On Different Choice of Basis for Reduced Order Models

Yanjie Tong, Zhiyu Xu

Abstract

Reduced order models (ROM) are widely adopted to fast recover a flow field. In ROM, a key part is to obtain the score and basis. It is believed that non-linear information extraction and covariates, like the location of elements, would make the score and basis more informative, achieving better performance than linearity-constraint Principal Component Analysis (PCA). In this work, kernel Principal Component Analysis (kPCA) and Graph Regularized Non-negative Matrix Factorization (GNMF) are adopted to obtain interpretable non-linear or covariate-assisted basis. The result shows that PCA still outperforms these methods. We discuss possible reasons for this phenomenon.

1 Introduction

Reduced order models (ROM) are widely adopted to fast recover a flow field. To reduce the dimension for prediction, Principal Component Analysis (PCA) is traditionally preferred. However, due to the linear property of PCA, it can only extract linear information from the data. Furthermore, PCA treats every node independently. Physically speaking, when two nodes are physically close to each other, their flow properties should be similar. Recent literature in genetics (Xu and McCord 2022; Miao et al. 2021) also show superiority by integrating information from different sources like scATAC-seq, scRNA-seq data in cell type clustering. Hence, we want to incorporate the location information of each node as well as multiple physical properties like pressure, velocity and temperature into consideration to construct the basis simultaneously, in order to achieve better performance. However, due to the limited performance of our initial attempts, data integration is not performed under this task.

Based on these two intuitions, our attempts in this project are 2-fold.

1. We want to extend the basis to non-linear decompositions via kernel Principal Component Analysis (kPCA).

2. We want to incorporate location information into basis construction through several methods.

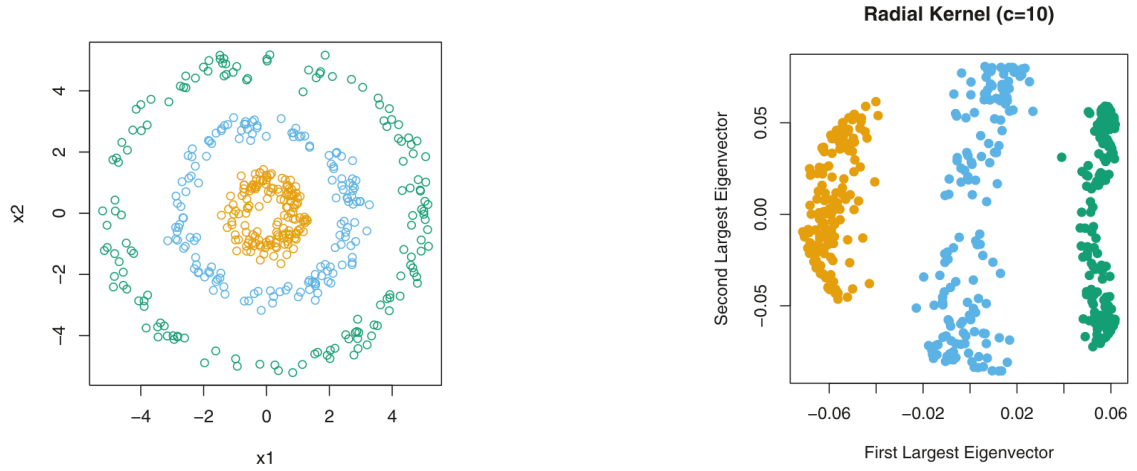
Other than these two thoughts, we also tried HeteroPCA (Zhang, T. T. Cai, and Wu 2021), which is a novel principal component analysis method that provides better performance in several tasks. Because its performance is not significant in this task, we omit the introduction here.

In this section, we will introduce several matrix decomposition methods that may enable nonlinear principal component extraction or incorporate covariate information. In Section 2, we will compare different methods using a classical problem, flow around a circular cylinder. In the Section 3, we will discuss the implication of experiments and point out possible future directions.

1.1 Kernel Principal Component Analysis

Thanks to the development of kernel methods like support vector machines, kernel Principal Component Analysis (kPCA) (Schölkopf, Smola, and Müller 1997) has utilized the non-linear property of kernel methods to help find non-linear principal components of data. Kernel methods project data into a higher dimension where linearity may include more information and implies non-linearity in the original dimension. Since PCA only needs specification of the inner product between two objects, only the inner product form of the projection is needed, instead of a specified projection function, which makes kernel methods easily employable in PCA. Figure 1 (Hastie et al. 2009) illustrates the non-linear effects of kPCA. The data in Figure 1a has three labels in concentric circles with different diameter. If PCA is applied, the non-linearity of the data would prevent the separation of labels in principal space.

However, with proper parameters and kernels in kernel PCA, they can be well separated in the largest eigenvector as shown in Figure 1b.



(a) Toy Example with three different labels in concentric circles

(b) kernel PCA with radial basis kernel with proper parameters

Figure 1: Illustration of kernel PCA using a toy example

Similar to other kernel methods, kernel PCA enjoys a wide range of kernels to choose from. Shawe-Taylor and Cristianini (2004) provides a rather comprehensive review of commonly adopted kernels. The most widely applied kernel is probably radial kernels,

$$K(x, x') = \exp\left(-\|x - x'\|_2^2/c\right),$$

where c is a tuning parameter controlling the sensitivity of the kernel. Other commonly applied kernels like polynomial kernels can be considered as a special case of the radial kernel due to Taylor Expansion, and often the radial kernel's performance is better than polynomial kernels in Support Vector Machines. Hence, we here only apply the radial kernel for further investigation. There are also literature illustrating how to incorporate domain information like invariance to perform kernel engineering (Decoste and Schölkopf 2002; Schölkopf, Simard, et al. 1997). However, we are dealing with steady state flow field data in this case, and fail to find invariance properties in steady state flow field.

The algorithm of kPCA is presented in Algorithm 1. For notational simplicity, $\mathbf{1}_{1/N}$ denotes the matrix with each entry being $1/N$.

Algorithm 1 kernel PCA algorithm

Input: data $S \in \mathbb{R}^{N \times J}$, dimension K, kernel κ
Output: principal space U
1: $\mathbf{K}_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, $i, j = 1, \cdots, N$
2: $\mathbf{K} = \mathbf{K} - \mathbf{1}_{1/N}\mathbf{K} - \mathbf{K}\mathbf{1}_{1/N} + \mathbf{1}_{1/N}\mathbf{K}\mathbf{1}_{1/N}$
3: $[\mathbf{U}, \mathbf{\Lambda}] = \text{eig}(\mathbf{K})$

1.2 Graph Regularized Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) (D. Lee and Seung 2000) is an unsupervised matrix factorization algorithm for non-negative matrices to obtain non-negative decompositions. It enjoys the advantage of "learning by parts" and has superior performance in facial recognition than PCA (D. D. Lee and Seung 1999). We adopt the original multiplicative update formula as initially proposed in D. Lee and Seung (2000), and denote data $\mathbf{X} = \mathbf{US}$, where $\mathbf{S}$ is the score and $\mathbf{U}$ is the basis. There are two forms of objective functions. One is to optimize the L_2 norm, $\|\mathbf{X} - \mathbf{US}\|_2$. The other is to

optimize divergence defined as,

$$D(\mathbf{A}||\mathbf{B}) = \sum_{i,j} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right),$$

where in this case, $\mathbf{A} = \mathbf{X}$ and $\mathbf{B} = \mathbf{US}$. They applied MM algorithm (Hunter and Lange 2004) to derive the multiplicative update rule, which is fast and guaranteed to converge to a fixed point.

Motivated by literature in spectral clustering and manifold learning, information like similarity between data can aid the non-negative matrix factorization. D. Cai et al. (2011) proposes a graph regularized non-negative matrix factorization (GNMF) that aims to incorporate the information in the similarity graph to the matrix factorization process. To achieve this goal, they penalize the loss function to, respectively,

$$\|\mathbf{X} - \mathbf{US}\|_2^2 + \lambda \mathbf{Tr}(\mathbf{U}^\top \mathbf{LU}),$$

where the similarity graph is denoted as $\mathbf{W}$, with $\mathbf{D}$ a diagonal matrix where $D_{jj} = \sum_l W_{jl}$, and $\mathbf{L} = \mathbf{D} - \mathbf{W}$, and

$$D(\mathbf{X}||\mathbf{US}) + \frac{\lambda}{2} \sum_{j=1}^{N} \sum_{l=1}^{N} \sum_{k=1}^{K} \left(U_{jk} \log \frac{U_{jk}}{U_{lk}} + U_{lk} \log \frac{U_{lk}}{U_{jk}} \right) \mathbf{W}_{jl},$$

where in both cases, $\lambda \geq 0$ is a penalization parameter. In D. Cai et al. (2011), it has shown that although the result does vary over λ, but it is uniformly better than plain NMF over a wide range. Similarly, by applying MM algorithm, one can obtain the multiplicative law of update. As for the choice of similarity graph, von Luxburg (2007) provides a thorough introduction of spectral clustering and the construction of commonly applied adjacency matrices. In this work, we tested K-nearest neighbor graphs and Gaussian kernel based K-nearest neighbor graph.

As for this specific scenario for flow fields, we should first note that the data matrix is not non-negative. However, due to the carefully chosen design points, by translation of the data can we achieve the non-negative property over the entire domain. Secondly, concerning the construction of neighborhood graphs, not only can the data itself be useful, but the location parameter can of great help. In the experiment section, we will test the effects of both approaches.

1.3 Kriging

With the spatial basis modes $\boldsymbol{\Phi}$ and corresponding mode coefficients $\boldsymbol{\beta}$ obtained through the methods above, we have produced a low-dimensional subspace, where the dataset in all training cases is projected. The coefficients associated with the modes can be interpreted as the coordinates in the subspace. In order to configure the coordinates of unknown design points, regression technique should be used to establish the mapping between the parametric input and ROM output. Previous study (Ni, Ding, and Wang 2023) illustrates that ROM with Kriging presents a superior performance in predicting almost all physical fields. Therefore, we use Kriging here to achieve the best possible result and ensure the fair comparison between matrix decomposition methods.

Kriging, also known as Gaussian process (GP) regression, is a statistical interpolation technique used for spatial modeling and prediction (Cressie 1990). The basic principle of kriging involves modeling the spatial variability between sample data points using a covariance function, fitting the covariance parameters using the sample data, and then using the covariance to calculate the weights of the sample data to estimate the value at the target point. By carefully choosing the model parameters, kriging can provide the best linear unbiased estimator for responses at unknown design points. The predictor of the coefficient vector at a new design point can be given as,

$$\widehat{\boldsymbol{\beta}}^{(new)} = \boldsymbol{\mu} + \boldsymbol{c}^{\boldsymbol{T}} \boldsymbol{C}^{-1} \otimes \boldsymbol{I_r} \, (\boldsymbol{\beta} - \mathbf{1_n} \otimes \boldsymbol{\mu}),$$

where $\boldsymbol{\beta}$ consists of the coefficient vectors of the training samples, $\boldsymbol{c}$ is an n-row vector identifying the correlation between the new design point and the training sampling points, and $\boldsymbol{C}$ is the correlation matrix of the training points. It is thus natural to recover the flow field at unknown design points,

$$\widehat{\boldsymbol{y}}^{(new)} = \boldsymbol{\Phi} \widehat{\boldsymbol{\beta}}^{(new)}$$

2 Experiment Result

2.1 Data Description

To examine the performance of each matrix decomposition method, we conduct a representative case: flow around a circular cylinder. Two design variables $\boldsymbol{h} = (\mu, V)$ are considered here: fluid kinematic viscosity in the range of 0.01 - 1 Pa·s and freestream streamwise velocity in the range of 0.5 - 2 m/s, which results in Reynolds number between 0.5 and 200. Velocity-x field is selected as the output. For simplicity, we focus on steady state flow field. The sample size of design points is set as $n = 30$, and 100 test points are randomly chosen in the design space. To allow for fast validation, total number of grids is set as $N = 3988$. Figure 2 is an example of velocity-x field.

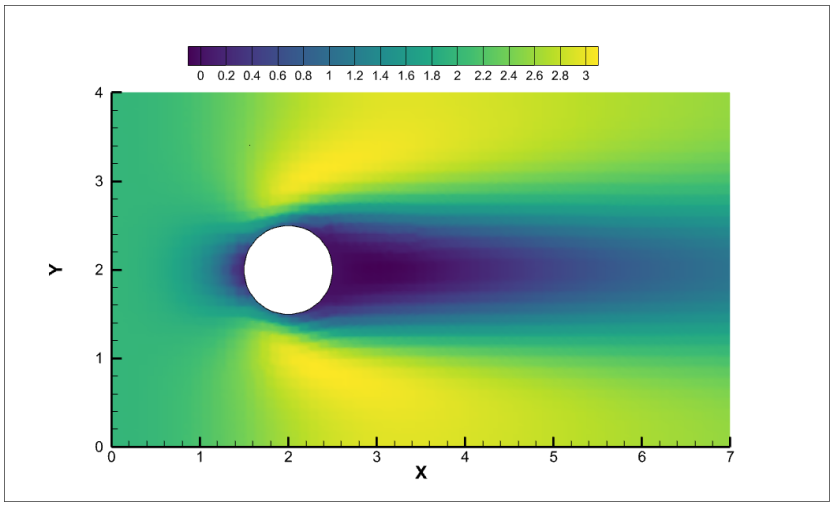


Figure 2: Velocity-x field ground truth in test case 39

To quantify the uncertainty of prediction, two performance metrics are introduced: relative root mean square error (rRMSE) and mean absolute error (MAE). The former is defined as $rRMSE = \sqrt{\sum_{l=1}^{N} (u_l - \widehat{u}_l)^2 / \sum_{l=1}^{N} (\widehat{u}_l)^2}$, where N is the total number of grids. The metrics are averaged over 100 test cases.

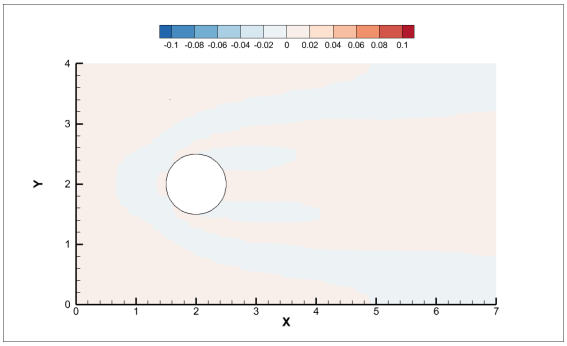
2.2 Performance Comparison

Table 1: rRMSE and MAE of velocity-x field predicted by different methods

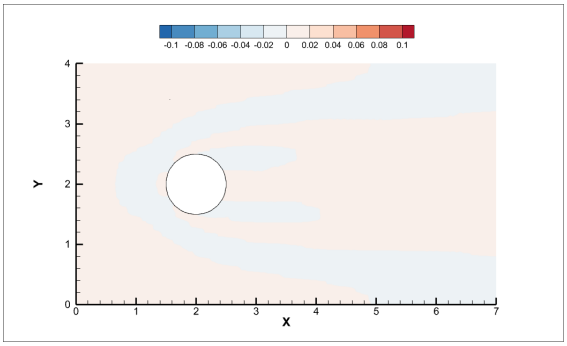
Evaluation metrics	PCA	Hetero PCA	kPCA	NMF	GNMF	self GNMF
rRMSE (%)	1.2077	1.2077	1.3316	1.6620	1.6448	1.6827
MAE (10^{-3} m/s)	9.2	9.2	10.3	15.5	15.4	15.9

The number of modes in the ROM is uniformly set as $K = 3$, where PCA is able to capture 99.96% of the overall energy. Due to the sensitivity of kriging model, parameters are carefully chosen using grid search, which is included as a Matlab function in the repository. In general, PCA outperforms other alternative methods in this case. Hetero PCA seems to converge to exactly the same basis in PCA. kPCA is second to PCA, but it should be noted that K is set as 50 here. This is reasonable since kPCA performs dimensionality reduction with the $N \times N$ matrix. NMF and its variants achieve decent results, but show no potential to improve the accuracy.

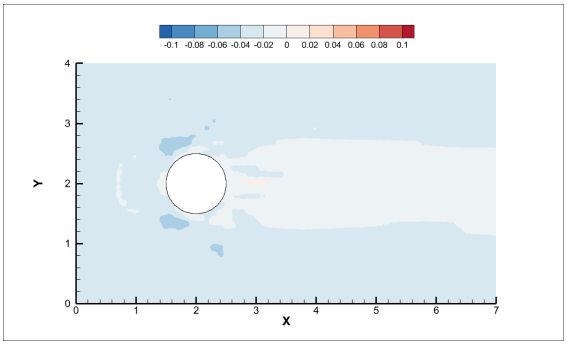
Figure 3 shows the error of different methods in test case 39. The figure is made on the same scale for easy comparison. It is clear that PCA and Hetero PCA achieve the same result with the minimal error. kPCA has a global shift, which is explained in **section 3.1**. NMF and its variants have much larger error around the cylinder and in the wake region.



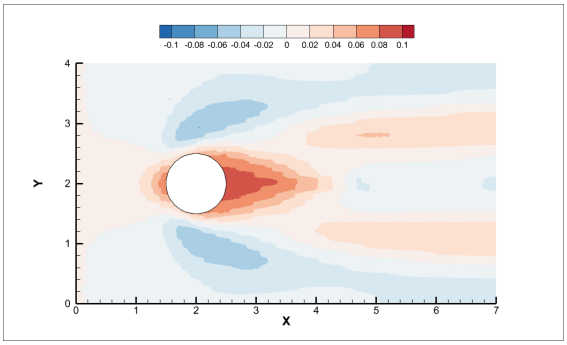
(a) PCA



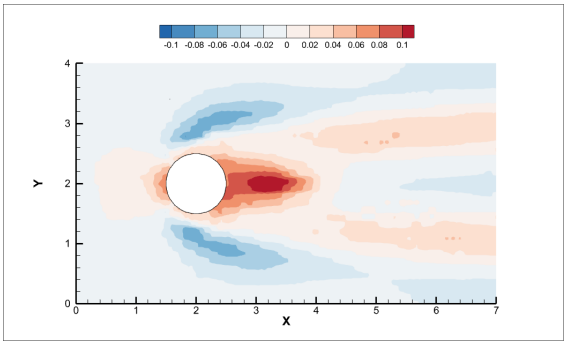
(b) Hetero PCA



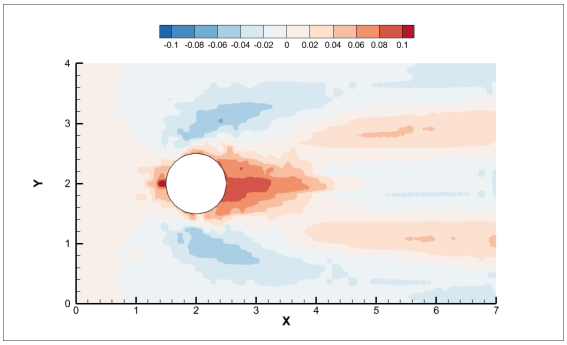
(c) kPCA



(d) NMF



(e) GNMF



(f) self GNMF

Figure 3: Error of different methods in test case 39

3 Conclusion

Compared with PCA, all the matrix decomposition methods are not as good. We will discuss the results respectively as follows.

3.1 Kernel Principal Component Analysis

Initially, we suspect that the problem of kPCA lies in the distribution of eigenvalues. Figure 4 illustrates how the eigenvalues distribute in the kernel matrix. Apparently, to achieve high proportion of total variance, the number of eigenvalues would be large, which will hamper the performance of machine learning prediction algorithm.

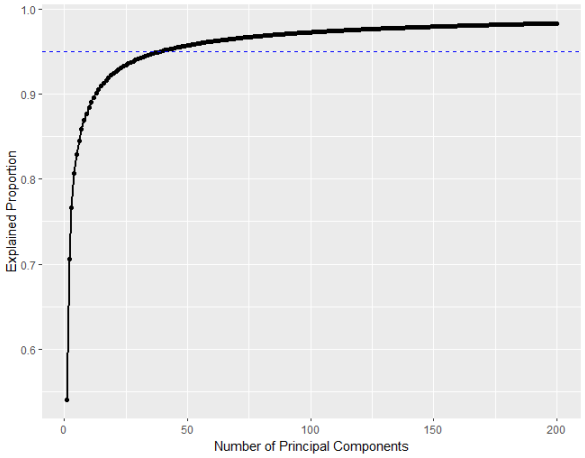


Figure 4: Explained proportion of the first 200 principal components when $c = 100$.

However, by careful inspection, we find that the proportion of total variance is not very important. This is reasonable, for PCA is the best decomposition in explaining the most variance. By tuning the kept dimensionality K from 5 to 100, the RMSE is similar. The actual problem lies in the shift. In PCA, the data matrix is centered and how much it centered is easily measurable. However, because the projection function itself in kPCA is often unavailable, let alone the inverse, the amount of centering effect in kPCA cannot be known. The information of the distance of the data from the original point diminishes when taking pseudo inverse, which is equivalent to being projected to the subspace spanned by U. Hence, the recovered flow field is shifted by the distance, and thus gives bad performance. We try applying another regression method for the shift. After careful comparison, we choose Gaussian Process Regression for the best accuracy. However, the result is still worse than PCA, which may result from the fusion of two sources of uncertainty from two regression models. Unfortunately, further investigation shows that kPCA is worse than PCA even if we restore the shift with the average field of the ground truth. This means kPCA just fails in this case regardless of the global shift.

3.2 Graph Regularized Non-negative Matrix Factorization

The plain NMF performs a bit worse than PCA can be explained as follows. First, the data itself is not guaranteed to be non-negative, and physically, it may not enjoy the "learn by parts" property. Furthermore, SVD is the optimal low-rank estimation in the sense of Frobenius norm. Even though changing the metric to other matrix norms or divergences may result in different result, the consistency is mainly retained. As for NMF, although it is designed to obtain the optimal decomposition in L_2 norm or divergence, the constraint of non-negativity (the solution space is limited) and the non-convex optimization problem (often trapped in local optimal) hinder its performance.

As for the GNMF, there may be different reasons for this case. First, it may be due to the limited aid of location information. Hence, even though we incorporate this source of information, we cannot achieve a significantly better performance. Secondly, it may still suffer from local optimal. The decomposition is not optimal under the loss function and the potential of location is not fulfilled.

In conclusion, though the theory of GNMF provides a way to incorporate more information in matrix decomposition, due to optimization and the characteristics of flow data, it does not show promising results. Additionally, the multiplicative update is quite computationally challenging compared with PCA, with slow convergence and local optimal issues. Therefore, NMF may not be useful under this scenario. However, due to computation limits, we haven't been able to tune the parameters thoroughly. By adjusting a good parameter, it may still be able to outperform PCA.

To sum up, we still find it confident that incorporating multi sources of data would help construct a more effective basis. However, due to the limited scope of our knowledge in covariate-assisted matrix decomposition and non-linear factor analysis methods, we still could not provide a better dimensionality reduction method than PCA. The limitation lies in that this task is actually different from ordinary task in statistics and machine learning. In traditional statistical analysis, we wish to find out the expectation through noise-polluted real world data. Yet in this task, there is no noise in the data, and domain-specific analysis is needed for this interesting specific task.

4 Code Availability

All codes attached are executable in Matlab or R. It is also available on Github.

References

[1] Deng Cai et al. "Graph Regularized Nonnegative Matrix Factorization for Data Representation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.8 (Aug. 2011), pp. 1548–1560.

[2] Noel Cressie. "The origins of kriging". In: *Mathematical Geology* 22 (1990), pp. 239–252.

[3] Dennis Decoste and Bernhard Schölkopf. "Training Invariant Support Vector Machines". In: *Machine Learning* 46.1 (Jan. 1, 2002), pp. 161–190.

[4] Trevor Hastie et al. *The elements of statistical learning: data mining, inference, and prediction*. Vol. 2. Springer, 2009.

[5] David R Hunter and Kenneth Lange. "A tutorial on MM algorithms". In: *The American Statistician* (2004), pp. 30–37.

[6] Daniel Lee and H. Sebastian Seung. "Algorithms for Non-Negative Matrix Factorization". In: *Advances in Neural Information Processing Systems*. Vol. 13. MIT Press, 2000.

[7] Daniel D. Lee and H. Sebastian Seung. "Learning the Parts of Objects by Non-Negative Matrix Factorization". In: *Nature* 401.6755 (6755 Oct. 1999), pp. 788–791.

[8] Zhen Miao et al. "Multi-Omics Integration in the Age of Million Single-Cell Data". In: *Nature Reviews Nephrology* 17.11 (11 Nov. 2021), pp. 710–724.

[9] Chenxu Ni, Siyu Ding, and Xingjian Wang. *Parametric reduced order models with machine learning for spatial emulation of mixing and combustion problems*. 2023. arXiv: 2308.14566 [physics.flu-dyn].

[10] Bernhard Schölkopf, Patrice Simard, et al. "Prior Knowledge in Support Vector Kernels". In: *Advances in Neural Information Processing Systems*. Vol. 10. MIT Press, 1997.

[11] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. "Kernel Principal Component Analysis". In: *Artificial Neural Networks — ICANN'97*. Ed. by Wulfram Gerstner et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1997, pp. 583–588. ISBN: 978-3-540-69620-9.

[12] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[13] Ulrike von Luxburg. "A Tutorial on Spectral Clustering". In: *Statistics and Computing* 17.4 (Dec. 1, 2007), pp. 395–416.

[14] Yang Xu and Rachel Patton McCord. "Diagonal Integration of Multimodal Single-Cell Data: Potential Pitfalls and Paths Forward". In: *Nature Communications* 13.1 (1 June 18, 2022), p. 3505.

[15] Anru R. Zhang, T. Tony Cai, and Yihong Wu. *Heteroskedastic PCA: Algorithm, Optimality, and Applications*. Apr. 1, 2021. arXiv: 1810.08316. preprint.