scientific reports

OPEN

Cost function for low-dimensional manifold topology assessment

Kamila Zdybał[1,2]✉, Elizabeth Armstrong[3], James C. Sutherland[3]✉ & Alessandro Parente[1,2]✉

In reduced-order modeling, complex systems that exhibit high state-space dimensionality are described and evolved using a small number of parameters. These parameters can be obtained in a data-driven way, where a high-dimensional dataset is projected onto a lower-dimensional basis. A complex system is then restricted to states on a low-dimensional manifold where it can be efficiently modeled. While this approach brings computational benefits, obtaining a good quality of the manifold topology becomes a crucial aspect when models, such as nonlinear regression, are built on top of the manifold. Here, we present a quantitative metric for characterizing manifold topologies. Our metric pays attention to non-uniqueness and spatial gradients in physical quantities of interest, and can be applied to manifolds of arbitrary dimensionality. Using the metric as a cost function in optimization algorithms, we show that optimized low-dimensional projections can be found. We delineate a few applications of the cost function to datasets representing argon plasma, reacting flows and atmospheric pollutant dispersion. We demonstrate how the cost function can assess various dimensionality reduction and manifold learning techniques as well as data preprocessing strategies in their capacity to yield quality low-dimensional projections. We show that improved manifold topologies can facilitate building nonlinear regression models.

In the era of big data, numerous science and engineering disciplines use dimensionality reduction to obtain lower-dimensional representations of complex physical systems with many degrees of freedom[1–8]. Large data coming from system measurements or simulations is frequently the starting point of reduced-order modeling. These high-dimensional datasets, ubiquitous in areas such as plasma physics, chemically reacting flows, neuroscience, genomics and transcriptomics, electrochemistry or atmospheric physics, often exhibit strongly attracting low-dimensional manifolds[9–20]. Describing the system evolution on those manifolds alone can thus be a viable modeling strategy[21–24]. After projecting the original variables onto a lower-dimensional basis, system dynamics can be tracked on a lower-dimensional manifold, embedded in the original state-space. This approach provides substantial reduction to the number of parameters needed to visualize, describe and predict complex systems. To date, linear and nonlinear dimensionality reduction techniques have been used to find lower-dimensional spaces to represent multivariate datasets and build reduced-order models in those spaces.

Some topological properties of low-dimensional data representations can make reduced-order modeling difficult. A particularly undesired behavior are overlapping states on a manifold which can result in non-uniqueness in dependent variable values. For instance, with the manifold parameters used as regressors in nonlinear regression tasks, any ambiguity in dependent variable values can hinder successful modeling. Another characteristic of a problematic manifold are large gradients in the dependent variable values. These can appear when observations on manifolds in important regions are compressed with respect to other, less important regions. If dependent variable values change rapidly over such compressed regions, features of small sizes are formed that can pose modeling difficulty[25]. Maintaining moderate gradients on manifolds is thus a desired characteristic.

In this paper, we are motivated by the emerging discussion on the need to characterize the quality of low-dimensional manifolds[26–30]. Quantitative tools are needed in areas where researchers tune the hyper-parameters of dimensionality reduction or manifold learning techniques to obtain improved manifolds of particular types of data[31–35]. When manifold learning is used for efficient data visualization, good quality manifolds can help uncover important multivariate relationships in complex datasets such as genomics or transcriptomics data[34,36–38]. In artificial intelligence, there is a need to determine the quality of manifold representations in neural networks, where the recent work on predictive learning demonstrated how those representations can vary throughout the learning stages[19]. Detecting intersection between several manifolds can be of interest in learning object manifolds

[1]Université Libre de Bruxelles, École polytechnique de Bruxelles, Aero-Thermo-Mechanics Laboratory, Brussels, Belgium. [2]Université Libre de Bruxelles and Vrije Universiteit Brussel, Combustion and Robust Optimization Group (BURN), Brussels, Belgium. [3]Department of Chemical Engineering, University of Utah, Salt Lake City, UT, USA. ✉email: Kamila.Zdybal@ulb.ac.be; kamilazdybal@gmail.com; James.Sutherland@utah.edu; Alessandro.Parente@ulb.be

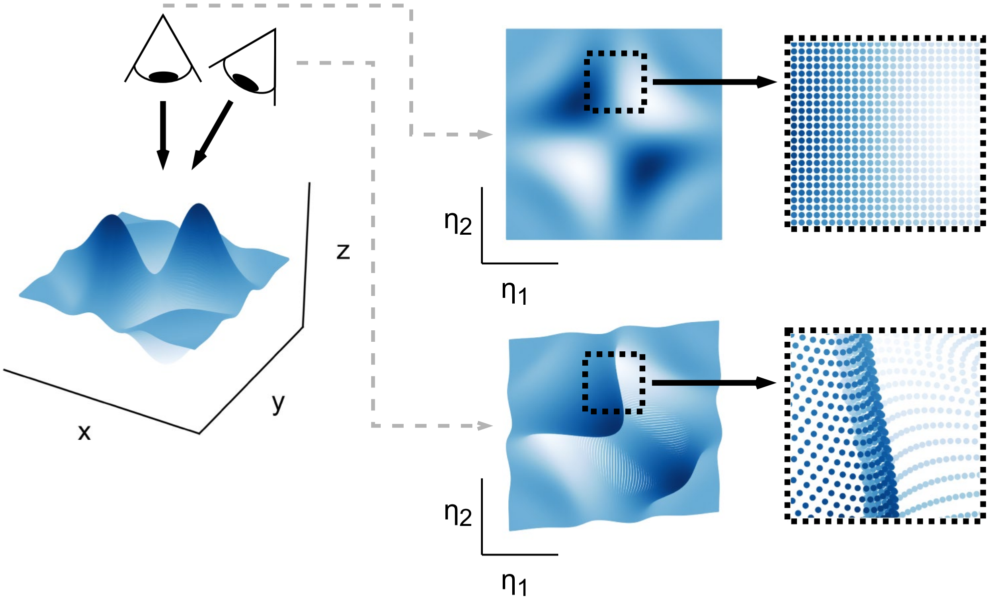

Figure 1. Illustrative demonstration of non-uniqueness introduced during low-dimensional data projection. Various 2D projections of a 3D synthetic dataset can be formed by looking at the dataset at various angles to the *z*-axis and collapsing all observations onto a plane of sight. We demonstrate two example projections that can be formed: a top-down projection and a projection resulting from looking at an angle to the *z*-axis. In the new projected coordinates, $[\eta_1, \eta_2]$, the top-down projection is unique everywhere, while the projection at an angle introduces regions of non-uniqueness with overlapping observations.

in deep neural networks (DNNs)[39]. Similar need applies in domains that study real neural networks and aim at discovering low-dimensional trajectories in neural activity[15,18,20,40–42]. Finally, quality manifold topologies become important in fields such as fluid dynamics, plasma physics or combustion, where nonlinear regression is commonly integrated into the reduced-order modeling workflow[43–48].

Here, we propose a manifold-informed metric that can quantitatively assess low-dimensional data parameterizations. The proposed metric takes into account two factors that affect modeling in particular: feature sizes and multiple scales of variation in manifold topology that can convey non-uniqueness. Our metric is scalable with target manifold dimensionality and can work across different dimensionality reduction and manifold learning techniques. Moreover, dependent variables that are most relevant in modeling can be selected a priori and the manifold topology can be assessed for those variables specifically.

The proposed metric can further serve as a cost function in manifold optimization tasks. In this work, we delineate and test a few applications. Primarily, we demonstrate how the cost function can help identify appropriate data preprocessing to mitigate problems with ill-behaved manifolds. While in some cases insights are available as to which data preprocessing results in accurate modeling[49–54], such decisions still have to be largely made through trial and error. Finally, with the many linear and nonlinear dimensionality reduction techniques available in the research community, we show how they can be assessed in their capacity to generate well-defined manifolds. Our quantitative measure can optimize hyper-parameters of a reduction technique. This can be helpful especially for techniques where only heuristic guidelines are available[55]. Often, data preprocessing and hyper-parameter settings are ultimately tied to a given dataset. It is also unclear how (or whether) the settings will carry over if changes such as increasing the manifold dimensionality or extending the state-space with additional data are made later on[56]. Our approach proposed in this work thus provides a way to quantify and automate decisions that need to be made prior to applying a reduction technique.

Results

Cost function formulation.

We base our discussion on the observation that regions of poor manifold topology (such as non-uniqueness) will only affect a dependent variable if there is variation in the variable's values over those regions (see Fig. 1). With this premise, our cost function is derived from variance in dependent variable values happening across different length scales on a manifold. As recently proposed in Ref.[30], we compute the normalized variance of a dependent variable, $\mathcal{N}(\sigma)$, for selected length scales, σ, on a manifold. We then analyze the derivative of $\mathcal{N}(\sigma)$ with respect to σ, denoted as $\hat{\mathcal{D}}(\sigma)$. This derivative captures the information about how the normalized variance changes as the manifold is scanned at varying length scales given by $\sigma \in \langle \sigma_{min}, \sigma_{max} \rangle$. The detailed mathematical description of $\mathcal{N}(\sigma)$ and $\hat{\mathcal{D}}(\sigma)$ can be found in the "Methods" section and in Ref.[30]. Given q manifold parameters, $\boldsymbol{\eta} = [\eta_1, \eta_2, \dots, \eta_q]$, the resulting normalized variance derivative, $\hat{\mathcal{D}}(\sigma)$, can be computed for each dependent variable in a relevant set, $\boldsymbol{\phi} = [\phi_1, \phi_2, \dots, \phi_m]$. Figure 2a shows a visual example of how the normalized variance derivative assesses information content at various manifold length scales for one dependent variable, ϕ. The length scales at which peaks occur in a variable's $\hat{\mathcal{D}}(\sigma)$ profile

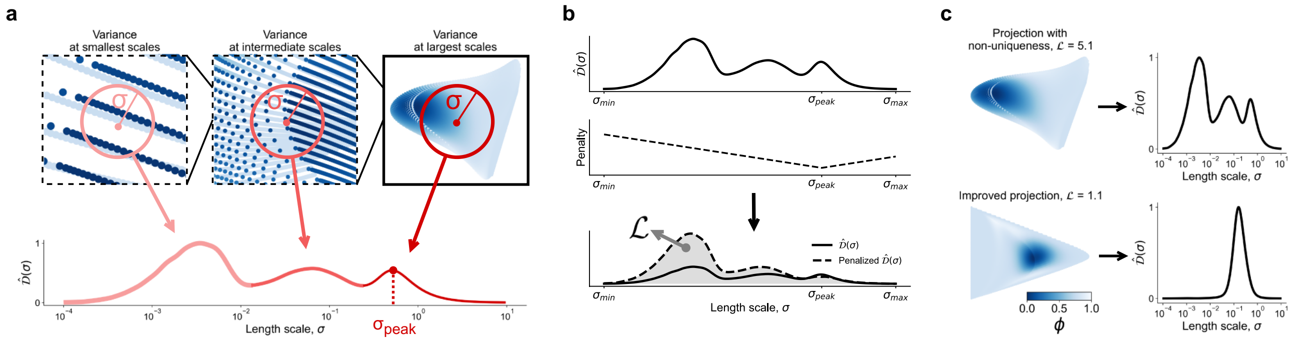

Figure 2. The proposed cost function, $\mathcal{L}$, distills information about variance in dependent variable's values occurring at different length scales, σ, on a manifold. (**a**) Manifold is first scanned at different values of σ for any variation in the dependent variable's values. Multiple scales of variation manifest themselves in multiple peaks seen in the constructed normalized variance derivative curve, $\hat{\mathcal{D}}(\sigma)$. The location of the rightmost peak, σ_{peak}, denotes the largest feature size on a manifold. Variance occurring at scales $\sigma \ll \sigma_{peak}$ is a strong indicator of non-uniqueness. (**b**) The cost function, $\mathcal{L}$, is computed by integrating the penalized $\hat{\mathcal{D}}(\sigma)$ curve. The introduced penalty amplifies the area under $\hat{\mathcal{D}}(\sigma)$ occurring at $\sigma < \sigma_{peak}$ and especially at $\sigma \ll \sigma_{peak}$. The contributions from all scales of variation are thus summed up and embedded in the cost value. (**c**) Assessing example 2D projections with the proposed cost function. The first projection exhibits non-uniqueness, with regions where smallest and highest values of the dependent variable, ϕ, overlap each other. The non-uniqueness is significantly reduced on the improved projection and the corresponding $\hat{\mathcal{D}}(\sigma)$ curve exhibits a single rise indicating a single feature size over the whole manifold. We show the associated costs for the two projections, with the cost for the projection with non-uniqueness being greater.

indicate feature sizes. Each scale at which a dependent variable shows variation over the manifold, including variation due to non-uniqueness, will create its own imprint in $\hat{\mathcal{D}}(\sigma)$. The largest spatial scale at which the dependent variable exhibits variation (the largest feature size) on a manifold is reflected in the rightmost peak location in the $\hat{\mathcal{D}}(\sigma)$ profile, and we define it as σ_{peak} (see Fig. 2a). In the "Methods" section, we present a more detailed discussion of how σ_{peak} is obtained. Any smaller feature sizes appear as additional peaks for $\sigma < \sigma_{peak}$. The 2D projection seen in Fig. 2a exhibits severe non-uniqueness which manifests itself in variance occurring at very small scales, $\sigma \ll \sigma_{peak}$.

Two particularly appealing characteristics of the normalized variance derivative are taken into account for the design of our cost function. First, the locations of peaks in the $\hat{\mathcal{D}}(\sigma)$ curve convey feature sizes on manifolds. With an appropriate penalty with respect to the largest feature size, σ_{peak}, we favor manifolds that maintain large feature sizes, as those should facilitate modeling. Second, with multiple scales of variation present on manifolds, the area under the $\hat{\mathcal{D}}(\sigma)$ curve will increase due to the additional peak(s) for $\sigma < \sigma_{peak}$. The cost function proposed herein, $\mathcal{L} = \mathcal{L}(\boldsymbol{\eta}, \boldsymbol{\phi})$, is computed from a penalized area under the $\hat{\mathcal{D}}(\sigma)$ curve(s). By integrating the penalized $\hat{\mathcal{D}}(\sigma)$ curve(s), we sum over the effects that multiple scales of variation have on the $\hat{\mathcal{D}}(\sigma)$ profile. A visualization of the penalty function and its effect on the $\hat{\mathcal{D}}(\sigma)$ curve for the manifold introduced in Fig. 2a is shown in Fig. 2b. We see how the area under $\hat{\mathcal{D}}(\sigma)$ weighted by the penalty function is amplified at scales far from σ_{peak}, therefore penalizing the variance occurring at smaller scales more heavily. Furthermore, the cost function gives a single number representing the parameterization "cost" for a given manifold. A larger cost indicates a worse manifold topology and a lower cost indicates an improved manifold topology in terms of modeling. If the cost is computed for m dependent variables, a norm over all $\mathcal{L}_i$ costs can be taken to yield a single cost value for the entire manifold: $\mathcal{L} = ||\mathcal{L}_i||, \forall i \in [1, 2, \ldots, m]$. Figure 2c demonstrates costs computed for a single dependent variable, ϕ, over two different 2D projections. The first projection is the same as analyzed in Fig. 2a. The second projection has an improved topology and non-uniqueness is significantly reduced. The corresponding $\hat{\mathcal{D}}(\sigma)$ curve for the improved projection exhibits a single dominant peak which indicates a single scale of variation in ϕ values over the entire manifold. We report the cost, $\mathcal{L} = \mathcal{L}_{\phi}$, for each projection, with the cost for the projection with non-uniqueness being greater. The mathematical description of the proposed cost function and additional details are provided in the "Methods" section.

There are three key advantages of our proposed cost function. First, manifolds obtained using any technique can be assessed. This includes any ad hoc selected manifold parameters or empirical manifolds obtained directly from training data using dimensionality reduction or manifold learning techniques. Second, manifolds of any dimensionality can be assessed. Finally, manifolds can be assessed with respect to an arbitrary set of m relevant dependent variables, $\boldsymbol{\phi}$. Dependent variables that we are most interested in accurately modeling can be selected, which can include any of the original state variables and any functions of them.

Cost function response to feature size and non-uniqueness. To demonstrate the behavior of the proposed cost function to an increasing feature size, we generate bivariate functions on a uniform square x–y grid, centered around (0, 0). The dependent variable, ϕ, is calculated from a multivariate Gaussian normal distribution, $\phi = \exp(-(x^2 + y^2)/(2s^2))$, where the standard deviation, s, is gradually increased to imitate an increasing feature size. The smallest feature is selected with $s = 0.05$ and the largest feature with $s = 0.6$. Figure 3a shows these functions corresponding to ten Gaussians with increasing s. Below each Gaussian, we plot

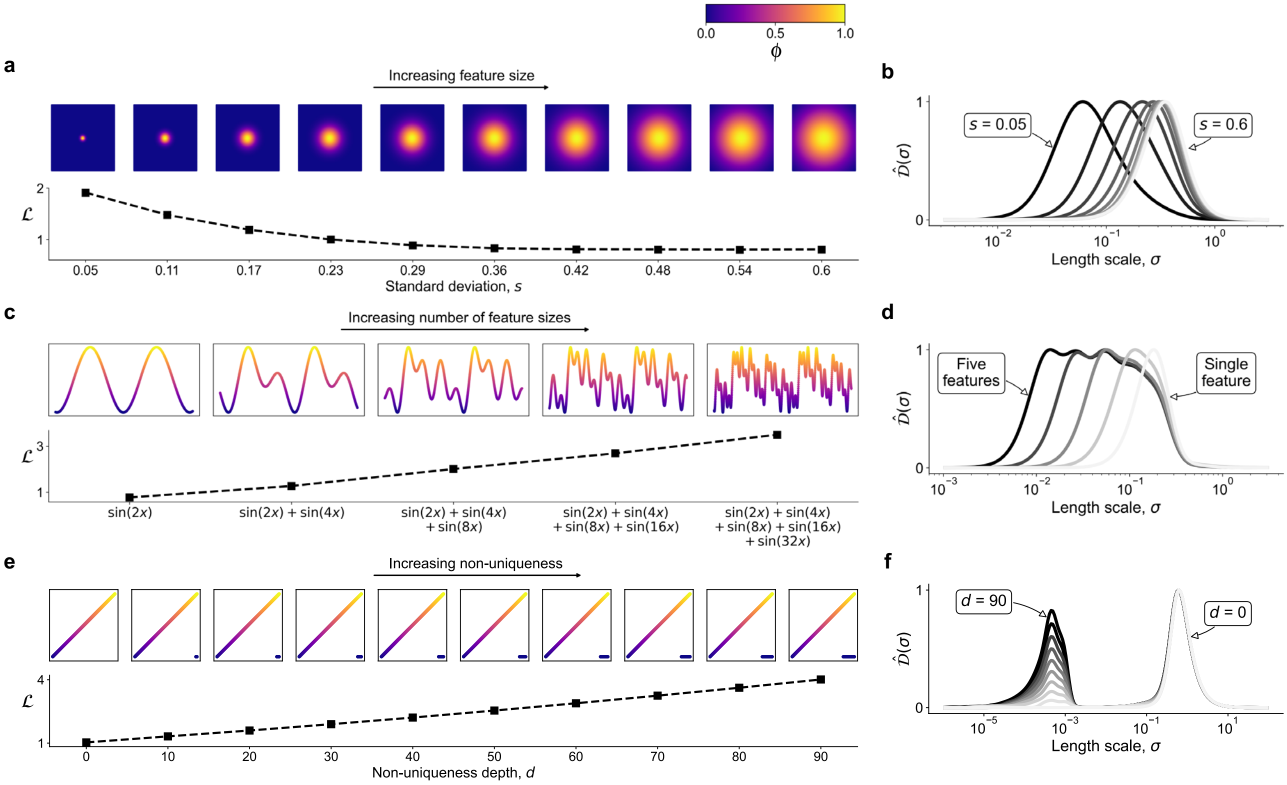

Figure 3. Cost function behavior on toy functions. (**a**) The response of the cost function to increasing feature size. The dependent variable ϕ is generated from the bivariate Gaussian normal distribution with varying standard deviation, s. (**b**) The $\hat{\mathcal{D}}(\sigma)$ curves corresponding to functions with increasing feature size. The curves are plotted in gray-scale denoting an increasing value of s from the darkest curve ($s = 0.05$) to the lightest ($s = 0.6$). (**c**) The response of the cost function to multiple feature sizes. The dependent variable ϕ is generated from a superposition of sine functions with varying frequencies. (**d**) The $\hat{\mathcal{D}}(\sigma)$ curves corresponding to functions with increasing number of feature sizes. The curves are plotted in gray-scale, such that the darkest curve corresponds to five different feature sizes and the lightest curve corresponds to a single feature size. (**e**) The response of the cost function to increasing non-uniqueness. The dependent variable ϕ is generated as a linear function of x with an additional set of overlapping observations for which $\phi = 0$. (**f**) The $\hat{\mathcal{D}}(\sigma)$ curves corresponding to functions with increasing non-uniqueness. The curves are plotted in gray-scale denoting an increasing value of the non-uniqueness depth, d, from the lightest curve ($d = 0$) to the darkest ($d = 90$). In all examples, the dependent variables, ϕ, have been normalized to a [0, 1] range.

the corresponding cost, $\mathcal{L} = \mathcal{L}_{\phi}$, and we observe a decreasing trend in $\mathcal{L}$ with increasing feature size. This is a desired behavior, since larger features should facilitate modeling. A very small feature, like the one obtained from $s = 0.05$, introduces relatively steep gradients, which can be more challenging to model. The response of the $\hat{\mathcal{D}}(\sigma)$ curves to an increasing feature size is shown in Fig. 3b. The location of σ_{peak} gradually shifts to the right with increasing feature size and the area under the $\hat{\mathcal{D}}(\sigma)$ curves decreases. The decrease in cost with an increasing feature size seen in Fig. 3a is thus a result of rewarding the increasing σ_{peak} location and the slightly decreasing area under the $\hat{\mathcal{D}}(\sigma)$ curve. The latter is the effect of a decreasing gradient of ϕ with increasing feature size, leaving less variance in ϕ present at scales away from σ_{peak}.

To further test the response of the cost function to multiple feature sizes, we generate another set of functions, such that the dependent variable, ϕ, is now computed from a superposition of sine functions with various frequencies. Multiple feature sizes are generated from the general formula $\phi = \sum_{k=1}^{n} \sin(2^k x)$ for $n = 1, 2, \ldots, 5$. Thus, for $n = 1$, the function has only one feature size. In Fig. 3c, we observe increasing cost values with each added feature size. The reason for this increase becomes clear once we look at the corresponding $\hat{\mathcal{D}}(\sigma)$ curves in Fig. 3d, where each new feature generates an additional rise in $\hat{\mathcal{D}}(\sigma)$ at length scales smaller than σ_{peak}.

Next, we perform a similar test using functions which introduce increasing levels of non-uniqueness. The dependent variable, ϕ, is now calculated as a linear function of the independent variable, $\phi = x$, being the x-axis. This results in a constant gradient along the x-axis. We introduce non-uniqueness in ϕ by adding overlapping observations whose ϕ values are set to zero. The number of observations added to increase the non-uniqueness is measured with the overlap depth, d, and is increased from no overlapping observations ($d = 0$ observations) to a maximum number of overlapping observations ($d = 90$ observations). Figure 3e presents ten such functions with an increasing depth of non-uniqueness. Here, we observe an increasing trend in $\mathcal{L}$ as we increase the non-uniqueness depth. The multiple scales of variation introduced from non-uniqueness lead to an increased area under the $\hat{\mathcal{D}}(\sigma)$ curves, which is penalized by the cost function. In order to observe how severe the increase in area is under the $\hat{\mathcal{D}}(\sigma)$ curves associated with manifold non-uniqueness, Fig. 3f shows the $\hat{\mathcal{D}}(\sigma)$ curves corresponding to the ten functions with non-uniqueness. The first interesting observation is that the location of σ_{peak} for all $\hat{\mathcal{D}}(\sigma)$ curves is almost identical. This is understandable, since the size of the main feature (the constant gradient) stays the same. The only reason for the increased area under the $\hat{\mathcal{D}}(\sigma)$ curves is the appearance of additional peaks at length scales $\sigma \ll \sigma_{peak}$. Only in the case $d = 0$ (no overlap) the $\hat{\mathcal{D}}(\sigma)$ curve exhibits a single rise. As long as any overlapping observations are introduced, an additional area shows up under the $\hat{\mathcal{D}}(\sigma)$ curve at $\sigma \ll \sigma_{peak}$. This additional area is most pronounced for the largest non-uniqueness depth, $d = 90$.

Finally, it is instructive to discuss the scales, σ, at which we detect overlap in Fig. 3f. The highest rise in $\hat{\mathcal{D}}(\sigma)$ linked to non-uniqueness is happening at $\sigma \approx 5 \times 10^{-4}$ for all cases $d > 0$. This exact value can be linked to the sample density for the ten toy functions with increasing overlap depth. For a normalized manifold, where $x \in [0, 1]$, the distance along the x-axis between data points in the unique region is $\sigma \approx 10^{-3}$. The overlapping observations are located in-between every other unique observation, such that the distance along the x-axis between any observation from the unique region and its nearest overlapping observation is of the order of 10^{-4}. Thus, as the manifold is scanned with varying length scales, σ, once $\sigma \approx 10^{-4}$, the variation in a dependent variable values is captured between a single point from the unique region ($\phi \rightarrow 1$) and a single point from the overlapping region ($\phi = 0$). This creates a sudden increase in the captured variance and shows up as the peak at $\sigma \approx 5 \cdot 10^{-4}$ in the $\hat{\mathcal{D}}(\sigma)$ curves in Fig. 3f. A more in-depth discussion of how scales of variation can be linked to data density on a manifold grid can be found in Ref.[30].

Assessing data preprocessing strategies.

We now turn our attention to multivariate datasets and their lower-dimensional projections. Each dataset considered next is represented by a matrix $\mathbf{X} \in \mathbb{R}^{N \times Q}$, where N is the number of observations and Q is the number of state variables. Since most often $N \gg Q$, Q determines data dimensionality. We consider three disciplines that deal with datasets that are notoriously high dimensional: plasma physics, reacting flows and atmospheric physics. For example, the state-space of a reacting flow is described by the temperature, pressure and chemical composition. There, the high state-space dimensionality originates from many chemical species involved that can easily reach the order of hundreds. In this and the following sections, we demonstrate a few practical applications of the proposed cost function using those datasets.

Reduced-order models leverage the fact that multivariate datasets can often be successfully re-parameterized by a reduced set of low-dimensional parameters. To date, numerous techniques have been employed for dimensionality reduction of multivariate data. Those include linear techniques such as principal component analysis (PCA), independent component analysis (ICA), distance metric learning (DML) or linear discriminant analysis (LDA) and nonlinear techniques such as kernel PCA (KPCA), Isomap[57],locally linear embedding (LLE) and its variants[58] or autoencoders[59]. For the purpose of data visualization, t-distributed stochastic neighbor embedding (t-SNE)[55] and uniform manifold approximation and projection (UMAP)[60] are gaining popularity in various research disciplines[7,36–38,61–64]. A short summary of dimensionality reduction and manifold learning techniques explored in this work can be found in the "Methods" section.

Prior to applying dimensionality reduction, training datasets are often preprocessed. The most straightforward strategy is data normalization by centering and scaling each state variable. Other preprocessing approaches involve data sampling to mitigate imbalance in observation density, or feature selection. The effect of data preprocessing alone can have a large impact on the resulting low-dimensional manifold topology constructed from such data[53,65]. To demonstrate how significant those changes can be, Fig. 4a shows 2D PCA projections of a dataset describing argon plasma (N = 100,700, Q=36)[66,67], where the state-space is spanned by the following variables: temperature of heavy species, T_h, temperature of electrons, T_e, and 34 species mass fractions that contain 31 electronic states of argon, two levels of ionized argon, Ar and Ar^+, and electrons. The projections are generated with various scaling techniques applied on the dataset (see the "Methods" section for the summary of all scaling techniques used in this work). In the top row of Fig. 4a, the projections are colored by T_e and in the bottom row by the electron mass fraction, Y_e. Our visual analysis shows that preprocessing can affect manifold topologies significantly. Various topologies are expected to perform differently in reduced-order modeling due to changes in feature sizes or the level of non-uniqueness.

We quantitatively assess those changes and rank the preprocessing strategies in their capacity to generate quality manifold topologies. In Fig. 4b, we show costs, $\mathcal{L}$, for the various 2D PCA projections visualized above, and for comparison, for the analogous 3D PCA projections (although not visualized). Here, $\mathcal{L}$ is computed as the L_1-norm over the individual costs for the relevant dependent variables: T_h, T_e, Y_e and Y_{Ar}. Thus, in Fig. 4a, the projections are colored by the two target dependent variables to allow us for making a visual connection between how the projection topologies look like versus how the cost function assesses a given projection. We note that VAST scaling results in the lowest cost for 2D projections. Looking at the corresponding visualization in Fig. 4a, we can see that T_e and Y_e values change relatively smoothly across the manifold, even though the VAST projection introduces the "spike" region, where many data observations are compressed. For comparison, projections resulting from the remaining scalings introduce relatively steep gradients and overlap for a few of them, in some region of the projection, either for the T_e or the Y_e variable. Finally, we note that costs drop across all scaling strategies explored when the PCA projection dimensionality is increased to 3D.

We additionally explore feature selection (otherwise known as variable subset selection) as another strategy in the data preprocessing pipeline. Feature selection involves finding a meaningful subset, $\mathbf{X}_S \in \mathbb{R}^{N \times S}$, of the original Q state variables ($S < Q$), and using this subset in data science or machine learning algorithms, instead of the full set of state variables[68]. This time, we use a reacting flow dataset describing combustion of syngas in air. In Fig. 4c, we show 3D PCA projections of an 11-dimensional dataset (N = 14,550, Q = 11), where the eleven dimensions are spanned by temperature, T, and ten mass fractions of chemical species, denoted Y_i for species i. The different projections seen in Fig. 4c correspond to different preprocessing applied to the original combustion dataset. In the upper row, the 3D projections are generated with various scaling techniques applied on the training

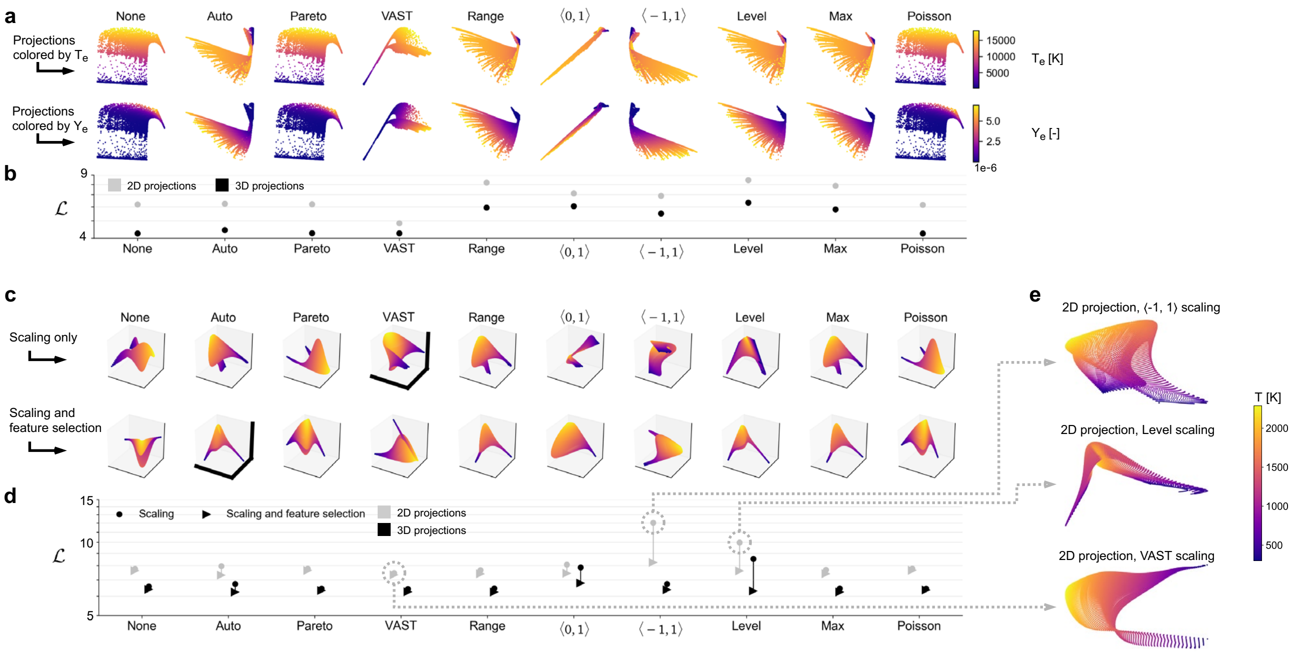

Figure 4. Ranking of different training data preprocessing strategies in their capacity to produce quality low-dimensional projections using PCA. (**a**) 2D PCA projections generated from an argon plasma dataset. We show projections resulting from various scaling applied to the original data. In the top row, the projections are colored by the electron temperature, T_e, and in the bottom row, by the electron mass fraction, Y_e. (**b**) The overall cost, $\mathcal{L}$, computed as the L_1-norm over the individual costs for the selected dependent variables: temperature of heavy species, T_h, temperature of electrons, T_e, electron mass fraction, Y_e, and argon mass fraction, Y_{Ar}. For comparison, costs are shown for 2D and 3D PCA projections. (**c**) 3D PCA projections generated from a reacting flow dataset describing combustion of syngas in air. We compare projections resulting from only scaling the data (top row) with projections resulting from scaling combined with feature selection (bottom row). All projections are colored by the temperature variable. (**d**) The overall cost, $\mathcal{L}$, computed as the L_1-norm over the individual costs for the selected dependent variables: temperature and six important chemical species mass fractions. We compare costs corresponding to only scaling the data (circles) versus scaling with feature selection (triangles). We also compare costs corresponding to the 3D projections visualized above (black markers) with the analogous costs of 2D projections (gray markers). The optimal manifold topology corresponding to the lowest $\mathcal{L}$ for each preprocessing strategy is highlighted with thicker axes in (**c**). (**e**) Visualization of 2D projections corresponding to three selected cases of only scaling the original data: $\langle -1, 1\rangle$ and Level scaling (corresponding to the two highest $\mathcal{L}$) and VAST scaling (corresponding to the lowest $\mathcal{L}$).

dataset (see "Methods"). The bottom row shows 3D projections resulting from different scalings combined with feature selection applied to the original training data. All PCA projections are colored by the temperature. For the purpose of this demonstration, we use a newly developed feature selection algorithm that uses the cost function to guide the optimal selection of the variables from the original training data[65]. By minimizing the cost of the resulting data projection, we optimize the feature selection process from the point of view of manifold topology. The detailed description of the feature selection algorithm can be found in the "Methods" section. In Fig. 4d we show costs, $\mathcal{L}$, tied to the two preprocessing strategies explored: just scaling the data (circles) versus scaling with feature selection (triangles). Here, $\mathcal{L}$ is computed as the L_1-norm over the individual costs for the relevant dependent variables: T, Y_{H_2}, Y_{O_2}, Y_{H_2O}, Y_{OH}, Y_{CO} and Y_{CO_2}. The dependent variables were selected manually as the most important candidates in modeling. Black markers show costs corresponding to the 3D PCA projections visualized in Fig. 4c. For comparison, with gray markers we show costs for analogous 2D PCA projections. Three of these 2D projections corresponding to scaling only are visualized in Fig. 4e—the two worst projections (largest $\mathcal{L}$) and the best projection (smallest $\mathcal{L}$). We note that the worst 2D projection corresponding to $\langle -1, 1\rangle$ scaling is severely folded over itself which is likely the reason for the high cost. The Level projection also introduces significant non-uniqueness, with the low-temperature regions being represented over narrow geometry on a manifold. The best 2D projection corresponds to VAST scaling. As seen in Fig. 4e, this projection is much more unique compared to $\langle -1, 1\rangle$ or Level scaling 2D projections. However, there is a visible "twist" on the VAST 2D projection, which is untangled when a 3D projection is considered instead. This is a possible factor for a decreased cost between the 2D and the 3D PCA projection. The lowest costs for 3D projections happened for VAST (for scaling only) and Auto (for scaling with feature selection). The corresponding visualized projections are marked with thicker axes in Fig. 4c. These two projections are characterized by relatively well-spaced feature sizes and reduced non-uniqueness. Finally, similarly as we have observed for the argon plasma data, costs drop across all scaling strategies when the PCA projection dimensionality is increased from 2D to 3D.

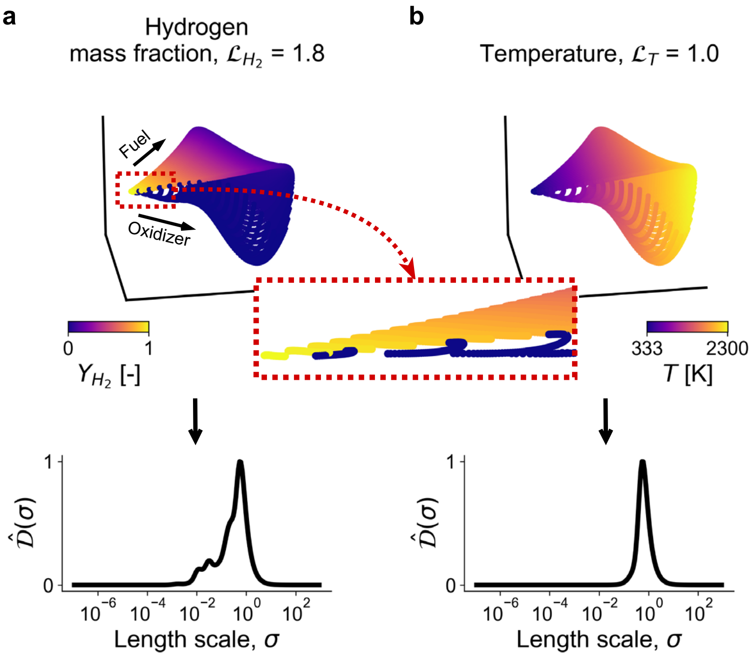

Figure 5. Example 3D manifold in a reacting flow dataset describing combustion of hydrogen in air. (**a**) The manifold exhibits region (marked with the red dashed box) with a step-change in hydrogen (fuel) mass fraction values, Y_{H_2}. (**b**) For comparison, the temperature variable, T, on the same manifold does not experience large variation within the considered region and the cost associated with the temperature variable is lower.

Our analysis from Fig. 4 reveals that appropriate data preprocessing can help in generating a better quality low-dimensional manifold. In particular, optimized feature selection can be beneficial as it decreases costs over the relevant dependent variables with respect to costs associated with only scaling the data. With the many possible data preprocessing techniques encountered in the data science community, the cost function allows for quantitative rankings without the need to analyze manifold topologies manually. This, in turn, can help fine-tune data preprocessing to a given dataset and a desired manifold dimensionality.

Detecting large gradients on manifolds. Using the proposed cost function, manifolds can be assessed with respect to different relevant dependent variables separately. If there is a good coverage of regions where the selected variables vary over the entire manifold, problematic regions on manifolds can be detected with greater reliability. Figure 5a demonstrates a 3D manifold embedded in a 9-dimensional state-space of a reacting flow dataset describing combustion of hydrogen in air (N = 13,468, Q = 9). The topology of this manifold is curved such that two regions where fuel and oxidizer originate from are brought closely together. This region is better visualized in the zoomed-in dashed box. When the manifold is colored by the hydrogen (fuel) mass fraction in Fig. 5a, we see a step-change in the $\phi = Y_{H_2}$ values over the considered region. This potentially undesired behavior can be detected with the cost function as it increases the area under the $\hat{\mathcal{D}}(\sigma)$ curve at length scales smaller than σ_{peak}. The cost associated with the hydrogen mass fraction is $\mathcal{L}_{H_2} = 1.8$. For comparison, the temperature variable does not experience large variation within the considered region (Fig. 5b) and the cost associated with temperature is lower ($\mathcal{L}_T = 1.0$). We observe a single peak in the $\hat{\mathcal{D}}(\sigma)$ profile corresponding to the temperature variable.

Regions where opposing physical phenomena meet over small length scales on a manifold (in this case fuel and oxidizer streams) can prove to be difficult to model accurately. For instance, some regression models are known to struggle in the presence of large gradients in dependent variable values[25]. Such problematic regions of compressed observations on manifolds can be detected by analyzing costs across several dependent variables. If there exists an important dependent variable that varies across the problematic region (such as the Y_{H_2} variable in Fig. 5a), it can help discard the problematic manifold topology. We note that the choice of the relevant dependent variables is important in assessing whether a given manifold topology is appropriate or not from the modeling perspective. Some variables, such as the temperature variable alone in the example shown in Fig. 5b, might not be effective at exposing regions of non-uniqueness in data projections, that can affect other relevant variables. We also note that the dependent variables for which the cost is computed can be arbitrary and do not have to be selected from the set of the original state variables as we have done here. They can equally be functions of the state variables. In reacting flow applications, these can for instance be the production rates of chemical species, related to the state-space through nonlinear Arrhenius expressions.

Manifold assessment across dimensionality. We now demonstrate how the cost function applies to manifolds of arbitrary dimensionality. This can be useful in determining an appropriate dimensionality for rep-

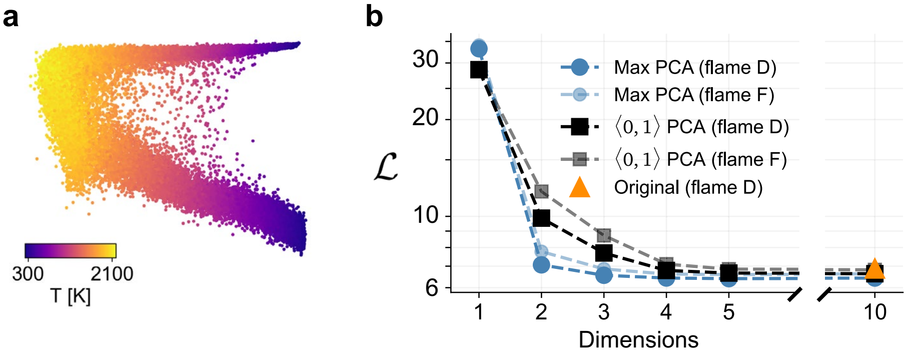

Figure 6. Cost function applied to PCA projections of experimental Sandia flame data. (**a**) A 2D projection of flame D using PCA with Max scaling colored by temperature. (**b**) The overall cost, $\mathcal{L}$, computed as the L_1-norm over the individual costs for selected dependent variables: temperature and six important chemical species mass fractions when using PCA for reduction between one and ten dimensions. The figure includes curves for using Max and $\langle 0, 1\rangle$ scalings with PCA as well as the cost for the original ten-dimensional manifold.

resenting the variables of interest when using a reduction technique. The demonstration is first done using an experimental combustion dataset known as the Sandia flame D dataset[69]. This dataset contains approximately $N = 57{,}000$ observations each for temperature and various species mass fractions ($Q = 10$) over six different heights in a methane and air piloted jet flame. The experimental dataset also presents the opportunity to demonstrate the cost function behavior on manifolds containing noise. Figure 6a shows an example 2D PCA projection with Max scaling of the Sandia flame D data colored by temperature. This manifold is less structured than those seen in Figs. 4 and 5 for the numerically simulated data. With noisy data, such as seen from experiment, it can be difficult to visually assess the quality of manifold topologies, especially in higher dimensions. The proposed cost function should adequately assess such topologies since the $\hat{\mathcal{D}}(\sigma)$ curves remain smooth even for noisy data[30].

Figure 6b demonstrates how the cost function behaves with increasing dimensionality of a projection using PCA, up to the original dimensionality of the system. The two opaque curves correspond to the scalings that resulted in the highest and lowest costs for the projected experimental data. Also included is the cost for the original ten-dimensional manifold. For the most part, we see a decrease in the cost with increasing dimensionality. Increasing the dimensionality further after four dimensions appears to have negligible effects on the feature sizes of the optimizing variables since the cost function flattens out. At this point, the computational cost of adding dimensions to a model most likely outweighs any benefit seen from slight increases in feature sizes. Any differences between the various scalings also become less noticeable as dimensionality increases. We also see that rotating the manifold along new principal component axes, even without reducing dimensionality, can slightly reduce the cost compared to the original manifold. This is due to variance being measured along new coordinates. For comparison, in Fig. 6b we also include costs with increasing dimensionality for the Sandia flame F dataset (transparent curves), which is another experimental case that exhibits stronger effects of flow turbulence than flame D. As a result, flame F experiences conditions with local flame extinction, absent in the flame D dataset. Thus, flame F includes many more states of methane combustion than flame D and the manifold corresponding to flame F covers a wider range of attainable thermo-chemical states. For instance, for a specific stoichiometric condition, the flame can be either in an ignited or in an extinguished state. We thus expect an increased required manifold dimensionality for flame F to achieve the same parameterization quality as compared to flame D. This is indeed the case seen in Fig. 6b, where costs are generally higher for flame F than for flame D. The proposed cost function could be used in this way to determine an appropriate minimum dimensionality for the projection of data, with the goal of facilitating modeling of the optimizing variables[70,71].

Manifold assessment across various dimensionality reduction and manifold learning techniques. We now revisit a recent PCA-based reduction of a high-fidelity dataset obtained from delayed detached eddy simulation in atmospheric physics[46]. This data simulates the Cedval A1–5 wind tunnel measurements of atmospheric dispersion of a pollutant in the vicinity of a rectangular building[72]. The numerical data that we use here represents a planar slice through the computational domain, perpendicular to the downstream wind direction. The dataset has $N = 18{,}540$, $Q = 16$ and the original state-space is composed of variables common to atmospheric physics applications, such as the three velocity components, pressure, Reynolds number, or the turbulent viscosity, kinetic energy and dissipation rate. All parameters in the data have been averaged over a 3.5 s period. We use an outlier detection algorithm to remove outliers from the data (see "Methods"). The aim of the original study[46] was to predict the turbulent Schmidt number, Sc_t, from the PCA-derived manifold parameters. This demonstrates an application where regression is the end goal for obtaining a data-driven correlation of a physical quantity with the manifold parameters, $\boldsymbol{\eta}$. We thus seek such regression function f that $Sc_t \approx f(\boldsymbol{\eta})$. While in the original work PCA was used to reduce the state-space dimensionality, here we benchmark three additional techniques: UMAP, Isomap and t-SNE. Following a similar methodology as demonstrated in Fig. 4, we first searched for the best scaling option for each reduction technique. In addition, we introduce three novel scaling methods inspired by the variable stability (VAST) scaling[49]. We denote them S1, S2 and S3 (see Table 1).

Name	Scaling factor d_j	Center c_j
None	1	0
Auto[51]	s_j	$\bar{X}_j$
Pareto[52]	$\sqrt{s_j}$	$\bar{X}_j$
VAST[49]	$s_j^2/\bar{X}_j$	$\bar{X}_j$
Range[51]	$\max(X_j) - \min(X_j)$	$\bar{X}_j$
$\langle 0, 1 \rangle$	$\max(X_j) - \min(X_j)$	$\min(X_j)$
$\langle -1, 1 \rangle$	$1/2(\max(X_j) - \min(X_j))$	$1/2(\max(X_j) + \min(X_j))$
Level[51]	$\bar{X}_j$	$\bar{X}_j$
Max	$\max(X_j)$	$\bar{X}_j$
Poisson[50]	$\sqrt{\bar{X}_j}$	$\bar{X}_j$
S1	$s_j^2 k_j^2/\bar{X}_j$	$\bar{X}_j$
S2	$s_j^2 k_j^2/\max(X_j)$	$\bar{X}_j$
S3	$s_j^2 k_j^2/(\max(X_j) - \min(X_j))$	$\bar{X}_j$

Table 1. Data scaling techniques used in this work.

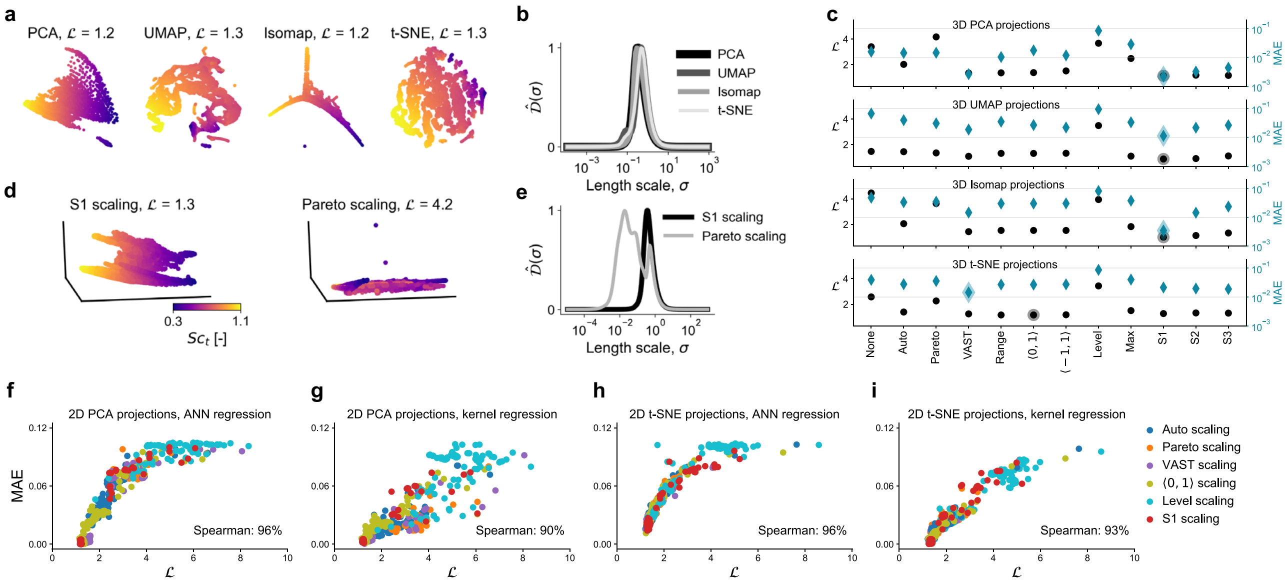

Figure 7. Relating the cost function to nonlinear regression predictions based on various low-dimensional projections of the atmospheric pollutant dispersion data. The relevant dependent variable is the turbulent Schmidt number, $\phi = Sc_t$. (**a**) 2D data projections using PCA, UMAP, Isomap and t-SNE with the cost for $\phi = Sc_t$ reported for each projection. (**b**) The $\hat{\mathcal{D}}(\sigma)$ curves corresponding to the 2D projections. (**c**) Ranking of different data scaling techniques using the cost function (black dots) with the mean absolute errors (MAE) for ANN prediction of Sc_t across all four projection techniques (blue diamonds) using 3D data projections. The minimum $\mathcal{L}$ and the minimum MAE for each technique is marked with a shaded outline. Scalings that generally exhibit lowest costs (VAST, S1, S2, S3) also result in the smallest MAE. (**d**) Example 3D PCA projections resulting from applying two scaling options to the original data: S1 and Pareto scaling. For this dataset, S1 scaling allowed for generating a manifold where the dependent variable of interest, Sc_t, has a smooth gradient across one of the manifold dimensions. Pareto scaling, exhibiting high cost, collapses most of the data observations onto a planar structure. (**e**) The $\hat{\mathcal{D}}(\sigma)$ curves corresponding to the 3D PCA projections with S1 and Pareto scaling. (**f**–**i**) Scatter plots of $\mathcal{L}$ versus MAE from ANN and kernel regression predictions of Sc_t. We show predictions based on 600 different 2D PCA projections (**f**, **g**), and based on 600 different 2D t-SNE projections (**h**, **i**) as the independent manifold parameters. We test a few selected scaling techniques applied to the atmospheric dispersion data; legend applies to all figures (**f**–**i**).

S1 scaling is an extension of VAST, where the effect of non-normality is considered by multiplying the standard deviation by the data kurtosis. S2 and S3 are variations of S1 obtained by replacing the mean value in the coefficient of variation by the maximum and the range of each variable respectively. In Fig. 7a, we compare qualities of 2D projections, corresponding to the best scaling, both visually and using our cost function. We report costs for the $\phi = Sc_t$ only, since that was the modeled dependent variable. An appropriate scaling allowed us, in many cases, to find significantly better projections. For instance, the worst 2D PCA projection is associated with $\mathcal{L} = 5.2$, while the best (visualized) with $\mathcal{L} = 1.2$. Figure 7b shows the $\hat{\mathcal{D}}(\sigma)$ curves corresponding to the 2D projections visualized in Fig. 7a and to $\phi = Sc_t$. The $\hat{\mathcal{D}}(\sigma)$ curves are mostly composed of a single rise, suggesting that the non-uniqueness in the 2D projections has been remedied by selecting an appropriate data scaling.

Improved manifold topologies yield more accurate regression.

Nonlinear regression can be used in combination with dimensionality reduction to provide mapping functions, *f*, between the manifold parameters and physical quantities of interest. Regression can help discover robust relationships between manifold parameters and physical quantities that can then be injected in computational models and simulations[18,44,46,73]. Techniques, such as artificial neural networks (ANNs), Gaussian process regression (GPR) or kernel regression are commonly used in this context.

Below, we continue the example of the atmospheric dispersion dataset and demonstrate improvements in nonlinear regression performance when parameters of an improved manifold topology are used as regressors. We first train an ANN model to predict Sc_t based on 3D PCA, UMAP, Isomap and t-SNE projections resulting from different data scaling options. The details on the ANN model used here are provided in the "Methods" section. The scalings ranking for these 3D projections, along with the mean absolute errors (MAE) for Sc_t predictions using ANN are reported in Fig. 7c. We note that generally, the scalings which resulted in lowest costs (VAST, S1, S2 and S3) across all four reduction techniques exhibit lowest MAE. The minimum $\mathcal{L}$ and MAE for each technique is marked with a shaded outline. With the exception of t-SNE, the minimum $\mathcal{L}$ and the minimum MAE happened for the same scaling option.

The reason for a good ANN performance on those four scalings can be further understood by visualizing example projections. In Fig. 7d, we show 3D PCA projections of the atmospheric dispersion data, corresponding to the best scaling option (S1 with $\mathcal{L} = 1.3$) and the worst scaling option (Pareto with $\mathcal{L} = 4.2$) as per the ranking shown in Fig. 7c. The quality of projections changes visibly with the change of data scaling. For the S1 scaling projection, we observe a clear gradient of Sc_t throughout the manifold, while for the Pareto scaling case, the 3D projection introduced significant scatter in Sc_t values, with many observations compressed onto a nearly planar structure in the reduced 3D space. The latter projection thus introduces significant non-uniqueness in Sc_t values and is more difficult to accurately regress over. The difference in costs for the two 3D PCA projections from Fig. 7d can be further understood by looking at the comparison of $\hat{\mathcal{D}}(\sigma)$ curves in Fig. 7e. Pareto scaling case creates much more variance present at smaller length scales which is consistent with our visual inspection of the projection.

More generally, we observe correlation between the verdict given by $\mathcal{L}$ and the nonlinear regression performance. In Fig. 7f–i, we show scatter plots of $\mathcal{L}$ versus MAE for varying manifold topologies obtained from the atmospheric dispersion dataset. Here, we focus on PCA and t-SNE as the two projection techniques. Each scatter plot takes into account six selected scaling techniques: Auto, Pareto, VAST, $\langle 0, 1 \rangle$, Level and S1. For each scaling technique, we generate 100 distinct 2D projections through creating random variable subsets (feature selections) of the full dataset, before applying PCA or t-SNE. This allows us to have sufficiently many distinct manifold topologies (600 for each scatter plot) for a trend to emerge in Fig. 7f–i. In addition to ANN regression, we apply kernel regression of Sc_t to observe if the correlation between $\mathcal{L}$ and MAE is still present for a different nonlinear regression technique. The details on the kernel regression model used here are provided in the "Methods" section. Although the trends of $\mathcal{L}$ vs. MAE are different between ANN and kernel regression, the correlation measured with Spearman coefficient is high in each case. For ANN regression we observe 96% Spearman correlation for 2D PCA and t-SNE projections. The correlation is lower for kernel regression as compared to ANN regression, with 90% and 93% Spearman coefficient for PCA and t-SNE respectively. In Fig. 7f–i, manifolds corresponding to Level scaling are clustered in the region of high $\mathcal{L}$ and high MAE. This result is consistent with Fig. 7c, where Level scaling is seen to yield high costs and high regression errors. Conversely, the region of smallest $\mathcal{L}$ and smallest MAE in Fig. 7f–i is occupied by topologies resulting from S1 scaling, although some random subsets can result in poorer manifold topologies even with S1 scaling. We observe similar correlation trends for 3D PCA and t-SNE projections; these are shown in the Supplementary material. With the regression hyper-parameters kept constant throughout this exercise, the results presented in Fig. 7f–i suggest that better regression performance can be achieved when manifold topologies are improved. The cost function can thus be used to find optimized regressors for nonlinear regression techniques such as ANN or kernel regression. Other regression techniques are expected to be similarly affected.

Detecting overlap between classes in categorical data.

So far, we have shown examples where the cost was computed using continuous dependent variables. Here, we briefly explore the cost function application to categorical data. The dependent variable, ϕ, can now be formed from numerical values of the class labels. Figure 8a shows in an illustrative way the capability of the $\hat{\mathcal{D}}(\sigma)$ metric to detect overlap between classes on a projection. The overlap between the two clouds of points, each representing one class, is reflected in an additional peak seen in the $\hat{\mathcal{D}}(\sigma)$ profile. When the clouds become sufficiently separated in space, $\hat{\mathcal{D}}(\sigma)$ exhibits a single rise. This behavior is then translated in the cost value, with the cost for the case with overlapping classes being greater. Next, we generate 2D projections of the MNIST handwritten digits dataset[74] using PCA and t-SNE. The data observations are divided into ten classes, each representing one digit. We sample the full MNIST dataset by selecting only 1500 random samples from each class. The projections are visualized in Fig. 8b. The PCA projection introduces significant amount of overlap between classes and the cost associated with this projection is $\mathcal{L} = 4.4$. The cost for a much more unique t-SNE map, with clearly separated classes, is lower, $\mathcal{L} = 1.7$. Figure 8c additionally shows a comparison of the $\hat{\mathcal{D}}(\sigma)$ curves corresponding to the 2D PCA projection and the 2D t-SNE map, showing that the $\hat{\mathcal{D}}(\sigma)$ metric detected the significant class overlap in PCA. With some amount of scattered

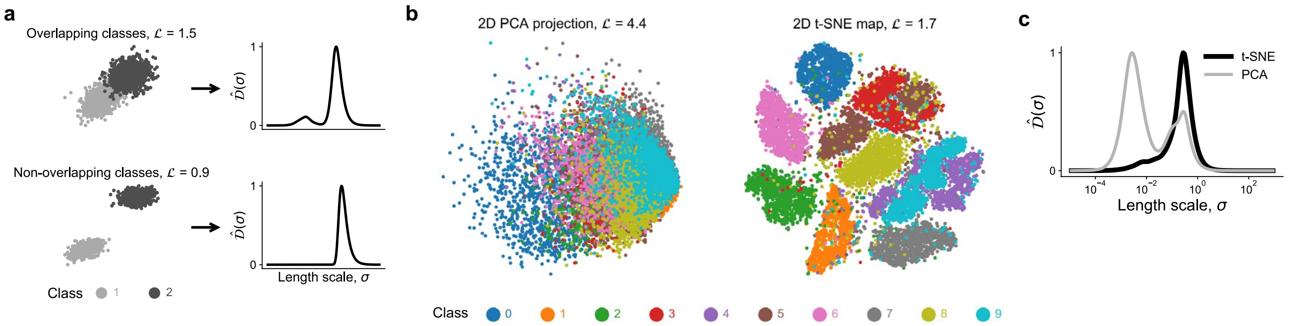

Figure 8. Using the cost function to detect overlap between classes in categorical data. (**a**) Toy dataset composed of two clouds. The first case emulates a projection where two classes overlap, while the second case presents good separation between classes. The cost associated with overlapping classes is higher. (**b**) 2D PCA projection and 2D t-SNE map of the MNIST handwritten digits dataset. With a significant overlap between classes seen in the PCA projection, the associated cost is higher than for the t-SNE map. (**c**) The $\hat{\mathcal{D}}(\sigma)$ curves corresponding to the 2D PCA projection and the 2D t-SNE map of the MNIST handwritten digits dataset.

observations still present in the t-SNE map, the $\hat{\mathcal{D}}(\sigma)$ is not composed of a single rise as was seen in the fully non-overlapping classes example in Fig. 8a. This behavior in $\hat{\mathcal{D}}(\sigma)$ can be exploited to inform the appropriate hyper-parameter tuning to improve t-SNE maps. In the Supplementary material, we further demonstrate the potential of the cost function to guide the selection of an important hyper-parameter of t-SNE called perplexity.

Discussion

Many factors can affect the quality of low-dimensional data parameterizations and there is a need to quantitatively assess those factors from the reduced-order modeling perspective. We propose a cost function that reduces the low-dimensional manifold topology to a single number. The two topological properties that the cost function pays attention to are uniqueness and feature sizes in relevant dependent variable(s). There are two main strengths of the proposed cost function. First, the manifold topology can be optimized for any target dimensionality. Second, the manifold topology can be optimized with respect to any user-selected dependent variables. Only the most important dependent variables need to be included in the manifold topology optimization. This can become particularly helpful in approaches where large state-spaces are compressed to a smaller number of parameters and regression is used to predict a physical quantity based on the compressed representation. Optimal manifolds can be found specifically from the perspective of the physical quantities of interest.

We demonstrate applications on numerically generated datasets and on experimental data containing noise. Our cost function can have useful applications in searching for the best data preprocessing strategies. This can include benchmarking existing strategies with newly invented ones. Often, those settings need to be tailored to a specific dataset and even to the target manifold dimensionality. In addition, various dimensionality reduction and manifold learning techniques can be assessed from the point of view of generating quality manifolds. A possible application might be to optimize the hyper-parameters to result in improved manifold topologies, especially for techniques such as t-SNE or UMAP which are known to strongly depend on hyper-parameter settings. While visual inspection of low-dimensional manifolds can be helpful, our cost function is a quantitative metric that can aid in qualitative assessments, especially when the manifold dimensionality exceeds 3D. Good quality of manifold topology is crucial when nonlinear regression is employed on manifolds. We show that improved projections can bring modeling benefits in regression tasks. We also briefly delineate possible applications of the cost function in dealing with categorical data where the cost function is computed from a dependent variable containing discrete values (class labels). For categorical data, the numerical values for the class labels and the distances between these values will affect the cost. Future work can investigate further the application to categorical data, especially for datasets where the number of classes becomes large or where classes exhibit a meaningful hierarchical structure, such as genomics or transcriptomics data. Future work can also include incorporating the cost function directly as an objective function in dimensionality reduction.

There remain some limitations of our proposed cost function. First, the computational cost for the normalized variance derivative, $\hat{\mathcal{D}}(\sigma)$, becomes high for large datasets. A potential solution to this restriction is to sample the dataset prior to computing the cost function. The Supplementary material sheds some light on how the cost function might respond to data sampling. Second, the current definition of the cost function cannot distinguish between the multiple scales of variation on a unique manifold and the non-uniqueness alone. This is due to the fact that the additional peaks in $\hat{\mathcal{D}}(\sigma)$ will show up in both scenarios, and both will increase the area under the $\hat{\mathcal{D}}(\sigma)$ curve. In our experience, however, the raise in $\hat{\mathcal{D}}(\sigma)$ due to non-uniqueness usually happens for $\sigma \ll \sigma_{peak}$, while the raise in $\hat{\mathcal{D}}(\sigma)$ due to varying feature sizes on an otherwise unique manifold happens at length scales closer σ_{peak} (cf. Fig. 3d,f). The work in Ref.[30] demonstrates how peak locations in $\hat{\mathcal{D}}(\sigma)$ due to non-uniqueness show a sensitivity to data sampling that peak locations due to unique features do not. Future work can consider incorporating information such as this into the cost function to better distinguish manifolds with non-uniqueness from those with small features. Finally, the resulting cost for a given manifold can only be interpreted in relation to cost(s) for other manifold(s). Thus, the cost function can help identify the best manifold among a set of manifolds, but, as of yet, no objective judgment can be made for a single cost value obtained for any one manifold. The same applies to cost associated with any one dependent variable on a single manifold—it should be interpreted in relation to costs for other dependent variables.

Nevertheless, we believe that this is a timely contribution that can help researchers across various disciplines and across numerous applications where low-dimensional data projections play an important role. Although we focused the demonstrations in this paper to four main datasets (argon plasma, reacting flows, atmospheric pollutant dispersion and categorical data), the cost function proposed can have broad applications in virtually any domain of science. We particularly look forward to exploring other applications, for instance on biological or medical data. We argue that further improvements in parameterization quality can be achieved in many areas of research if the low-dimensional parameter space is thoroughly explored and then assessed using the proposed cost function.

Methods

Data normalization.

Prior to applying dimensionality reduction, datasets are often normalized. This can be especially beneficial if the dataset is composed of variables that have very different numerical ranges. Given a dataset composed of Q variables, $\mathbf{X} = [X_1, X_2, \dots, X_Q]$, we normalize each variable X_j by subtracting its center, c_j, and dividing it by the scaling factor, d_j. In a matrix form, we can write the normalized dataset, $\tilde{\mathbf{X}}$, as:

$$\tilde{\mathbf{X}} = (\mathbf{X} - \mathbf{C})\mathbf{D}^{-1}, \tag{1}$$

where $\mathbf{C} \in \mathbb{R}^{N \times Q}$ is a matrix of centers with the jth column of that matrix populated with a value c_j and $\mathbf{D} \in \mathbb{R}^{Q \times Q}$ is a diagonal matrix of scales with the jth element on the diagonal equal to d_j. In this work, we adopt several data scaling criteria collected in Table 1, where s_j is the standard deviation and k_j is the kurtosis of X_j. Techniques S1–S3 are newly introduced scaling techniques that we explore in this work. Throughout the main text, we refer to a particular scaling by its name as per Table 1. The name "None" is equivalent to no scaling applied to the dataset.

Outlier removal.

For the atmospheric pollutant dispersion dataset, we perform outlier detection and removal using the principal component classifier method[53,75]. Outliers are detected based on major and minor principal components (PCs). The observation i is classified as an outlier if the first PC classifier based on the q first (major) PCs:

$$\sum_{j=1}^{q} \frac{z_{ij}^2}{L_j} > c_1 \tag{2}$$

or if the second PC classifier based on the $(Q - k + 1)$ last (minor) PCs:

$$\sum_{j=k}^{Q} \frac{z_{ij}^2}{L_j} > c_2, \tag{3}$$

where z_{ij} is the (i, j)th element from the PCA-transformed data matrix and L_j is the jth eigenvalue from PCA. Major PCs are selected such that the total variance explained is 50% (this determines the number q). Minor PCs are selected such that the remaining variance they explain is 20% (this determines the number k). PCA is performed with Auto scaling. Coefficients c_1 and c_2 are found such that they represent the quantile of the empirical distributions of the first and second PC classifier respectively. Here, we set the quantile to 98%, which allowed to find 509 outlying observations out of 18,540 total observations.

Dimensionality reduction.

A linear projection of a dataset onto a basis defined by $\mathbf{A} \in \mathbb{R}^{Q \times q}$ can be performed as:

$$\boldsymbol{\eta} = \mathbf{XA}, \tag{4}$$

where $\boldsymbol{\eta} = [\eta_1, \eta_2, \dots, \eta_q]$ defines the q-dimensional manifold parameters. In PCA, which is frequently used in this work, the basis matrix $\mathbf{A}$ is computed as the eigenvectors of a data covariance matrix. In the equation above, we assume that the dataset, $\mathbf{X}$, has already been appropriately preprocessed. The summary of linear and nonlinear dimensionality reduction techniques explored in this work is given in Table 2. For results reproducibility, random seed of 100 is used in all techniques that rely on randomness.

PCA.

We use the PCA implementation from the PCAfold Python package[77] developed by the authors.

LDA.

We use the LDA implementation from the scikit-learn library. For the swiss roll dataset example, all parameters are set to default.

DML.

We use the the DML implementation from the pyDML Python package[76]. For the swiss roll dataset example, all parameters are set to default.

MDS.

We use the MDS implementation from the scikit-learn library. For the swiss roll dataset example, all parameters are set to default.

Technique	Type	Most important hyper-parameters
Principal component analysis (PCA)	Linear	None
Linear discriminant analysis (LDA)	Linear	None
Distance metric learning (DML)	Linear	Number of neighbors, distance metric
Multidimensional scaling (MDS)	Nonlinear	Distance metric
Locally linear embedding (LLE)[58]	Nonlinear	Number of neighbors
Hessian locally linear embedding (H-LLE)	Nonlinear	Number of neighbors
Local tangent space alignment (LTSA)	Nonlinear	Number of neighbors
Isomap[57]	Nonlinear	Distance metric
t-Distributed stochastic neighbor embedding (t-SNE)[55]	Nonlinear	Initialization, perplexity, learning rate, early exaggeration
Uniform manifold approximation and projection (UMAP)[60]	Nonlinear	Initialization, number of neighbors, distance metric
Spectral embedding (SE)	Nonlinear	Number of neighbors
Autoencoders (AE)[59]	Linear/nonlinear	Network architecture, activation function(s), learning rate

Table 2. Dimensionality reduction and manifold learning techniques used in this work. In the last column, we list the most important hyper-parameters of each method. The cost function can help fine-tune the hyper-parameters to achieve quality manifold topologies.

LLE, H-LLE, LTSA.

We use the LLE implementation from the scikit-learn library. For the swiss roll dataset example, all parameters are set to default, except the number of neighbors is set to 12. This applies to LLE, H-LLE and LTSA.

Isomap.

We use the Isomap implementation from the scikit-learn library. For the swiss roll dataset example, all parameters are set to default.

t-SNE.

We use the t-SNE implementation from the scikit-learn library. For the atmospheric pollutant dispersion dataset, all hyper-parameters are set to default, except the initialization which is set to PCA initialization. For the MNIST dataset and the swiss roll dataset example, all parameters are set to default.

UMAP.

We use the UMAP implementation from the umap-learn library[60]. For the atmospheric pollutant dispersion dataset, all hyper-parameters are set to default, except the number of neighbors is set to 12. For the swiss roll dataset example, all parameters are set to default.

SE.

We use the SE implementation from the scikit-learn library. For the swiss roll dataset example, all parameters are set to default, except the number of neighbors is set to 12.

Autoencoder.

We use the Keras Python library to set up the autoencoder for the swiss roll dataset example. For results reproducibility, random seed of 100 is used in TensorFlow (`tf.random.set_seed(100)`). The model architecture is 3-2-3 with hyperbolic tangent activations in all layers. The network weights are initialized from the Glorot uniform distribution and the biases are initially set to zeros. We use batch size of 50 and 200 epochs. The Adam optimizer is used with the learning rate 0.001 and the loss function is the mean squared error. We train the autoencoder model on 80% of the data.

The proposed cost function.

The starting point for formulating our cost function is computing the normalized variance proposed by Armstrong and Sutherland[30]. The goal is to assess the low-dimensional parameterization quality defined by q manifold parameters, $\boldsymbol{\eta} \in \mathbb{R}^{N \times q}$, obtained using any dimensionality reduction or manifold learning technique. For length scales on a manifold given by the parameter $\sigma \in \langle \sigma_{min}, \sigma_{max} \rangle$, the normalized variance, $\mathcal{N}(\sigma)$, is computed for the ith dependent variable, ϕ_i, as:

$$\mathcal{N}_i(\sigma) = \frac{\sum_{k=1}^{N} (\phi_{k,i} - \mathcal{K}_i(\eta_k, \sigma))^2}{\sum_{k=1}^{N} (\phi_{k,i} - \bar{\phi_i})^2}, \tag{5}$$

where N is the number of observations in a dataset, $\bar{\phi_i}$ is an arithmetic average of ϕ_i, and $\mathcal{K}_i$ is computed as a weighted average of observations of ϕ_i:

$$\mathcal{K}_i(\eta, \sigma) = \frac{\sum_{j=1}^{N} w_j \phi_{j,i}}{\sum_{j=1}^{N} w_j}, \tag{6}$$

where the weights, w_j, are determined using a Gaussian kernel:

$$w_j = \exp\left(\frac{-||\eta_j - \eta||_2^2}{\sigma^2}\right). \tag{7}$$

In Eq. (7), the quantity $||\eta_j - \eta||_2^2$ is the squared Euclidean distance between the current location on a manifold, η, and any jth point on a manifold, η_j. We then construct the normalized variance derivative function as per[30]:

$$\mathcal{D}_i(\sigma) = \frac{\mathrm{d}\mathcal{N}_i(\sigma)}{\mathrm{d}\log_{10}(\sigma)} + \lim_{\sigma \to 0} \mathcal{N}_i(\sigma) \tag{8}$$

and normalize it by its maximum value:

$$\hat{\mathcal{D}}_i(\sigma) = \frac{\mathcal{D}_i(\sigma)}{\max(\mathcal{D}_i(\sigma))}. \tag{9}$$

The basis for the cost function proposed in this work is integration of the penalized normalized variance derivative function, $\hat{\mathcal{D}}(\sigma)$, over length scales given by σ. For the ith dependent variable, ϕ_i, the area under the $\hat{\mathcal{D}}_i(\sigma)$ curve can be computed as:

$$A_i = \int_{\widetilde{\sigma}_{min,i}}^{\widetilde{\sigma}_{max,i}} \hat{\mathcal{D}}_i(\sigma)\mathrm{d}\widetilde{\sigma}, \tag{10}$$

where the tilde denotes a $\log_{10}$-transformed quantity (e.g. $\widetilde{\sigma} = \log_{10}\sigma$). We compute the area in the $\log_{10}$-space of the length scales σ so that all scales of variation with different orders of magnitude are treated equally. We further introduce two penalties when computing the area:

1. Penalty for peak locations relative to the rightmost peak, σ_{peak}. This favors large feature sizes on a manifold over small ones.
2. Penalty for the area under $\hat{\mathcal{D}}(\sigma)$ happening at $\sigma < \sigma_{peak}$, and especially for $\sigma \ll \sigma_{peak}$. This penalizes multiple scales of variation that might show up as additional peaks in $\hat{\mathcal{D}}(\sigma)$ and becomes particularly useful when these additional peaks can be linked to non-uniqueness.

The cost function proposed herein takes these two penalties into account and is defined for ϕ_i as:

$$\mathcal{L}_i = \int_{\widetilde{\sigma}_{min,i}}^{\widetilde{\sigma}_{max,i}} P_i(\sigma, \sigma_{peak,i}) \cdot \hat{\mathcal{D}}_i(\sigma)\mathrm{d}\widetilde{\sigma}, \tag{11}$$

where $P_i(\sigma, \sigma_{peak,i})$ is the penalty function defined as:

$$P_i(\sigma, \sigma_{peak,i}) = \left|\widetilde{\sigma} - \widetilde{\sigma}_{peak,i}\right|^r + b \cdot \frac{\widetilde{\sigma}_{max,i} - \widetilde{\sigma}_{min,i}}{\widetilde{\sigma}_{peak,i} - \widetilde{\sigma}_{min,i}}. \tag{12}$$

The first term in P_i penalizes non-zero values in $\hat{\mathcal{D}}_i(\sigma)$ at length scales $\sigma < \sigma_{peak}$. It will especially amplify any area under the $\hat{\mathcal{D}}_i(\sigma)$ curve happening for $\sigma \ll \sigma_{peak}$. By increasing the power, the user can increase the amount of penalty for the variance occurring at $\sigma \ll \sigma_{peak}$. The second term in P_i introduces a gentle penalty for the location of the rightmost peak and essentially increases or decreases the entire penalty function by a constant value. This second term acts to reward larger feature sizes as those should be easier to model. The parameter b is another hyper-parameter which controls the amount of the constant vertical shift of the entire penalty function. By increasing b, the user can increase the amount of penalty for the rightmost peak location, σ_{peak}. Throughout this work, we use $r = 1$ and $b = 1$. For completeness, in the next section, we illustrate the effect of setting r and b to values other than unity.

Python implementation of $\mathcal{N}(\sigma)$, $\hat{\mathcal{D}}(\sigma)$ and the proposed cost function, $\mathcal{L}$, has been developed by the authors and is available in the PCAfold software package[77]. In practice, numeric integration of $\hat{\mathcal{D}}(\sigma)$ is performed using a composite trapezoid rule. The peak values in $\hat{\mathcal{D}}(\sigma)$ are computed using the `scipy.find_peaks` function, which finds all local maxima by comparing the neighborhood of all discrete values of $\hat{\mathcal{D}}(\sigma)$. The value for σ_{peak} is taken as the rightmost peak found by `scipy.find_peaks`, but the user can also select a percentage, p, of the value σ_{peak} (such that the true rightmost peak becomes $p(\sigma_{max} - \sigma_{peak})$). Our recommended range for the parameter σ is $\langle 10^{-7}; 10^3 \rangle$, with logarithmically-spaced in-between values. For instance, throughout this work we typically set `sigma = numpy.logspace(-7, 3, 200)`. Throughout this work, the manifold parameters $\boldsymbol{\eta}$ are scaled to a unit box (each η_i is scaled to a [0, 1] range) before computing the cost. This is done so that the length scale, σ, has the same meaning in each manifold dimension.

Effect of hyper-parameters *r* and *b* on the cost function. The hyper-parameters r and b from Eq. (12) may be changed to emphasize penalties applied to non-uniqueness and/or small feature sizes when computing overall costs. Figure 9 illustrates the effect of setting r and b to non-unity values. In Fig. 9a–c, we utilize the toy functions from Fig. 3 and show how $\mathcal{L}$ behaves for various r and b. For clearer analysis, whenever r is varied, b is set to unity, and vice versa. For comparison, with the red dashed lines, we mark costs corresponding to $r = 1$ and $b = 1$. In principle, increasing b increases $\mathcal{L}$ more severely for manifolds where a dependent variable exhibits small feature sizes (see Fig. 9a). Conversely, increasing r for unique manifolds does not affect the $\mathcal{L}$ values significantly. As can be understood from Fig. 9c, increasing r increases $\mathcal{L}$ more severely for manifolds with

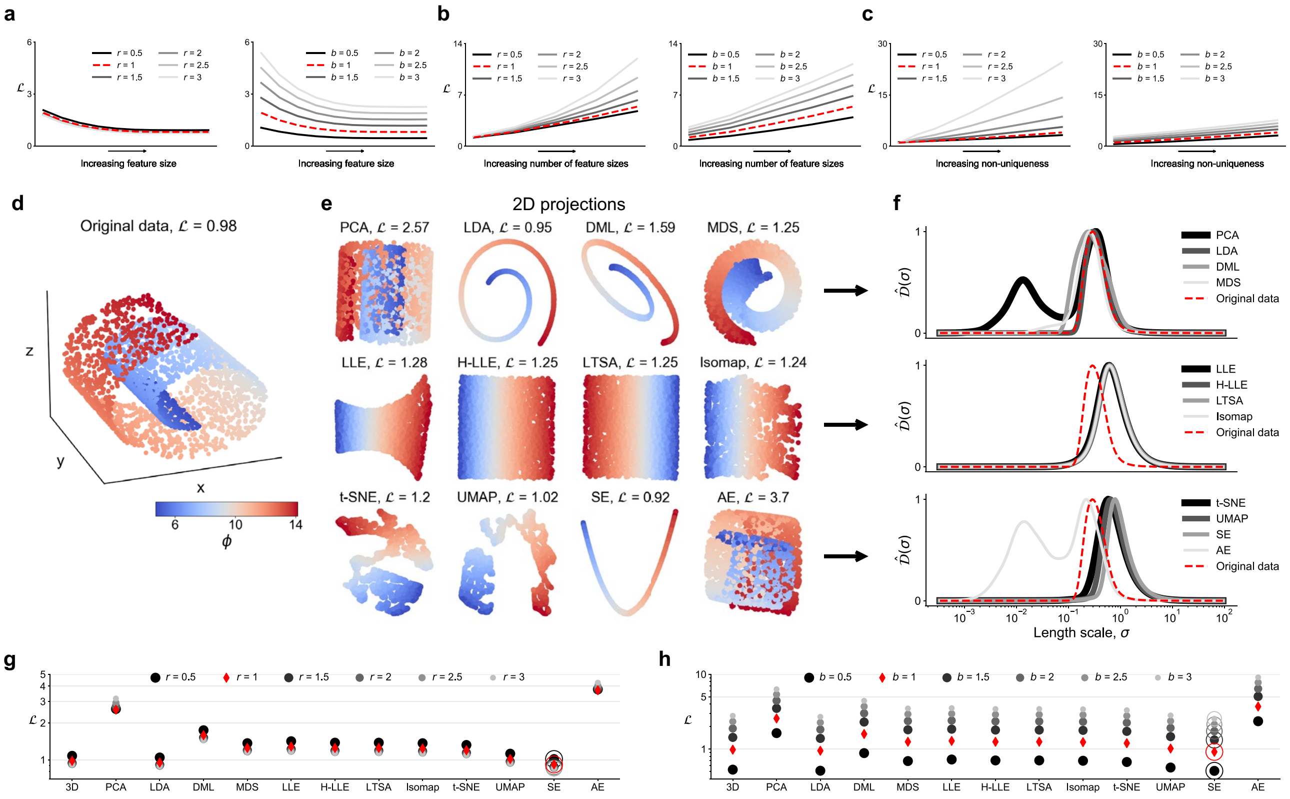

Figure 9. Illustrative demonstration of the effect of changing the cost function's hyper-parameters, r and b. (**a–c**) Costs corresponding to toy functions from Fig. 3 when setting r and b to non-unity values. (**d**) The original 3D swiss roll dataset with the cost reported for the original 3D parameterization using $r = 1$ and $b = 1$. (**e**) 2D swiss roll dataset projections with the resulting costs using $r = 1$ and $b = 1$. (**f**) Comparison of the $\hat{\mathcal{D}}(\sigma)$ curves between generated 2D swiss roll dataset projections. The $\hat{\mathcal{D}}(\sigma)$ curve for the original 3D parameters is shown with the red dashed line for reference. (**g**) Costs for the 3D swiss roll data parameters and its various 2D projections with b set fixed to unity, and with changing r. (**h**) Costs for the 3D swiss roll data parameters and its various 2D projections with r set fixed to unity, and with changing b. With the circular outlines in (**g,h**), we mark the lowest costs for any of the 2D projections which consistently happens for the SE manifold, irrespective of the values selected for r and b.

non-uniqueness. Even if non-uniqueness is present on a manifold with a fixed largest feature size, changing b does not affect the $\mathcal{L}$ values significantly. For a unique manifold with multiple feature sizes, increasing r or b can increase $\mathcal{L}$ by a similar amount (Fig. 9b). Due to the blending of multiple scales of variation in the $\hat{\mathcal{D}}(\sigma)$ profile seen in Fig. 3d, we have also applied a $p = 70\%$ shift in σ_{peak} in Fig. 9b, as per discussion in the previous section. In Fig. 9a–c, the overall trend of the cost function is preserved when changing r and b.

We further test the impact of r and b on the verdict given by the cost function across various dimensionality reduction and manifold learning techniques. We use the classic 3D swiss roll dataset and generate various 2D projections. The original 3D topology of this dataset can be seen in Fig. 9d, colored by a dependent variable, ϕ. The cost corresponding to the parameterization in the original 3D space is $\mathcal{L} = 0.98$ with $r = 1$ and $b = 1$. Figure 9e shows twelve different 2D projections of that dataset generated with various linear and nonlinear techniques. The cost value, $\mathcal{L}$, is reported for each projection taking $r = 1$ and $b = 1$. Among all the reduction techniques selected for this demonstration, PCA and a linear autoencoder (AE) introduce the most significant non-uniqueness. The costs for AE and PCA projections are the highest, while SE, LDA and UMAP show the lowest costs. The small cost for SE, LDA or UMAP projections (comparable to the cost in the original 3D space) is likely due to a combination of two factors: projection uniqueness and large feature sizes created through sufficient separation of distinct ϕ values in space. Figure 9f shows the $\hat{\mathcal{D}}(\sigma)$ curves for the original 3D data (red dashed lines) and for the 2D projections, in groups of four. The location of σ_{peak} is shifted to the right for the SE and UMAP projections with respect to the original 3D data parameters. A similar observation holds for LLE, H-LLE, LTSA and Isomap projections. This can be due to a relatively large distance between the smallest and the largest ϕ values on these projections. In the original 3D space, the high and low values of ϕ are relatively close to each other due to manifold curvature. In Fig. 9g,h, we show the effect on the cost function evaluations of the various swiss roll data projections when changing the hyper-parameters r and b. In Fig. 9g, costs increase with increasing r only for the PCA and AE projections. This is understandable, since those are the only projections with significant levels of non-uniqueness. Increasing the hyper-parameter r thus increasingly amplifies the "secondary" rise seen in the $\hat{\mathcal{D}}(\sigma)$ profile at $\sigma < \sigma_{peak}$ for PCA and AE. Interestingly, this behavior with r can potentially be used to detect manifolds with non-uniqueness among a set of parameterizations. In Fig. 9h, increasing b increases costs for all parameterizations (3D and 2D). Costs increase more rapidly for PCA and AE projections than for any other explored projection (note the logarithmic scale of the vertical axis). This is also due to emphasizing the "secondary" rise seen in the $\hat{\mathcal{D}}(\sigma)$ profile when computing the penalized area, but this time through increasing the vertical shift of the entire penalty function. With the circled outlines in Fig. 9g,h, we mark the lowest cost that happened for any of the 2D projections across the explored values of r and b. Among the r and b values explored, the smallest cost consistently happens for the SE projection. We also note that the ratio between $\mathcal{L}$ for the worst projection (AE) and $\mathcal{L}$ for the best projection (SE) is generally higher when r is set above unity than when b is set above unity. Thus, increasing r above unity can help create clearer separations in $\mathcal{L}$ values when ranking manifolds with varying levels of non-uniqueness.

On the computational impact of evaluating the cost function.

We note that the computational time required to compute $\mathcal{L}$ for a dataset with N observations and for m dependent variables scales as $\mathcal{O}(mN^2)$, assuming the same number of discrete values of σ. The code for computing the cost function, that we provide in the PCAfold library, is parallelized with respect to σ, since the normalized variance computations are entirely independent of one another for different values of σ. Thus, our code can be readily ran on multiple CPUs, which we generally recommend for $N > 10^5$. In the Supplementary material, we show the effect of data sampling on the cost function. We note, that while subsampling the data can ease the computational cost of $\mathcal{L}$, it has an enhanced effect on dependent variables which exhibit variation at multiple scales (e.g. from non-uniqueness). A possible future implementation that could reduce the computational time of evaluating Eq. (6) is to approximate the weighted average with a sum over points within/near the currently considered length scale, σ. This implementation can make the computational time scale approximately as $\mathcal{O}(mN)$ for small σ and only scale as $\mathcal{O}(mN^2)$ as σ encompasses all points on a manifold.

There is a second aspect to the question of the computational cost, that is more elusive to quantify, and that is the time required for the researcher to find a good manifold topology through trial and error exploration, if no quantitative tools were applied to guide the choice of a manifold. We argue that this second aspect can become a bottleneck in effective application of reduced-order models. In this regard, one may find that our cost function allows for "quick" assessments of projections resulting from a range of dimensionality reduction and manifold learning techniques, as well as a variety of data preprocessing strategies applied to the training data. Moreover, since $\mathcal{L}$ can be implemented in optimization tasks, it is likely that an optimum in manifold topologies is found in an automated way, compared to a trial and error approach.

$\mathcal{L}$-informed feature selection.

We developed a feature selection algorithm that iteratively eliminates state variables from the dataset based on minimizing the cost, $\mathcal{L}$, of PCA projections[65]. There are three inputs to the algorithm: the original dataset, $\mathbf{X} \in \mathbb{R}^{N \times Q}$, the target dependent variables, $\boldsymbol{\phi} \in \mathbb{R}^{N \times m}$, and the target manifold dimensionality, q. The pseudocode below shows in closer detail how the algorithm computes the optimized subset of the original state variables. At each iteration i, the algorithm computes PCA projections resulting from removing each variable, one at a time. At the end of the iteration, the variable whose removal decreased the cost value the most is discarded from the dataset. At the next iteration, the process repeats, but now on a dataset with one less variable. We only allow $Q - q$ iterations so that we never reduce original data dimensionality below q requested by the user. Once all iterations have finished, the algorithm looks back at final costs from all iterations and returns the optimized subset, $\mathbf{X}_S$, corresponding to the iteration that showed the minimum cost value. Such subset should then generate an optimized manifold topology. The algorithm is available in the PCAfold software package[77].

```
Input: $\mathbf{X} \in \mathbb{R}^{N \times Q}$, $\boldsymbol{\phi} \in \mathbb{R}^{N \times m}$, $q$;
Output: $\mathbf{X}_S \in \mathbb{R}^{N \times S}$;
for $i \in \{1...Q - q\}$ do
    for $j \in \{1...Q - i + 1\}$ do
        Compute $q$-dimensional PCA projection of a dataset without variable $X_j$;
        Compute $\mathcal{L}_j$ for that projection for the set $\boldsymbol{\phi}$;
    end
    Compute $\mathcal{L}_i = \min(\mathcal{L}_j)$;
    Find variable $X_k$ whose removal corresponds to the cost $\mathcal{L}_i$;
    Remove $X_k$ from the dataset;
end
Find subset, $\mathbf{X}_S$, corresponding to the iteration where $\mathcal{L} = \min(\mathcal{L}_i)$;
Return $\mathbf{X}_S$;
```

Nonlinear regression using artificial neural networks (ANNs).

In the atmospheric pollutant dispersion example, we use nonlinear regression using ANNs. We use the Keras Python library to set up the ANN regression. For results reproducibility, random seed of 100 is used in TensorFlow (`tf.random.set_seed(100)`). The inputs of the network are the three manifold parameters, $\boldsymbol{\eta} = [\eta_1, \eta_2, \eta_3]$, either from PCA, UMAP, Isomap or t-SNE. The output is the turbulent Schmidt number, $\phi = Sc_t$. The model architecture is 2-5-5-1 for 2D projections and 3-5-5-1 for 3D projections with sigmoid activations in all layers except the output layer, where we use linear activation. The network weights are initialized from the Glorot uniform distribution and the biases are initially set to zeros. We use batch size of 100, validation split of 0.2 and 500 epochs. The Adam optimizer is used with the learning rate 0.001 and the loss function is the mean squared error. We train the regression model on 80% of the data and measure the MAE on the remaining 20% test data not seen by the ANN model. The results reported in Fig. 7 are for the 20% test data.

Nonlinear regression using kernel regression.

In the atmospheric pollutant dispersion example, we use kernel regression from the PCAfold software package[77]. We use the Nadaraya-Watson estimator with a Gaussian kernel with a varying bandwidth. The bandwidth is determined locally based on 50 nearest neighbors of the query point. We train the regression model on 80% of the data and measure the MAE on the remaining 20% test data not seen by the kernel regression model. The results reported in Fig. 7 are for the 20% test data.

Reacting flow data generation.

The reacting flow datasets for combustion of syngas in air and hydrogen in air were generated using Spitfire Python package[78] available at: `github.com/sandialabs/Spitfire`. The datasets were generated using a steady laminar flamelet model for a range of dissipation rates from chemical equilibrium to extinction and a range of mixture fractions between 0 and 1. For the syngas/air dataset, the fuel stream is composed of a mixture of carbon monoxide and hydrogen in 10:1 molar proportion[79]. The oxidizer stream is air. Both streams have initial temperature 300 K and pressure 101,325 Pa. For the hydrogen/air dataset, the fuel stream is composed of hydrogen and the oxidizer stream is air[80]. Both streams have initial temperature 300 K and pressure 101,325 Pa. For both fuels, the mixture fraction grid is denser closer to the stoichiometric conditions.

Data availability

The datasets generated during or analyzed during the current study are available in the GitHub repository: https://github.com/kamilazdybal/cost-function-manifold-assessment. The atmospheric pollutant dispersion dataset[46] and the argon plasma dataset[66,67] are property of Université libre de Bruxelles. The Sandia flames data[69] can be accessed at: `tnfworkshop.org/data-archives/pilotedjet/ch4-air`.

Code availability

The code used during the current study is publicly available in the GitHub repository: https://github.com/kamilazdybal/cost-function-manifold-assessment.

Received: 2 May 2022; Accepted: 17 August 2022
Published online: 25 August 2022

References

1. Noack, B. R., Afanasiev, K., Morzyński, M., Tadmor, G. & Thiele, F. A hierarchy of low-dimensional models for the transient and post-transient cylinder wake. *J. Fluid Mech.* **497**, 335–363 (2003).
2. Singer, A., Erban, R., Kevrekidis, I. G. & Coifman, R. R. Detecting intrinsic slow variables in stochastic dynamical systems by anisotropic diffusion maps. *Proc. Natl. Acad. Sci.* **106**(38), 16090–16095 (2009).
3. Lusch, B., Kutz, J. N. & Brunton, S. L. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.* **9**(1), 1–10 (2018).
4. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).
5. Mendez, M. A., Balabane, M. & Buchlin, J.-M. Multi-scale proper orthogonal decomposition of complex fluid flows. *J. Fluid Mech.* **870**, 988–1036 (2019).
6. Farooq, H., Saeed, A., Akhtar, I. & Bangash, Z. Neural network-based model reduction of hydrodynamics forces on an airfoil. *Fluids* **6**(9), 332 (2021).
7. Willett, F. R., Avansino, D. T., Hochberg, L. R., Henderson, J. M. & Shenoy, K. V. High-performance brain-to-text communication via handwriting. *Nature* **593**(7858), 249–254 (2021).
8. Martens, C. *et al.* Voxelwise principal component analysis of dynamic [S-methyl-11C] methionine PET data in glioma patients. *Cancers* **13**(10), 2342 (2021).
9. Maas, U. & Pope, S. B. Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space. *Combust. Flame* **88**(3), 239–264 (1992).
10. Holmes, P. J., Lumley, J. L., Berkooz, G., Mattingly, J. C. & Wittenberg, R. W. Low-dimensional models of coherent structures in turbulence. *Phys. Rep.* **287**(4), 337–384 (1997).
11. Lowe, R. & Tomlin, A. Low-dimensional manifolds and reduced chemical models for tropospheric chemistry simulations. *Atmos. Environ.* **34**(15), 2425–2436 (2000).
12. Tomlin, A. S., Whitehouse, L., Lowe, R. & Pilling, M. J. Low-dimensional manifolds in tropospheric chemical systems. *Faraday Discuss.* **120**, 125–146 (2002).
13. Yang, Y., Pope, S. B. & Chen, J. H. Empirical low-dimensional manifolds in composition space. *Combust. Flame* **160**(10), 1967–1980 (2013).
14. Mendez, M. A., Scelzo, M. T. & Buchlin, J.-M. Multiscale modal analysis of an oscillating impinging gas jet. *Exp. Therm. Fluid Sci.* **91**, 256–276 (2018).
15. Shine, J. M. *et al.* Human cognition involves the dynamic integration of neural activity and neuromodulatory systems. *Nat. Neurosci.* **22**(2), 289–296 (2019).
16. Verma, A. & Engelhardt, B. E. A robust nonlinear low-dimensional manifold for single cell RNA-seq data. *BMC Bioinform.* **21**(1), 1–15 (2020).
17. Brunton, S. L., Noack, B. R. & Koumoutsakos, P. Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508 (2020).
18. Nieh, E. H. *et al.* Geometry of abstract learned knowledge in the hippocampus. *Nature* **595**(7865), 80–84 (2021).
19. Recanatesi, S. *et al.* Predictive learning as a network mechanism for extracting low-dimensional latent space representations. *Nat. Commun.* **12**(1), 1–13 (2021).
20. Iyer, K. K. *et al.* Focal neural perturbations reshape low-dimensional trajectories of brain activity supporting cognitive performance. *Nat. Commun.* **13**(1), 1–8 (2022).

21. Noack, B. R. & Eckelmann, H. A low-dimensional Galerkin method for the three-dimensional flow around a circular cylinder. *Phys. Fluids* **6**(1), 124–143 (1994).
22. Sutherland, J. C. & Parente, A. Combustion modeling using principal component analysis. *Proc. Combust. Inst.* **32**(1), 1563–1570 (2009).
23. Callaham, J. L., Koch, J. V., Brunton, B. W., Kutz, J. N. & Brunton, S. L. Learning dominant physical processes with data-driven balance models. *Nat. Commun.* **12**(1), 1–10 (2021).
24. Cenedese, M., Axås, J., Bäuerlein, B., Avila, K. & Haller, G. Data-driven modeling and prediction of non-linearizable dynamics via spectral submanifolds. *Nat. Commun.* **13**, 1–13 (2022).
25. Zhang, Y. *et al.* Large eddy simulation of spray combustion using flamelet generated manifolds combined with artificial neural networks. *Energy AI* **2**, 100021 (2020).
26. Lee, J. A. & Verleysen, M. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing* **72**(7–9), 1431–1443 (2009).
27. Venna, J., Peltonen, J., Nybo, K., Aidos, H., Kaski, S. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *J. Mach. Learn. Res.* **11**(2) (2010).
28. Zhang, P., Ren, Y. & Zhang, B. A new embedding quality assessment method for manifold learning. *Neurocomputing* **97**, 251–266 (2012).
29. Mokbel, B., Lueks, W., Gisbrecht, A. & Hammer, B. Visualizing the quality of dimensionality reduction. *Neurocomputing* **112**, 109–123 (2013).
30. Armstrong, E. & Sutherland, J. C. A technique for characterising feature size and quality of manifolds. *Combust. Theory Modell.* **25**, 1–23 (2021).
31. Amir, E. D. *et al.* viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nat. Biotechnol.* **31**(6), 545–552 (2013).
32. Belkina, A. C. *et al.* Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nat. Commun.* **10**(1), 1–12 (2019).
33. Moon, K. R. *et al.* Visualizing structure and transitions in high-dimensional biological data. *Nat. Biotechnol.* **37**(12), 1482–1492 (2019).
34. Kobak, D. & Berens, P. The art of using t-SNE for single-cell transcriptomics. *Nat. Commun.* **10**(1), 1–14 (2019).
35. Kobak, D. & Linderman, G. C. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nat. Biotechnol.* **39**(2), 156–157 (2021).
36. Becht, E. *et al.* Dimensionality reduction for visualizing single-cell data using UMAP. *Nat. Biotechnol.* **37**(1), 38–44 (2019).
37. Szubert, B., Cole, J. E., Monaco, C. & Drozdov, I. Structure-preserving visualisation of high dimensional single-cell datasets. *Sci. Rep.* **9**(1), 1–10 (2019).
38. Dorrity, M. W., Saunders, L. M., Queitsch, C., Fields, S. & Trapnell, C. Dimensionality reduction by UMAP to visualize physical and genetic interactions. *Nat. Commun.* **11**(1), 1–6 (2020).
39. Cohen, U., Chung, S. Y., Lee, D. D. & Sompolinsky, H. Separability and geometry of object manifolds in deep neural networks. *Nat. Commun.* **11**(1), 1–13 (2020).
40. Shine, J. M. *et al.* The low-dimensional neural architecture of cognitive complexity is related to activity in medial thalamic nuclei. *Neuron* **104**(5), 849–855 (2019).
41. Chaudhuri, R., Gerçek, B., Pandey, B., Peyrache, A. & Fiete, I. The intrinsic attractor manifold and population dynamics of a canonical cognitive circuit across waking and sleep. *Nat. Neurosci.* **22**(9), 1512–1520 (2019).
42. Ju, H. & Bassett, D. S. Dynamic representations in networked neural systems. *Nat. Neurosci.* **23**(8), 908–917 (2020).
43. Nilsson, J., Sha, F., Jordan, M. I. Regression on manifolds using kernel dimension reduction. In *Proceedings of the 24th International Conference on Machine learning* 697–704 (2007).
44. Malik, M. R., Isaac, B. J., Coussement, A., Smith, P. J. & Parente, A. Principal component analysis coupled with nonlinear regression for chemistry reduction. *Combust. Flame* **187**, 30–41 (2018).
45. Dalakoti, D. K., Wehrfritz, A., Savard, B., Day, M. S., Bell, J. B., & Hawkes, E. R. An a priori evaluation of a principal component and artificial neural network based combustion model in diesel engine conditions. In *Proceedings of the Combustion Institute* (2020).
46. Longo, R., Bellemans, A., Derudi, M. & Parente, A. A multi-fidelity framework for the estimation of the turbulent Schmidt number in the simulation of atmospheric dispersion. *Build. Environ.* **185**, 107066 (2020).
47. Malpica Galassi, R., Ciottoli, P. P., Valorani, M. & Im, H. G. Local combustion regime identification using machine learning. *Combust. Theory Modell.* **26**, 1–17 (2021).
48. Dominique, J., Van den Berghe, J., Schram, C. & Mendez, M. A. Artificial neural networks modeling of wall pressure spectra beneath turbulent boundary layers. *Phys. Fluids* **34**(3), 035119 (2022).
49. Keun, H. C. *et al.* Improved analysis of multivariate data by variable stability scaling: Application to NMR-based metabolic profiling. *Anal. Chim. Acta* **490**(1–2), 265–276 (2003).
50. Keenan, M. R. & Kotula, P. G. Accounting for poisson noise in the multivariate analysis of ToF-SIMS spectrum images. *Surf. Interface Anal. Int. J. Devoted Dev. Appl. Tech. Anal. Surf. Interfaces Thin Films* **36**(3), 203–212 (2004).
51. van den Berg, R. A., Hoefsloot, H. C. J., Westerhuis, J. A., Smilde, A. K. & van der Werf, M. J. Centering, scaling, and transformations: Improving the biological information content of metabolomics data. *BMC Genom.* **7**(1), 1–15 (2006).
52. Noda, I. Scaling techniques to enhance two-dimensional correlation spectra. *J. Mol. Struct.* **883–884**, 216–227 (2008) (**Progress in two-dimensional correlation spectroscopy**).
53. Parente, A. & Sutherland, J. C. Principal component analysis of turbulent combustion data: Data pre-processing and manifold sensitivity. *Combust. Flame* **160**(2), 340–350 (2013).
54. Frénay, B., Doquire, G. & Verleysen, M. Is mutual information adequate for feature selection in regression?. *Neural Netw.* **48**, 1–7 (2013).
55. Van der Maaten, L. & Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**(11) (2008).
56. Karniadakis, G. E. *et al.* Physics-informed machine learning. *Nat. Rev. Phys.* **3**(6), 422–440 (2021).
57. Tenenbaum, J. B., de Silva, V. & Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500), 2319–2323 (2000).
58. Roweis, S. T. & Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**(5500), 2323–2326 (2000).
59. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-propagating errors. *Nature* **323**(6088), 533–536 (1986).
60. McInnes, L., Healy, J., & Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint*arXiv:1802.03426 *(2018).*
61. Wattenberg, M., Viégas, F. & Johnson, I. How to use t-SNE effectively. *Distill* **1**, e2 (2016).
62. Fooladgar, E. & Duwig, C. A new post-processing technique for analyzing high-dimensional combustion data. *Combust. Flame* **191**, 226–238 (2018).
63. Grollemund, V. *et al.* Development and validation of a 1-year survival prognosis estimation model for Amyotrophic Lateral Sclerosis using manifold learning algorithm UMAP. *Sci. Rep.* **10**(1), 1–12 (2020).
64. Gundogdu, P., Loucera, C., Alamo-Alvarez, I., Dopazo, J. & Nepomuceno, I. Integrating pathway knowledge with deep neural networks to reduce the dimensionality in single-cell RNA-seq data. *BioData Mining* **15**(1), 1–21 (2022).

65. Zdybał, K., Sutherland, J. C., & Parente, A. Manifold-informed state vector subset for reduced-order modeling. In *Proceedings of the Combustion Institute* 39 (2022).
66. Bellemans, A., Munafo, A., Magin, T. E., Degrez, G. & Parente, A. Reduction of a collisional-radiative mechanism for argon plasma based on principal component analysis. *Phys. Plasmas* **22**(6), 062108 (2015).
67. Bellemans, A., Magin, T., Coussement, A. & Parente, A. Reduced-order kinetic plasma models using principal component analysis: Model formulation and manifold sensitivity. *Phys. Rev. Fluids* **2**(7), 073201 (2017).
68. Hastie, T., Tibshirani, R., & Friedman, J. H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. (Springer, 2009).
69. Barlow, R. S. & Frank, J. H. Effects of turbulence on species mass fractions in methane/air jet flames. In *Symposium (International) on Combustion*, vol. 27(1), 1087–1095 (1998).
70. Ihme, M., Shunn, L. & Zhang, J. Regularization of reaction progress variable for application to flamelet-based combustion models. *J. Comput. Phys.* **231**(23), 7715–7721 (2012).
71. Pope, S. B. Small scales, many species and the manifold challenges of turbulent combustion. *Proc. Combust. Inst.* **34**(1), 1–31 (2013).
72. CEDVAL at Hamburg University Compilation of Experimental Data for Validation of Microscale Dispersion Models. Website provided by the environmental wind tunnel laboratory (EWTL) of the meteorological institute (2021).
73. Duraisamy, K., Iaccarino, G. & Xiao, H. Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**, 357–377 (2019).
74. Deng, L. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Process. Mag.* **29**(6), 141–142 (2012).
75. Shyu, M. L., Chen, S. C., Sarinnapakorn, K., & Chang, L. W. A novel anomaly detection scheme based on principal component classifier. Technical report, Miami Univ Coral Gables Fl Dept of Electrical and Computer Engineering (2003).
76. Suárez, J. L., García, S. & Herrera, F. pyDML: A Python library for distance metric learning. *J. Mach. Learn. Res.* **21**(96), 1–7 (2020).
77. Zdybał, K., Armstrong, E., Parente, A. & Sutherland, J. C. PCAfold: Python software to generate, analyze and improve PCA-derived low-dimensional manifolds. *SoftwareX* **12**, 100630 (2020).
78. Hansen, M. A., Armstrong, E., Sutherland, J. C., McConnell, J., Hewson, J. C. & Knaus, R. (Spitfire, 2020).
79. Hawkes, E. R., Sankaran, R., Sutherland, J. C. & Chen, J. H. Scalar mixing in direct numerical simulations of temporally evolving plane jet flames with skeletal CO/H2 kinetics. *Proc. Combust. Inst.* **31**(1), 1633–1640 (2007).
80. Burke, M. P., Chaos, M., Ju, Y., Dryer, F. L. & Klippenstein, S. J. Comprehensive H2/O2 kinetic model for high-pressure combustion. *Int. J. Chem. Kinet.* **44**(7), 444–474 (2012).

Acknowledgements

The authors would like to thank Miguel A. Mendez for valuable discussions. K.Z. acknowledges the support of F.R.S.-FNRS. E.A. acknowledges the support of Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. Aspects of this material are based upon work supported by the National Science Foundation under Grant No. 1953350. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program under grant agreement no. 714605.

Author contributions

K.Z., E.A. and J.C.S. conceptualized the initial ideas; K.Z., E.A., J.C.S., and A.P. designed the research; K.Z. and E.A. developed the code, performed simulations, analyzed the results and wrote the first draft of the manuscript; K.Z., E.A., J.C.S. and A.P. reviewed and edited the manuscript; J.C.S. and A.P. supervised the work.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-022-18655-1.

Correspondence and requests for materials should be addressed to K.Z., J.C.S. or A.P.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.