low-rank representation*, in Proc. Ann. Conf. Neur. Inform. Proc. Syst., Vol. 24, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, eds., Curran Associates, 2011, https://proceedings.neurips.cc/paper_files/paper/2011/file/18997733ec258a9fcaf239cc5d53363-Paper.pdf.

[47] J. L. Lumley, *The Structure of Inhomogeneous Turbulence*, A. M. Yaglom and V. I. Tatarski, eds., Nauka, Moscow, 1967, pp. 166–178.

[48] A. Mendible, S. L. Brunton, A. Y. Aravkin, W. Lowrie, and J. N. Kutz, *Dimensionality reduction and reduced-order modeling for traveling wave physics*, Theor. Comput. Fluid Dyn., 34 (2020), pp. 385–400, https://doi.org/10.1007/s00162-020-00529-9.

[49] A. Mendible, J. Koch, H. Langer, S. L. Brunton, and J. N. Kutz, *Data-driven modeling of rotating detonation waves*, Phys. Rev. Fluids., 6 (2021), 050507, https://doi.org/10.1103/PhysRevFluids.6.050507.

[50] R. Mojgani and M. Balajewicz, *Low-rank registration based manifolds for convection-dominated PDEs*, Proceedings of the AAAI Conference on Artificial Intelligence, 35 (2021), pp. 399–407, https://doi.org/10.1609/aaai.v35i1.16116.

[51] J.-J. Moreau, *Proximité et dualité dans un espace hilbertien*, Bull. Soc. Math. France, 93 (1965), pp. 273–299, http://eudml.org/doc/87067.

[52] B. R. Noack, K. Afanasiev, M. Morziński, G. Tadmor, and F. Thiele, *A hierarchy of low-dimensional models for the transient and post-transient cylinder wake*, J. Fluid. Mech., 497 (2003), pp. 335–363, https://doi.org/10.1017/S0022112003006694.

[53] M. Nonino, F. Ballarin, G. Rozza, and Y. Maday, *A reduced basis method by means of transport maps for a fluid-structure interaction problem with slowly decaying Kolmogorov n-width*, Adv. Comput. Sci. Eng., 1 (2019), pp. 36–58, https://doi.org/10.3934/acse.2023002.

[54] K. Oberleithner, M. Sieber, C. Nayeri, C. O. Paschereit, C. Petz, H.-C. Hege, B. R. Noack, and I. Wygnanski, *Three-dimensional coherent structures in a swirling jet undergoing vortex breakdown: Stability analysis and empirical mode construction*, J. Fluid. Mech., 679 (2011), pp. 383–414, https://doi.org/10.1017/jfm.2011.141.

[55] M. Ohlberger and S. Rave, *Reduced basis methods: Success, limitations and future challenges*, in Proceedings of the Conference Algoritmy, 2016, pp. 1–12.

[56] D. Papadimitriou and B. C. Vũ, *An augmented Lagrangian method for nonconvex composite optimization problems with nonlinear constraints*, Optim. Eng., 25 (2024), pp. 1921–1990, https://doi.org/10.1007/s11081-023-09867-z.

[57] D. Papapicco, N. Demo, M. Girfoglio, G. Stabile, and G. Rozza, *The neural network shifted-proper orthogonal decomposition: A machine learning approach for non-linear reduction of hyperbolic equations*, Comput. Methods Appl. Mech. Engrg., 392 (2022), 114687, https://doi.org/10.1016/j.cma.2022.114687.

[58] B. Peherstorfer, *Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling*, SIAM J. Sci. Comput., 42 (2020), pp. A2803–A2836, https://doi.org/10.1137/19M1257275.

[59] B. Peherstorfer, *Breaking the Kolmogorov barrier with nonlinear model reduction*, Not. Amer. Mat. Soc., 69 (2022), 1, https://doi.org/10.1090/noti2475.

[60] B. Peherstorfer and K. Willcox, *Online adaptive model reduction for nonlinear systems via low-rank updates*, SIAM J. Sci. Comput., 37 (2015), pp. A2123–A2150, https://doi.org/10.1137/140989169.

[61] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501, https://doi.org/10.1137/070697835.

[62] J. Reiss, *Optimization-based modal decomposition for systems with multiple transports*, SIAM J. Sci. Comput., 43 (2021), pp. A2079–A2101, https://doi.org/10.1137/20M1322005.

[63] J. Reiss, P. Schulze, J. Sesterhenn, and V. Mehrmann, *The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena*, SIAM J. Sci. Comput., 40 (2018), pp. A1322–A1344, https://doi.org/10.1137/17M1140571.

[64] D. Rim, S. Moe, and R. J. LeVeque, *Transport reversal for model reduction of hyperbolic partial differential equations*, SIAM-ASA J. Uncertain. Quantif., 6 (2018), pp. 118–150, https://doi.org/10.1137/17M1113679.

[65] D. Rim, B. Peherstorfer, and K. T. Mandli, *Manifold approximations via transported subspaces: Model reduction for transport-dominated problems*, SIAM J. Sci. Comput., 45 (2023), pp. A170–A199, https://doi.org/10.1137/20M1316998.

[66] R. T. Rockafellar, *Convergence of augmented Lagrangian methods in extensions beyond nonlinear programming*, Math. Program., 199 (2022), pp. 375–420, https://doi.org/10.1007/s10107-022-01832-5.

[67] C. W. Rowley, I. G. Kevrekidis, J. E. Marsden, and K. Lust, *Reduction and reconstruction for self-similar dynamical systems*, Nonlinearity, 16 (2003), pp. 1257–1275, https://doi.org/10.1088/0951-7715/16/4/304.

[68] M. Salvador, L. Dedè, and A. Manzoni, *Non intrusive reduced order modeling of parametrized PDEs by kernel POD and neural networks*, Comput. Math. Appl., 104 (2021), pp. 1–13, https://doi.org/10.1016/j.camwa.2021.11.001.

[69] L. Sirovich, *Turbulence and the dynamics of coherent structures part I: Coherent structures*, Quart. Appl. Math., 45 (1987), pp. 561–571, https://doi.org/10.1090/qam/910462.

[70] T. Taddei and L. Zhang, *Space-time registration-based model reduction of parameterized one-dimensional hyperbolic PDEs*, ESAIM Math. Model. Numer. Anal., 55 (2021), pp. 99–130, https://doi.org/10.1051/m2an/2020073.

[71] M. M. Valero, L. Jofre, and R. Torres, *Multifidelity prediction in wildfire spread simulation: Modeling, uncertainty quantification and sensitivity analysis*, Environ. Model. Softw., 141 (2021), 105050, https://doi.org/10.1016/j.envsoft.2021.105050.

[72] S. Verma, G. Novati, and P. Koumoutsakos, *Efficient collective swimming by harnessing vortices through deep reinforcement learning*, Proc. Natl. Acad. Sci. USA, 115 (2018), pp. 5849–5854, https://doi.org/10.1073/pnas.1800923115.

[73] L. Vilar, S. Herrera, E. Tafur-García, M. Yebra, J. Martínez-Vega, P. Echavarría, and M. P. Martín, *Modelling wildfire occurrence at regional scale from land use/cover and climate change scenarios*, Environ. Model. Softw., 145 (2021), 105200, https://doi.org/10.1016/j.envsoft.2021.105200.

[74] Q. Wang, J. S. Hesthaven, and D. Ray, *Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem*, J. Comput. Phys., 384 (2019), pp. 289–307, https://doi.org/10.1016/j.jcp.2019.01.031.

[75] G. Welper, *Interpolation of functions with parameter dependent jumps by transformed snapshots*, SIAM J. Sci. Comput., 39 (2017), pp. A1225–A1250, https://doi.org/10.1137/16M1059904.

[76] G. Welper, *Transformed snapshot interpolation with high resolution transforms*, SIAM J. Sci. Comput., 42 (2020), pp. A2037–A2061, https://doi.org/10.1137/19M126356X.

[77] B. Zorawski, S. Burela, P. Krah, A. Marmi, and K. Schneider, *Automated Transport Separation Using the Neural Shifted Proper Orthogonal Decomposition*, preprint, https://arxiv.org/abs/2407.17539, 2024.