

**UCLA**

**UCLA Electronic Theses and Dissertations**

**Title**

Data-Driven Dimensionality Reduction for Aerodynamic Applications

**Permalink**

<https://escholarship.org/uc/item/8d552374>

**ISBN**

9798186371410

**Author**

Tran, Jonathan

**Publication Date**

2026-08-17

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Data-Driven Dimensionality Reduction for Aerodynamic Applications

A dissertation submitted in partial satisfaction  
of the requirements for the degree of Doctor of Philosophy  
in Aerospace Engineering

by

Jonathan Tran

2026



© Copyright by

Jonathan Tran

2026

# ABSTRACT OF THE DISSERTATION

Data-Driven Dimensionality Reduction for Aerodynamic Applications

by

Jonathan Tran

Doctor of Philosophy in Aerospace Engineering

University of California, Los Angeles, 2026

Professor Kunihiko Taira, Co-Chair

Professor Jeff D. Eldredge, Co-Chair

Fluid flows are high-dimensional, nonlinear dynamical systems governed by partial differential equations, making their direct simulation computationally expensive for many practical applications. Nevertheless, many flows evolve on low-dimensional manifolds, motivating reduced-order models that capture the essential dynamics while significantly reducing computational complexity. Recent advances in machine learning provide powerful nonlinear dimensionality reduction techniques capable of learning such representations directly from data.

This thesis investigates data-driven latent representations for reduced-order modeling in fluid mechanics, with an emphasis on aerodynamic applications. Rather than treating latent spaces solely as compact embeddings optimized for reconstruction, this work explores representations that incorporate physical and mathematical structure to improve interpretability, robustness, and predictive capability.

First, an autoencoder-based framework is developed for aerodynamic design optimization of industrial automobile geometries. The learned latent representation efficiently captures

geometric variations, enabling accelerated design exploration while achieving an 11% reduction in drag coefficient. The framework predicts aerodynamic performance within 2% of experimentally validated large-eddy simulations, demonstrating sufficient fidelity for practical design optimization.

Second, a physically structured latent space based on optimal transport distances is introduced for flow control analysis. Applied to separated flow over a NACA 0012 airfoil with leading-edge thermal actuation, the approach identifies distinct control regimes associated with changes in separation bubble size and the onset of partial and global laminarization, providing an interpretable representation of the underlying flow dynamics.

Finally, a probabilistic state estimation framework combining latent diffusion models with particle filtering is developed for reconstructing and predicting transient aerodynamic states from limited observations. Applied to a high-incidence airfoil subjected to gust disturbances, the framework enables uncertainty-aware estimation of complex unsteady flows.

Collectively, these studies demonstrate that physically informed latent representations provide efficient and interpretable reduced-order models for aerodynamic design optimization, flow control assessment, and probabilistic state estimation. The results highlight the importance of incorporating physical structure into representation learning, illustrating the potential of hybrid machine learning and physics-based approaches for reliable modeling and analysis of complex fluid systems.

The dissertation of Jonathan Tran is approved.

Shahriar Talebi

Ann R. Karagozian

Jeff D. Eldredge, Committee Co-Chair

Kunihiko Taira, Committee Co-Chair

University of California, Los Angeles

2026

*“The mathematician plays a game in which he himself invents the rules, while the physicist plays a game in which the rules are provided by nature. But as time goes on, it becomes increasingly evident that the rules which the mathematician finds interesting are the same as those which nature has chosen.”*

— Paul A. M. Dirac

# TABLE OF CONTENTS

|          |                                                                              |           |
|----------|------------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>Introduction . . . . .</b>                                                | <b>1</b>  |
| 1.1      | Motivation . . . . .                                                         | 1         |
| 1.2      | Notation . . . . .                                                           | 3         |
| 1.2.1    | Notation for Norms and Function Spaces . . . . .                             | 3         |
| 1.2.2    | Notation for Probability Measures and Distributions . . . . .                | 5         |
| 1.3      | Background on Data-Driven Dimensionality Reduction in Fluid Mechanics .      | 5         |
| 1.3.1    | The Manifold Hypothesis and Dimensionality Reduction . . . . .               | 6         |
| 1.3.2    | Proper Orthogonal Decomposition . . . . .                                    | 10        |
| 1.3.3    | Autoencoders . . . . .                                                       | 15        |
| <b>2</b> | <b>Data-Driven Dimensionality Reduction for Aerodynamic Design Optimiza-</b> |           |
|          | <b>tion of Automobiles . . . . .</b>                                         | <b>19</b> |
| 2.1      | Background and Motivation . . . . .                                          | 19        |
| 2.2      | Methods . . . . .                                                            | 24        |
| 2.2.1    | Automobile Dataset Preparation . . . . .                                     | 24        |
| 2.2.2    | PCA-Assisted Autoencoder setup . . . . .                                     | 25        |
| 2.2.3    | Shape Optimization Procedure . . . . .                                       | 27        |
| 2.3      | Results . . . . .                                                            | 29        |
| 2.3.1    | Automobile Aerodynamic Analysis . . . . .                                    | 29        |
| 2.3.2    | Latent Manifold Discovery . . . . .                                          | 31        |
| 2.3.3    | Data-Driven Shape Optimization . . . . .                                     | 34        |
| 2.4      | Conclusions . . . . .                                                        | 39        |

|          |                                                                                                                                      |           |
|----------|--------------------------------------------------------------------------------------------------------------------------------------|-----------|
| <b>3</b> | <b>Learning Minimum-Distortion Optimal Transport Latent Embeddings: Theory and Applications to Separated Flow Analysis . . . . .</b> | <b>41</b> |
| 3.1      | Background and Motivation . . . . .                                                                                                  | 41        |
| 3.2      | Optimal Transport for Comparison of Fluid Flows . . . . .                                                                            | 45        |
| 3.2.1    | Physical Intuition for Optimal Transport . . . . .                                                                                   | 46        |
| 3.2.2    | Balanced Optimal Transport Distances . . . . .                                                                                       | 47        |
| 3.2.3    | Definition of Csiszár Divergence Functionals . . . . .                                                                               | 49        |
| 3.2.4    | Unbalanced Optimal Transport Distances . . . . .                                                                                     | 50        |
| 3.2.5    | Numerical Computation of the Unbalanced Optimal Transport Based<br>Flow Field Dissimilarity . . . . .                                | 51        |
| 3.3      | Separated Flow Configuration and Control Setup . . . . .                                                                             | 55        |
| 3.4      | OT Based Autoencoder for Learning Minimum-Distortion Embeddings . . .                                                                | 58        |
| 3.5      | Results . . . . .                                                                                                                    | 63        |
| 3.5.1    | $\alpha = 9^\circ$ Latent Space . . . . .                                                                                            | 64        |
| 3.5.2    | $\alpha = 6^\circ$ Latent Space . . . . .                                                                                            | 71        |
| 3.6      | Conclusions . . . . .                                                                                                                | 76        |
| <b>4</b> | <b>Reduced-Order State Estimation of Fluid Flows via Diffusion Model Augmented Particle Filtering . . . . .</b>                      | <b>78</b> |
| 4.1      | Introduction . . . . .                                                                                                               | 78        |
| 4.2      | Probabilistic Autoencoder-Based Forward Model . . . . .                                                                              | 79        |
| 4.3      | Probabilistic Latent State Estimation from Sensor Measurements with Con-<br>ditional Score-Based Diffusion Models . . . . .          | 85        |
| 4.3.1    | Forward Diffusion Process . . . . .                                                                                                  | 86        |

|          |                                                                                 |            |
|----------|---------------------------------------------------------------------------------|------------|
| 4.3.2    | Variance-Exploding (VE) and Variance-Preserving (VP) Cases . . . . .            | 86         |
| 4.3.3    | Reverse-Time Generative Process . . . . .                                       | 88         |
| 4.4      | Ensemble Propagation and Assimilation of Measurements . . . . .                 | 91         |
| 4.4.1    | Neural ODE Prediction Step . . . . .                                            | 93         |
| 4.4.2    | Diffusion Augmentation . . . . .                                                | 96         |
| 4.4.3    | Importance Sampling . . . . .                                                   | 97         |
| 4.5      | Application to Laminar Gust-Airfoil Interaction . . . . .                       | 99         |
| 4.6      | Conclusions . . . . .                                                           | 104        |
| <b>5</b> | <b>Concluding Remarks and Future Directions . . . . .</b>                       | <b>110</b> |
| 5.1      | Summary of Contributions . . . . .                                              | 110        |
| 5.2      | Future Directions . . . . .                                                     | 113        |
| 5.2.1    | State Estimation for Highly Unsteady Aerodynamic Disturbances . .               | 113        |
| 5.2.2    | Latent-Space Flow Control . . . . .                                             | 114        |
| 5.2.3    | Geometric Machine Learning for Reduced-Order Modeling . . . . .                 | 115        |
| 5.2.4    | Algebraic Topology for Representation Learning of Fluids . . . . .              | 116        |
| 5.2.5    | Optimal Transport on Manifolds for Fluid Mechanics . . . . .                    | 118        |
| 5.2.6    | Toward Unified Physics-Informed Latent Representations . . . . .                | 122        |
| <b>A</b> | <b>Probability Theory Basics for Optimal Transport . . . . .</b>                | <b>124</b> |
| <b>B</b> | <b>OT Autoencoder Model Additional Details . . . . .</b>                        | <b>127</b> |
| <b>C</b> | <b>Positive Definiteness of the OT-Based Flow Field Dissimilarity . . . . .</b> | <b>132</b> |
| C.1      | Verification of the assumptions of Theorem C.0.1 . . . . .                      | 134        |



|                   |                                                               |            |
|-------------------|---------------------------------------------------------------|------------|
| C.1.1             | Symmetry and Lipschitz continuity . . . . .                   | 134        |
| C.1.2             | Properties of the associated kernel . . . . .                 | 136        |
| C.2               | Extension to signed and multi-component flow fields . . . . . | 141        |
| <b>References</b> | . . . . .                                                     | <b>146</b> |

## LIST OF FIGURES

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |    |
|-----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.1 | Overview of data-driven automobile design analysis. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | 21 |
| 2.2 | Principal component analysis (PCA) assisted observable augmented autoencoder. Voxelized vehicle geometries are taken as input and output and the drag coefficient is estimated from the encoded latent representation. . . . .                                                                                                                                                                                                                                                                                                                         | 25 |
| 2.3 | Depiction of the vehicle dataset consisting of sport utility vehicles (SUV), hatchbacks, sedans, and box geometries. (a) Representative voxelized vehicle geometries for different car types. (b) Example 3D and 2D (center plane) isocontours of normalized total pressure coefficient, $C_{p,t}$ , for different car types. $C_{p,t} = 0$ isocontour is shown indicating separated flow. (c) Probability density plot, $p(C_D)$ , of normalized drag coefficient, $C_D$ , with respect to vehicle type. . . . .                                      | 29 |
| 2.4 | Nonlinear machine-learning-based compression of vehicle geometries. Left plot depicts latent space of sport utility vehicles (SUV), hatchbacks, sedans, and box geometries. Right parity plots show comparison of estimated and reference normalized drag coefficients $C_D$ . Comparisons of example vehicle geometries and autoencoder reconstructions are shown. The value underneath each decoded geometry reports the percent $L_2$ reconstruction error metric, $\varepsilon = \ \mathbf{p} - \hat{\mathbf{p}}\ _2 / \ \mathbf{p}\ _2$ . . . . . | 33 |
| 2.5 | Data-driven vehicle shape optimization in the discovered latent space colored by the normalized drag coefficient. The latent optimization path with the corresponding normalized drag coefficient are shown. Orange points depict drag values obtained from validation computational fluid dynamics (CFD) simulation. (a) The given initial car and (b) optimized geometry are also depicted. The regions highlighted in yellow highlight some features that have been substantially modified. . . . .                                                 | 35 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                    |    |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2.6 | Validation large eddy simulation (LES) flow fields for smoothed voxel geometry for the initial high drag SUV case and the modified geometry obtained from the optimization procedure. (left) Zero total pressure ( $C_{p,t} = 0$ ) isocontour colored by pressure coefficient, $C_p$ . (right) Midplane slice of pressure coefficient field, $C_p$ . The pressure difference around the rear spoiler region is reduced. . . . .    | 37 |
| 2.7 | Instantaneous wake flow behind vehicle geometry simulated with LES visualized with $Q$ -criterion ( $Q = 16 \times 10^4 \text{ s}^{-2}$ ) isosurface colored by streamwise velocity value. . . . .                                                                                                                                                                                                                                 | 38 |
| 3.1 | Pedagogical schematic of optimal transport between a supply and demand distribution representative of two different flow fields. The optimal transport distance is the minimum total cost to move the supply to the demand distribution. . . .                                                                                                                                                                                     | 46 |
| 3.2 | Comparison of unbalanced optimal transport distance $d_{\text{UOT}}$ with the $L^2$ metric. (a) Gaussian pulse undergoing advection and diffusion. (b) Comparison of distances between the evolving pulse and the initial condition. . . . .                                                                                                                                                                                       | 48 |
| 3.3 | Examples of baseline flow and various responses of a separated wake past a NACA 0012 airfoil at $\alpha = 6^\circ$ to a heat flux actuator input at the leading edge [YT19]. The $QL_c^2/u_\infty^2 = 50$ isosurface colored by the normalized streamwise velocity is shown. . . . .                                                                                                                                               | 57 |
| 3.4 | Illustration of the augmented autoencoder problem setup. During training, the model learns to associate Euclidean distances in the latent space (red line) with flow field dissimilarities computed with optimal transport. . . . .                                                                                                                                                                                                | 59 |
| 3.5 | Example parameter study of the OT-based autoencoder for $\alpha = 9^\circ$ . (a) L-curve showing the trade-off between $\mathcal{L}_1$ and $\mathcal{L}_2$ for the test set with $10^{-4} \leq \lambda \leq 10^4$ . (b) Variation of the total loss $\mathcal{L}_1 + \lambda\mathcal{L}_2$ with respect to the latent space dimension for $\lambda = 0.1$ . Standard deviation for the last 500 epochs is colored in gray. . . . . | 62 |

|      |                                                                                                                                                                                                                                                                                                                                                                                                                                 |    |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 3.6  | Reconstructions of baseline flow fields by OT autoencoder. The $\bar{u}_x = 0$ isocontour is shown in black for all fields. The reconstructed vorticity is obtained by central differencing of the reconstructed velocity fields. The percent Frobenius norm reconstruction error is reported. . . . .                                                                                                                          | 65 |
| 3.7  | Comparison of learned latent spaces with $\alpha = 9^\circ$ colored by aerodynamic performance $\bar{C}_L/\bar{C}_D$ for (a) standard autoencoder ( $\mathcal{L}_1$ loss only) (b) OT autoencoder ( $\mathcal{L}_1$ and $\mathcal{L}_2$ losses). (c) Isocontours of time-averaged streamwise velocity $\bar{u}_x = 0$ shown for different representative cases labeled in (a) and (b). . . . .                                  | 67 |
| 3.8  | Plot of latent embeddings highlighting the influence of actuation parameters for $\alpha = 9^\circ$ . Labeled are example actuation cases. Examples of average turbulent kinetic energy, average vorticity fields, as well as instantaneous $Q$ -criterion ( $QL_c^2/u_\infty^2 = 50$ ) colored by streamwise velocity are shown for the labeled cases. The $\bar{u}_x = 0$ isocontour is shown for all average fields. . . . . | 69 |
| 3.9  | Comparison of learned latent spaces with $\alpha = 6^\circ$ colored by control performance $\bar{C}_L/\bar{C}_D$ for (a) standard autoencoder ( $\mathcal{L}_1$ loss only) (b) OT autoencoder ( $\mathcal{L}_1$ and $\mathcal{L}_2$ losses). (c) Isocontours of time-averaged streamwise velocity $\bar{u}_x = 0$ shown for different representative cases labeled in (a) and (b). . . . .                                      | 72 |
| 3.10 | Plot of latent embeddings highlighting the influence of actuation parameters for $\alpha = 6^\circ$ . Labeled are example actuation cases. Examples of average turbulent kinetic energy, average vorticity fields, as well as instantaneous $Q$ -criterion ( $QL_c^2/u_\infty^2 = 50$ ) colored by streamwise velocity are shown for the labeled cases. The $\bar{u}_x = 0$ isocontour is shown for all average fields. . . . . | 74 |
| 4.1  | Autoencoder with probabilistic observable decoder. . . . .                                                                                                                                                                                                                                                                                                                                                                      | 80 |
| 4.2  | Illustration of the latent diffusion process. . . . .                                                                                                                                                                                                                                                                                                                                                                           | 84 |

|     |                                                                                                                                                                                                                                                                                                                                                                                                                             |     |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 4.3 | Illustration of the latent particle filter procedure. Red particles indicate samples from the previous measurement ensemble. Blue particles indicate samples from the diffusion model conditioned on the current measurement. . . . .                                                                                                                                                                                       | 92  |
| 4.4 | (a) Example gust-airfoil interaction. Pressure sensors marked in red. (b) Probabilistic Obs.-AE latent space. Baseline trajectory highlighted in black. (b) Example maximum likelihood estimation of leading edge sensor pressure coefficient from latent space. . . . .                                                                                                                                                    | 100 |
| 4.5 | State estimation for the first latent coordinate $z_1$ from noisy pressure sensor information for an example gust encounter. Blue lines show the true latent state trajectory and the gray lines show ensemble particles. The time $t = 0$ corresponds to when the gust has convected a distance of $ x_0 /c$ . The gust parameters are $(x_0, y_0, R/c, G) = (-3, 0, 0.25, -2.5)$ . . . . .                                | 101 |
| 4.6 | Statistics of estimated vorticity field ensemble for the diffusion-augmented particle filter compared to the original flow field. . . . .                                                                                                                                                                                                                                                                                   | 103 |
| 4.7 | Comparison of reconstruction of different methods. Shown are ensemble statistics of the estimated vorticity field predicted by (a) the neural ODE, (b) the diffusion model, (c) the particle filter, and (d) the diffusion model-augmented particle filter, compared against the reference flow field at convective time $tU_\infty/c = -1.2$ . The gust parameters are $(x_0, y_0, R/c, G) = (-3, 0, 0.25, 2.5)$ . . . . . | 105 |
| B.1 | Plot of loss curves for (a) $\alpha = 9^\circ$ and (b) $\alpha = 6^\circ$ cases of both regular (top) and OT-based autoencoders (bottom). . . . .                                                                                                                                                                                                                                                                           | 128 |
| B.2 | Plot of $\alpha = 9^\circ$ latent space for both regular and OT-based autoencoder with (a) $n = 3$ and (b) $n = 8$ (showing the first 3 principal axes). . . . .                                                                                                                                                                                                                                                            | 129 |
| B.3 | Plot of $\alpha = 6^\circ$ latent space for both regular and OT-based autoencoder with (a) $n = 3$ and (b) $n = 8$ (showing the first 3 principal axes). . . . .                                                                                                                                                                                                                                                            | 130 |

B.4 Plot of pairwise UOT-based distances ( $\tilde{d}_{\text{field}}$ ) vs. pairwise  $L^2$  distances for discretized flow fields for (a)  $\alpha = 9^\circ$  and (b)  $\alpha = 6^\circ$ . Both axes have been normalized by the maximum pairwise distance. Red points denote example pairwise distances from the baseline flow corresponding to the cases in Figure 3.8 and Figure 3.10. 131

## LIST OF TABLES

|     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |     |
|-----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 3.1 | Network architecture of the encoder and decoder models. The activation function used is ReLU. . . . .                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 61  |
| 3.2 | Comparison of average percent error for field variables using different autoencoder architectures (with two latent variables) at angles of attack $\alpha = 6^\circ$ and $9^\circ$ . Field variable errors are reported as percent Frobenius norm error. The embedding loss is reported using the MDS stress. . . . .                                                                                                                                                                                                                                                 | 66  |
| 4.1 | Network architecture of the probabilistic autoencoder. The encoder predicts the latent Gaussian parameters $(\mu_z, \log \sigma_z^2)$ , from which the latent vector is sampled using the reparameterization trick. The field decoder reconstructs the input field, while the probabilistic decoder predicts the mean and log-variance of the sensor measurements. GN denotes Group Normalization [WH18] and BN denotes Batch Normalization [IS15]. All hidden layers use the SiLU activation function, $\text{SiLU}(x) = x \sigma(x) = \frac{x}{1+e^{-x}}$ . . . . . | 107 |
| 4.2 | Network architecture of the score-based diffusion model (SDM). Independent embedding networks first encode the diffusion time and sensor observations. These embeddings are concatenated with the perturbed latent state before being processed by a residual multilayer perceptron consisting of four residual blocks. The final output is the estimated score of the perturbed latent state. All hidden layers employ the SiLU activation function. . . . .                                                                                                         | 108 |
| 4.3 | Network architecture of the latent Neural ODE. A linear shortcut connection from the input layer is added to the network output to improve optimization and enable efficient learning of near-linear latent dynamics. . . . .                                                                                                                                                                                                                                                                                                                                         | 109 |

## ACKNOWLEDGMENTS

Since joining UCLA as an undergraduate in 2019, I have grown both personally and professionally through the support of my family, friends, mentors, colleagues, and collaborators. The diverse perspectives of faculty and students across many disciplines taught me to value interdisciplinary research, an outlook that continues to shape how I approach scientific inquiry.

I am deeply grateful to my friends who have remained by my side over the years. In particular, I thank Kino Gulbe, Johnny Pham, Byron James, Artyom Lisitsyn, Marcus Lee, Mitchel Graver, Emily Lee, Erin Lee, Jamie Chang, Rohan Rao, and Andrew Huynh for the countless conversations we have shared about life. I also thank my high school robotics mentors, John Wright, Alejandro Peralta, and Murat Veysoglu, whose guidance first inspired me to pursue engineering. I am especially grateful to Alejandro for helping me obtain my first internship at JPL.

I have been fortunate to work alongside an exceptional group of lab members and visitors, whose friendship made graduate school both enjoyable and memorable. I especially thank Kai Fukami for showing me around Tokyo and introducing me to the art of making figures; Veda Sri Godavarthi for guiding me through my first project on cluster-based control; Jean Helder Marques Ribeiro, whose legendary lack of spice tolerance I will never forget; Youngjae Kim for our countless walks around campus and constant support; and Yonghong Zhong for the occasional snacks. I also thank Alec Linot for always being someone I could talk to, and Victoria Rolandi for her kindness, generosity, and friendship. I am grateful as well to Gabrielle C. Claus, Chris McCormick, Hiroto Odaka, Chase Klewicki, Luke Smith, Vishal Anantharaman, Barbara Lopez-Doriga, Cheryl Lee, Hiroshi Omichi, Yueshan Yang, and Adam Schroeder, along with the members of the extended unsteady aerodynamics supergroup, especially Qiyuan (Bill) Wu, Hanieh Mousavi, and Zhecheng Liu.

I thank my advisor, Prof. Kunihiro (Sam) Taira, for his mentorship and for the many



opportunities he has given me since we first met during the data science course I audited back in Spring 2020. I am equally grateful to my co-advisor, Prof. Jeff D. Eldredge, for his guidance and insightful discussions throughout my graduate studies. I also thank my committee members, Prof. Ann R. Karagozian and Prof. Shahriar Talebi, for their thoughtful feedback, as well as the many UCLA faculty and staff who supported me along the way, including Prof. Y. Sungtaek Ju, Prof. Mason Porter, Prof. Pirouz Kavehpour, Prof. Eric Hester, and Zion Smith.

I also thank my collaborators Kenta Inada, Daisuke Umehara, Yoshimichi Ono, and Kenta Ogawa from Honda, together with Prof. Chi-An Yeh from North Carolina State University. In particular, we thank Kenta Inada for providing the simulation data for the automobiles and Professor Yeh for providing a majority of the simulation data for the separated-airfoil cases. This work was supported by Honda Motor Co., Ltd., the U.S. Air Force Office of Scientific Research (FA9550-23-1-0715), the U.S. Department of Defense Vannevar Bush Faculty Fellowship (N00014-22-1-2798), and the U.S. Army Research Office (W911NF-26-1-A065).

Finally, I thank my family for their unwavering love and encouragement. I am grateful to my brother, Kevin Tran, and my cousins, Shawn Nguyen and Eric Nguyen, for their support. I dedicate this thesis to my mother, Teresa Tran, and to the memory of my late father, Anthony Tran.

## VITA

- 2023            B.S. Aerospace Engineering, University of California, Los Angeles.
- 2023            B.S. Applied Mathematics, Specialization in Computing, University of California, Los Angeles.
- 2025            M.S. Mechanical Engineering, University of California, Los Angeles.

## PUBLICATIONS AND PRESENTATIONS

J. **Tran**, K. Fukami, K. Inada, D. Umehara, Y. Ono, K. Ogawa, and K. Taira. *Aerodynamics-Guided Machine Learning for Design Optimization of Electric Vehicles. Communications Engineering*, 2024. DOI: 10.1038/s44172-024-00322-0.

J. **Tran**, C.-A. Yeh, and K. Taira. *Using Optimal Transport Aligned Latent Embeddings for Separated Flow Analysis. Journal of Fluid Mechanics*, 2026. DOI: 10.1017/jfm.2025.11077.

J. **Tran**, J. D. Eldredge, and K. Taira., *Reduced-Order State Estimation of Fluid Flows via Diffusion Models and Particle Filtering*. Presented at the 19th Southern California Flow Physics Symposium (SoCal Fluids XIX), Riverside, California, April 2026.

# CHAPTER 1

## Introduction

### 1.1 Motivation

The modeling of physical systems such as fluid flows is of fundamental importance across engineering, physics, and mathematics. In engineering contexts, these models may be employed to understand how a fluid system may respond to some control input or change of parameters. In physics, they can provide a framework for investigating and interpreting the underlying mechanisms governing interesting flow phenomena. From a mathematical perspective, attention is frequently directed toward characterizing the regimes in which a given model loses validity, including limiting cases where its assumptions break down or its predictive capability deteriorates. Modeling is especially important for aerodynamics applications, in which relevant tasks include, but are not limited to:

- Identification and analysis of “important” flow field structures and their effects on aircraft performance
- Estimation and prediction of the flow field state from onboard avionics
- Inverse design, such as optimization of a performance metric like aerodynamic efficiency
- Design and assessment of control strategies for flight stability or disturbance rejection

Fluid flows fundamentally are described by an infinite-dimensional dynamical system whose evolution is governed by a set nonlinear partial differential equations (whether it be the

Navier-Stokes equations for viscous flows or Euler equations for inviscid flows), which often renders analytical treatment difficult except in highly simplified cases. Consequently, their study typically necessitates both qualitative and quantitative analysis from high-fidelity numerical simulations, experimental measurements, or hybrid approaches that integrate both.

In many cases, the relevant parameters or state variables are extremely high-dimensional and there is a need for the systematic exploration of high-dimensional parameter spaces, posing significant challenges for tractability, particularly in real-time or iterative settings. These difficulties are further exacerbated by the nonlinear nature of the governing dynamics and the presence of multiple interacting spatiotemporal scales. Consequently, directly studying the full-order models derived from first principles typically requires significant domain expertise in addition to being computationally prohibitive. With this in mind, it becomes advantageous to employ reduced-order models that capture the essential features of the system while representing it in a substantially lower-dimensional space.

The increasing availability of high-fidelity data from both computational simulations and experimental studies has led to a growing interest in data-driven reduced-order modeling. In particular, machine learning-based approaches have seen significant advancement, driven by both algorithmic developments and improvements in computational hardware. These methods offer a flexible framework for extracting low-dimensional structure from complex, high-dimensional systems and have proven effective across a range of applications in fluid dynamics.

In this thesis, we investigate the use of machine learning-based dimensionality reduction techniques for several downstream tasks. Specifically, we consider three primary applications:

1. An autoencoder-based approach for aerodynamic design optimization of industrial automobile geometries,
2. The use of optimal transport-based distances to learn an “interpretable” latent geometry, applied to the assessment of control performance in separated flows over an airfoil

with thermal actuation,

3. Probabilistic state estimation and data assimilation of gust encounters using a latent diffusion model coupled with a particle filtering framework.

## 1.2 Notation

In this thesis, we consider problems arising in fluid mechanics, with a particular emphasis on mathematically founded and data-driven approaches to the modeling, analysis, and reduced-order representation of fluid flows. In this context, flow fields are represented as functions of space (and possibly time) and both deterministic and probabilistic descriptions may be employed. In particular, depending on the setting, quantities of interest may be treated either as classical fields, probability measures, or empirical data distributions arising from numerical simulations or data-driven models. To avoid ambiguity across these perspectives, we adopt the following notational conventions commonly used in engineering mathematics and physics.<sup>1</sup>

### 1.2.1 Notation for Norms and Function Spaces

We use  $\ell^p$  (or  $\ell_p$ ) for finite-dimensional vector spaces equipped with the usual  $p$ -norm, denoted  $\|\cdot\|_{\ell^p}$ . For functions, we write  $L^p(\Omega)$  (or  $L_p(\Omega)$ ) for the Lebesgue spaces on a Lebesgue measurable domain  $\Omega \subseteq \mathbb{R}^d$ , equipped with the norm induced by the functional  $\|\cdot\|_{L^p(\Omega)}$ . For  $1 \leq p < \infty$ , these are given by

$$\|x\|_{\ell^p} = \left( \sum_{i=1}^n |x_i|^p \right)^{1/p}, \quad \|f\|_{L^p(\Omega)} = \left( \int_{\Omega} |f(x)|^p dx \right)^{1/p},$$

---

<sup>1</sup>Due to overlapping notation across fields, it may sometimes be necessary to redefine notation within individual sections for further clarity.

where the integral is taken with respect to Lebesgue measure unless otherwise stated. For  $p = \infty$ , we define

$$\|x\|_{\ell^\infty} = \max_{1 \leq i \leq n} |x_i|, \quad \|f\|_{L^\infty(\Omega)} = \operatorname{ess\,sup}_{x \in \Omega} |f(x)|.$$

When the underlying space is clear from context, we simply write  $\|\cdot\|_p$ .

Strictly speaking,  $L^p(\Omega)$  is not a space of functions but rather a space of equivalence classes of measurable functions. Indeed, let

$$f \sim g \iff f = g \text{ almost everywhere on } \Omega,$$

where a property is said to hold *almost everywhere* (a.e.) if the set on which it fails has Lebesgue measure zero. The functional  $\|\cdot\|_{L^p(\Omega)}$  is only a seminorm on the space of measurable functions with finite  $p$ -norm, since it vanishes precisely on functions that are zero almost everywhere. Consequently,

$$L^p(\Omega) = \{ [f] : f \text{ is Lebesgue measurable and } \|f\|_{L^p(\Omega)} < \infty \},$$

where

$$[f] = \{ g : \Omega \rightarrow \mathbb{R} : g \text{ is Lebesgue measurable and } g \sim f \}.$$

The seminorm therefore induces a genuine norm on the quotient space, and throughout this thesis we regard  $L^p(\Omega)$  as this quotient normed space, which is complete.<sup>2</sup>

In addition to function and vector spaces, when convenient, we will also work with abstracted ambient spaces  $\mathcal{X}$ , which will be specified by context. In particular,  $\mathcal{X}$  may denote either a measurable space or a Hilbert space, depending on the setting. When  $\mathcal{X}$  is a Hilbert

---

<sup>2</sup>By a slight abuse of notation, when writing  $f \in L^p(\Omega)$  we often write  $f$  for an arbitrary representative of its equivalence class. Unless explicitly stated otherwise, all subsequent constructions are well defined on equivalence classes and are understood to be independent of the choice of representative.

space, we denote its inner product by  $\langle \cdot, \cdot \rangle_{\mathcal{X}}$  and its induced norm by

$$\|x\|_{\mathcal{X}} = \sqrt{\langle x, x \rangle_{\mathcal{X}}}.$$

### 1.2.2 Notation for Probability Measures and Distributions

Since the present discussion involves both discrete and continuous probability laws, when possible we distinguish between probability measures, denoted by  $P$  or  $\pi$ , and probability density functions, denoted by  $p$ , when the probability measure admits a density with respect to the Lebesgue measure. When this distinction is not relevant, we may use the term “probability distribution” to refer to the underlying probability measure, identifying it with its density function whenever the latter exists.<sup>3</sup>

Likewise, Dirac masses are regarded as probability measures, denoted by  $\delta_{x_0}(\cdot)$ , rather than as functions. When convenient, we will abuse notation by writing  $\delta(x - x_0)$  or  $\delta(x - x_0)dx$  to denote the corresponding Dirac measure at  $x_0$ , identifying it with the associated Dirac delta distribution. For a review of the relevant concepts from probabilistic modeling, we refer the reader to any introductory text, such as [Fel91, Wil91, Tao11, Bre12, Eva12].

## 1.3 Background on Data-Driven Dimensionality Reduction in Fluid Mechanics

Before addressing the specific applications considered in this work, it is necessary to situate the present study within the broader literature on data-driven dimensionality reduction. This section provides an overview of relevant methodologies, with particular emphasis on the

---

<sup>3</sup>Recall that a *distribution* (in the sense of Schwartz) is a continuous linear functional on the space of smooth test functions with compact support, denoted  $C_c^\infty(\Omega)$  (or  $\mathcal{D}(\Omega)$ ) [Eva22]. Every locally integrable function and every finite Radon measure defines a distribution by integration against test functions, allowing for distributions to serve as a unified generalization of both functions and measures. In particular, the Dirac delta measure at  $x_0$  corresponds to the distribution defined by  $\delta_{x_0}[\varphi] = \varphi(x_0)$ .

theory and algorithms commonly employed in the analysis of fluid flows and other complex physical systems.

### 1.3.1 The Manifold Hypothesis and Dimensionality Reduction

In fluid mechanics, one typically encounters high-dimensional state vectors  $x \in \mathbb{R}^n$  arising from spatial discretizations of velocity, pressure, or other field variables, where the ambient dimension  $n$  can be extremely large. In contrast, the number of available snapshots is often comparatively small, making direct function approximation often ill-posed and susceptible to the curse of dimensionality, in which sample complexity grows rapidly with increasing dimension.

Despite this apparent difficulty, many fluid systems exhibit strong low-dimensional structure. This observation is often interpreted through the manifold hypothesis, which posits that high-dimensional data are concentrated near a low-dimensional manifold  $\mathcal{M} \subset \mathbb{R}^n$  with intrinsic dimension  $m \ll n$  [FMN16, KS23, Don00, GT18]. One motivation for this hypothesis comes from the geometry of high-dimensional spaces. In high dimensions, the concentration of measure phenomenon implies that many geometric quantities, such as distances and norms, become sharply concentrated around their mean values [Led01]. Additionally, many physical systems evolve according to a comparatively small number of degrees of freedom imposed by conservation laws, governing equations, and other physical constraints. For example, dissipative partial differential equations frequently generate attracting sets with finite effective dimension, such as inertial manifolds in strongly dissipative regimes [DT91, FST88, Tem89, FMR01, Tem12], while long-term dynamics of chaotic flows evolve on strange attractors [Lor63, Kur78]. As a result, although observations are embedded in a high-dimensional ambient space, they are often confined to a subset of much lower intrinsic dimension. This provides the geometric motivation for modeling the data as lying on or near a low-dimensional manifold.

Here we use the term *manifold* in the standard topological sense, although in many



applications it is natural to assume additional smooth structure [Lee03]. A topological space  $\mathcal{M}$  is called an *m-dimensional topological manifold* (or simply an *m-manifold*) if it satisfies the following properties:

- **Hausdorff:** For any distinct points  $p, q \in \mathcal{M}$ , there exist disjoint open sets  $U, V \subseteq \mathcal{M}$  such that  $p \in U$  and  $q \in V$ .
- **Second countable:** The topology of  $\mathcal{M}$  admits a countable basis.
- **Locally Euclidean:** Every point  $p \in \mathcal{M}$  admits an open neighborhood  $U \subseteq \mathcal{M}$  and a homeomorphism

$$\varphi : U \rightarrow \tilde{U} \subseteq \mathbb{R}^m,$$

where  $\tilde{U}$  is an open subset of  $\mathbb{R}^m$ .

The pair  $(U, \varphi)$  is called a *chart*, and a collection of charts whose domains cover  $\mathcal{M}$  forms an *atlas*. Charts provide local coordinate representations of the space, while the associated transition maps between overlapping charts are required to be continuous.

While this topological structure is sufficient to describe the notion of a space that is locally Euclidean, it does not provide a framework for differentiation. In many applications in fluid dynamics and dynamical systems, it is therefore natural to endow  $\mathcal{M}$  with additional regularity, leading to the notion of a *smooth (differentiable) manifold*. In this setting, the transition maps between overlapping charts are assumed to be smooth, which allows for the definition of tangent spaces, vector fields, and differentiable dynamical systems on  $\mathcal{M}$ .<sup>4</sup> This additional structure is essential when interpreting fluid flows as evolution equations constrained to a low-dimensional geometric set embedded in a high-dimensional ambient space. A smooth bijection  $f : \mathcal{M} \rightarrow \mathcal{N}$  between smooth manifolds whose inverse is also smooth is called a *diffeomorphism*. If such a map exists,  $\mathcal{M}$  and  $\mathcal{N}$  are said to be *diffeomorphic*, and

---

<sup>4</sup>A smooth structure is defined by requiring that for any overlapping charts  $(U, \varphi)$  and  $(V, \psi)$ , the transition map  $\psi \circ \varphi^{-1} : \varphi(U \cap V) \rightarrow \psi(U \cap V)$  is smooth. An atlas satisfying this condition is called a smooth atlas.

are regarded as equivalent as smooth manifolds, since they share the same differential structure. In particular, diffeomorphic manifolds admit equivalent notions of smooth functions, tangent spaces, and vector fields, and thus support the same class of differentiable dynamical systems up to smooth change of coordinates.

From the perspective of fluid dynamics, this notion of equivalence is relevant when interpreting low-dimensional state-space representations, as different coordinate descriptions of the same underlying invariant set may be related by diffeomorphic transformations. While the precise mathematical details are beyond the scope of this thesis, we refer the reader to [Lee03] for a more complete introduction to smooth manifolds and differential geometry.

While the manifold hypothesis postulates that high-dimensional data concentrate near a low-dimensional manifold  $\mathcal{M} \subset \mathbb{R}^n$ , it is natural to ask whether such an embedding into a Euclidean space is always mathematically possible. A fundamental result addressing this question is provided by Whitney’s embedding theorem.

Whitney’s embedding theorem states that any smooth  $m$ -dimensional manifold  $\mathcal{M}$  admits a smooth embedding into  $\mathbb{R}^{2m}$ , and moreover into  $\mathbb{R}^{2m+1}$  in the case of a generic embedding [Lee03]. That is, there exists a smooth injective map  $f : \mathcal{M} \hookrightarrow \mathbb{R}^{2m}$  whose differential is everywhere injective, and whose image is a submanifold diffeomorphic to  $\mathcal{M}$ . In particular, this result guarantees that any abstract smooth state space arising in fluid mechanics can, in principle, be realized as a smooth submanifold of a finite-dimensional Euclidean space.

From the perspective of fluid mechanics, Whitney’s theorem provides a theoretical justification for the common practice of representing infinite-dimensional flow fields through finite-dimensional coordinate embeddings. In particular, when the set of states of a fluid system exist on some finite-dimensional manifold (such as an inertial manifold or attractor for the long term dynamics of dissipative systems), Whitney’s theorem ensures that there exists an embedding into a finite-dimensional Euclidean space that preserves the smooth structure of the original manifold. Thus, we can interpret reduced-order models as coordinate representations of an underlying physical geometric object, rather than merely low-rank

approximations of data. The preservation of the smooth structure of the underlying data manifold is essential for applications in downstream tasks, such as interpolation, optimization, and control.

However, the theorem is fundamentally existential and does not provide constructive embeddings, nor does it guarantee that the embedding dimension  $2m$  is small in practical settings. In fluid dynamics applications, the intrinsic dimension  $m$  of the relevant attractor or invariant set is typically unknown and may itself be large or only weakly finite-dimensional. As a result, while Whitney’s theorem establishes that finite-dimensional smooth representations are always possible in principle, it does not by itself resolve the challenges of learning such representations from data.

This observation further motivates data-driven approaches in fluid mechanics, where the goal is not merely to embed a manifold into Euclidean space, but to learn a representation that is both low-dimensional and physically meaningful. In this context, nonlinear manifold learning methods, such as autoencoders and related neural architectures, can be interpreted as attempts to approximate unknown embeddings from finite, potentially noisy observations of fluid flow data.

In many practical engineering applications, however, the situation is further complicated by the fact that one does not always observe a single autonomous dynamical system evolving on a fixed attractor. Instead, one often considers ensembles of flow fields depending on external parameters (such as Reynolds number, boundary conditions, forcing, etc.), or data consisting only of time-averaged quantities rather than full temporal trajectories. In these settings, the notion of a single invariant manifold generated by a dynamical system may no longer be directly applicable. Consequently, one is led to interpret the available data as samples from a more general geometric or probabilistic structure, rather than as observations of a single low-dimensional dynamical evolution.

When the manifold of interest  $\mathcal{M}$  is well-approximated by a linear subspace, classical linear dimensionality reduction techniques such as proper orthogonal decomposition

(POD) [LYT67, BHL93, TBD17] and principal component analysis (PCA) [Hot33, Jol02, AW10] provide efficient and interpretable reduced representations. In fluid mechanics, this corresponds to the prevalence of coherent structures, such as large-scale vortices or dominant instabilities. However, in many regimes of practical interest, the underlying manifold can be nonlinear, and energy is distributed across a broader range of dynamically coupled spatio-temporal scales. In these settings, linear subspace methods require a rapidly increasing number of modes to achieve accurate reconstruction and may fail to capture the geometry of the underlying state manifold.

To address these limitations, nonlinear manifold learning techniques have been developed to better represent curved and dynamically complex state spaces. In particular, autoencoder-based encoder–decoder architectures have emerged as a flexible and scalable approach for learning nonlinear embeddings of fluid flow data, and have demonstrated strong performance across a range of applications in fluid mechanics [HS06, MFF20, VB22, ZLG22, RDM23, FT23, SFS24, TFI24, ME25].

### 1.3.2 Proper Orthogonal Decomposition

As mentioned previously, when the data lie “near” a low-dimensional linear submanifold, Proper Orthogonal Decomposition (POD) is a simple and effective approach for dimensionality reduction. Originally introduced to the fluid mechanics community for identifying coherent structures in turbulence [LYT67], POD is closely related to principal component analysis (PCA) [Hot33, Jol02, AW10], the Karhunen–Loève (KL) decomposition [Kar46, Kar47, Loe78], and the singular value decomposition (SVD) [Ger81, TBD17]. Intuitively, POD seeks an optimal orthonormal basis whose modes span a linear subspace that minimizes the mean squared projection error of the data.

A classical derivation of the POD basis is summarized as follows [KV02, Row02]. Consider a dataset of flow fields  $\{x_i\}_{i=1}^N$  where the data points  $x_i$  belong to some Hilbert space  $\mathcal{X}$  with inner product  $\langle \cdot, \cdot \rangle_{\mathcal{X}}$ . Often  $x_i$  have been preprocessed such that the sample mean of the

dataset is removed. Define

$$\mathcal{V} = \text{span} \left( \{x_i\}_{i=1}^N \right),$$

where  $\mathcal{V} \subseteq \mathcal{X}$  by construction. Let  $\{\phi_j\}_{j=1}^d$  be an orthonormal basis of  $\mathcal{V}$ , where  $d = \dim(\mathcal{V})$ .

Then any element of  $\mathcal{V}$  can be expressed as a linear combination of the basis elements:

$$x_i = \sum_{j=1}^d \langle x_i, \phi_j \rangle_{\mathcal{X}} \phi_j, \quad i = 1, \dots, N.$$

The POD basis is characterized as the orthonormal basis that optimally captures the variance of the dataset. Equivalently, the POD modes arise as extremals of an energy functional. For a unit vector  $\phi \in \mathcal{X}$ , define the functional

$$\mathcal{J}(\phi) = \sum_{i=1}^N |\langle x_i, \phi \rangle_{\mathcal{X}}|^2, \quad \text{subject to } \|\phi\|_{\mathcal{X}} = 1.$$

The quantity  $\mathcal{J}(\phi)$  measures the total projected energy of the dataset onto the direction  $\phi$ .

The first POD mode  $\phi_1$  is defined as the maximizer of this functional:

$$\phi_1 = \arg \max_{\substack{\phi \in \mathcal{X} \\ \|\phi\|_{\mathcal{X}}=1}} \mathcal{J}(\phi).$$

Subsequent POD modes are obtained recursively by maximizing the same functional over directions orthogonal to the previously computed modes:

$$\phi_k = \underset{\substack{\|\phi\|_{\mathcal{X}}=1 \\ \langle \phi, \phi_j \rangle_{\mathcal{X}}=0, j < k}}{\text{argmax}} \mathcal{J}(\phi), \quad k = 2, \dots, d.$$

This variational characterization is equivalent to minimizing the mean-square projection

error. Indeed, for a truncated basis of rank  $\ell$ , the POD basis solves

$$\min_{\{\phi_j\}_{j=1}^\ell} \sum_{i=1}^N \left\| x_i - \sum_{j=1}^\ell \langle x_i, \phi_j \rangle_{\mathcal{X}} \phi_j \right\|_{\mathcal{X}}^2$$

subject to

$$\langle \phi_i, \phi_j \rangle_{\mathcal{X}} = \delta_{ij}, \quad 1 \leq \ell \leq d.$$

The solution to this variational problem can be obtained by considering the covariance operator  $\mathcal{R}_N \in \mathcal{L}(\mathcal{X})$ , where  $\mathcal{L}(\mathcal{X})$  denotes the Banach space of bounded linear operators on  $\mathcal{X}$ . Define

$$\mathcal{R}_N \tilde{x} = \sum_{i=1}^N \langle \tilde{x}, x_i \rangle_{\mathcal{X}} x_i, \quad \tilde{x} \in \mathcal{X}.$$

Observe that the functional  $\mathcal{J}$  can be written in operator form

$$\mathcal{J}(\phi) = \langle \mathcal{R}_N \phi, \phi \rangle_{\mathcal{X}}, \quad \|\phi\|_{\mathcal{X}} = 1.$$

Hence, the POD modes are the extremals of this functional under the unit norm constraint.

Introducing a Lagrange multiplier  $\lambda$  for the normalization constraint, define the augmented functional

$$\mathcal{L}(\phi, \lambda) = \langle \mathcal{R}_N \phi, \phi \rangle_{\mathcal{X}} - \lambda (\langle \phi, \phi \rangle_{\mathcal{X}} - 1).$$

For  $\phi$  to be extremal, the first variation with respect to  $\phi$  in the direction  $\psi \in \mathcal{X}$  must be equal to 0 which yields

$$\left. \frac{d}{d\varepsilon} \mathcal{L}(\phi + \varepsilon\psi, \lambda) \right|_{\varepsilon=0} = 0.$$

Expanding this expression gives

$$\begin{aligned} 0 &= \left. \frac{d}{d\varepsilon} (\langle \mathcal{R}_N(\phi + \varepsilon\psi), \phi + \varepsilon\psi \rangle_{\mathcal{X}} - \lambda \langle \phi + \varepsilon\psi, \phi + \varepsilon\psi \rangle_{\mathcal{X}}) \right|_{\varepsilon=0} \\ &= \langle \mathcal{R}_N \psi, \phi \rangle_{\mathcal{X}} + \langle \mathcal{R}_N \phi, \psi \rangle_{\mathcal{X}} - \lambda (\langle \psi, \phi \rangle_{\mathcal{X}} + \langle \phi, \psi \rangle_{\mathcal{X}}). \end{aligned}$$

Using the self-adjointness of  $\mathcal{R}_N$ , we obtain

$$2\langle \mathcal{R}_N\phi - \lambda\phi, \psi \rangle_{\mathcal{X}} = 0, \quad \forall \psi \in \mathcal{X}.$$

The left-hand side defines a zero bounded linear functional on  $\mathcal{X}$ . By the Riesz representation theorem,<sup>5</sup> the representing vector is unique, and therefore

$$\mathcal{R}_N\phi - \lambda\phi = 0.$$

Hence, the first-order optimality condition is

$$\mathcal{R}_N\phi = \lambda\phi.$$

Note that  $\mathcal{R}_N$  is a bounded, self-adjoint, and positive-semidefinite operator. Since the range of  $\mathcal{R}_N$  is finite dimensional,  $\mathcal{R}_N$  is a finite-rank operator and therefore compact. By the Hilbert–Schmidt spectral theorem (i.e., the spectral theorem for compact self-adjoint operators on a Hilbert space), the nonzero spectrum of  $\mathcal{R}_N$  consists of finitely many positive eigenvalues  $\{\lambda_j\}_{j=1}^d$ , with corresponding orthonormal eigenvectors  $\{\phi_j\}_{j=1}^d$  spanning  $\text{range}(\mathcal{R}_N)$ . In particular,

$$\mathcal{R}_N\phi_j = \lambda_j\phi_j, \quad \lambda_j > 0, \quad j = 1, \dots, d.$$

Moreover, the theorem yields the orthogonal decomposition

$$\mathcal{X} = \text{range}(\mathcal{R}_N) \oplus \ker(\mathcal{R}_N),$$

---

<sup>5</sup>Here we use the Riesz representation theorem for Hilbert spaces (see, e.g., [Lax14]): every bounded linear functional  $F$  on a Hilbert space  $\mathcal{X}$  can be written uniquely as

$$F(v) = \langle u, v \rangle_{\mathcal{X}}, \quad \forall v \in \mathcal{X},$$

for a unique  $u \in \mathcal{X}$ .

where  $\ker(\mathcal{R}_N)$  is a closed subspace. Choosing an orthonormal basis of  $\ker(\mathcal{R}_N)$  and combining it with  $\{\phi_j\}_{j=1}^d$  produces an orthonormal basis of  $\mathcal{X}$  consisting entirely of eigenvectors of  $\mathcal{R}_N$ . For each  $\phi \in \ker(\mathcal{R}_N)$ ,

$$\mathcal{R}_N \phi = 0,$$

so the remaining spectrum consists solely of the eigenvalue 0, whose multiplicity is  $\dim(\ker(\mathcal{R}_N))$  (possibly infinite).

The eigenfunctions  $\{\phi_j\}_{j=1}^d$  are the POD modes, and the associated eigenvalues quantify the energy captured by each mode. Moreover, the optimal projection error associated with a rank- $\ell$  approximation satisfies

$$\sum_{i=1}^N \left\| x_i - \sum_{j=1}^{\ell} \langle x_i, \phi_j \rangle_{\mathcal{X}} \phi_j \right\|_{\mathcal{X}}^2 = \sum_{j=\ell+1}^d \lambda_j.$$

Physically, if the data  $x_i$  correspond to snapshots of velocity fields  $u_i : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3$  with  $\mathcal{X} = L^2(\Omega; \mathbb{R}^3)$  then the eigenvalues from the POD algorithm correspond to the kinetic energy captured in the respective POD modes, and this error formula corresponds to the amount of kinetic energy captured by the POD basis of rank  $\ell$ . Note that when  $\mathcal{X}$  is finite dimensional, POD is equivalent to PCA.

This formulation represents the classical version of the POD problem. Numerous extensions of POD have since been developed for specialized applications, including spectral POD [SPO16] and gappy POD [ES95, Wil06]. Despite the widespread success of linear modal decomposition techniques in engineering and fluid mechanics, methods such as POD are fundamentally constrained by the Kolmogorov barrier.

To quantify the best possible performance of any linear reduced-order model, if  $\mathcal{M} \subset \mathcal{X}$  is the solution manifold, i.e., the set of all states of interest. The *Kolmogorov  $m$ -width* of  $\mathcal{M}$



is defined by

$$d_m(\mathcal{M}) = \inf_{\substack{V_m \subset \mathcal{X} \\ \dim(V_m)=m}} \sup_{u \in \mathcal{M}} \inf_{v \in V_m} \|u - v\|_{\mathcal{X}}.$$

In this definition,  $m$  refers to the dimension of the linear subspace used for approximation. The Kolmogorov  $m$ -width therefore measures the smallest possible worst-case approximation error achievable by any  $m$ -dimensional linear subspace of  $\mathcal{X}$ . In other words, no linear reduced-order model of dimension  $m$  can achieve a worst-case error smaller than  $d_m(\mathcal{M})$ . Consequently, the decay rate of  $d_m(\mathcal{M})$  determines the intrinsic efficiency of all linear approximation methods, including POD.

For transport-dominated systems, such as high-Reynolds-number turbulent flows, the Kolmogorov  $m$ -width typically decays slowly [Peh22]. Consequently, the projection error associated with linear reduced-order models also decays slowly as the basis dimension increases. Physically, this reflects the fact that turbulent flows contain coherent structures spanning a wide range of spatial and temporal scales, making them difficult to represent efficiently using a low-dimensional linear subspace. As a result, linear approaches such as POD often require a prohibitively large number of modes to achieve accurate flow reconstruction in turbulent regimes.

### 1.3.3 Autoencoders

Deep neural network autoencoders are a popular choice for data-driven manifold learning [HS06]. Let  $\{x_i\}_{i=1}^N$  be a dataset with  $x_i \in \mathbb{R}^n$  and suppose that this dataset is sampled from a submanifold  $\mathcal{M} \subseteq \mathbb{R}^n$ . An autoencoder consists of an encoder–decoder pair  $(\mathcal{E}, \mathcal{D})$ :

$$\begin{aligned} \mathcal{E} : \mathbb{R}^n &\rightarrow \mathbb{R}^m, & \xi &= \mathcal{E}(x), \\ \mathcal{D} : \mathbb{R}^m &\rightarrow \mathbb{R}^n, & \hat{x} &= \mathcal{D}(\xi). \end{aligned}$$

Here,  $\xi \in \mathbb{R}^n$  is the low-dimensional latent representation with  $m \ll n$  and  $\hat{x} = \mathcal{D}(\mathcal{E}(x))$  is the reconstruction. The goal of the autoencoder is to learn an  $(\mathcal{E}, \mathcal{D})$  such that  $x = \mathcal{D}(\mathcal{E}(x))$  for all  $x \in \mathcal{M}$ . More concretely, the autoencoder is trying to learn a homeomorphism between  $\mathcal{M}$  and a subset of the latent space  $\mathbb{R}^m$ . The overall autoencoder therefore defines a nonlinear parameterized map

$$\mathcal{D} \circ \mathcal{E} : \mathbb{R}^n \rightarrow \mathbb{R}^n,$$

which is trained to act as an identity approximation on the data manifold  $\mathcal{M}$ .

The autoencoder is typically parameterized using neural networks. These networks consist of layers, with each layer consisting an affine map and a nonlinear activation function. For example, if  $\theta$  are the parameters of the neural network, the encoder can be expressed as:

$$\mathcal{E}(x) = \mathcal{E}^{(L)} \circ \dots \circ \mathcal{E}^{(1)}(x), \quad \text{where} \quad \mathcal{E}^{(l)}(x^{(l-1)}) = \sigma(W_{\theta}^{(l)} x^{(l-1)} + b_{\theta}^{(l)}),$$

where  $W_{\theta}^{(l)}$  is the weight matrix,  $b_{\theta}^{(l)}$  is the bias, and  $\sigma$  is a nonlinear activation function.

Essentially, the encoder  $\mathcal{E}$  belongs to a parameterized function class where the depth  $L$  controls the hierarchical structure of the learned representation. Each intermediate variable  $x^{(l)}$  defines a progressively more abstract representation of the input:

$$x^{(l)} = \mathcal{E}^{(l)}(x^{(l-1)}), \quad l = 1, \dots, L,$$

with  $x^{(0)} = x$  and  $\xi = x^{(L)}$ . The decoder  $\mathcal{D}$  is defined analogously as a composition of affine maps and nonlinearities mapping from latent space back to the ambient space.

From this perspective, the latent variable  $\xi = \mathcal{E}(x)$  defines a coordinate chart on (a subset of)  $\mathcal{M}$ , while the decoder  $\mathcal{D}$  acts as a learned parameterization of the embedding of this coordinate system into  $\mathbb{R}^n$ . Unlike classical coordinate charts, however, both mappings are restricted to lie within the expressive capacity of the chosen neural network architecture and are learned solely from data.

The parameters of the autoencoder are typically obtained through empirical risk minimization. Let  $P_{\text{data}}$  denote the (unknown) data-generating probability measure supported on  $\mathcal{M} \subseteq \mathbb{R}^n$ . The ideal training objective can be written as the expected reconstruction error

$$\min_{\mathcal{E}, \mathcal{D}} \mathbb{E}_{x \sim P_{\text{data}}} [\|x - \mathcal{D}(\mathcal{E}(x))\|_2^2].$$

In practice, the measure  $P_{\text{data}}$  is unknown and only a finite dataset  $\{x_i\}_{i=1}^N$  is available. The corresponding empirical probability measure is

$$\hat{P}_N = \frac{1}{N} \sum_{i=1}^N \delta_{x_i},$$

where  $\delta_{x_i}$  denotes the Dirac measure concentrated at  $x_i$ . Expanding the expectation operator and substituting the empirical measure  $\hat{P}_N$  in place of the probability measure gives

$$\begin{aligned} \mathbb{E}_{x \sim \hat{P}_N} [\|x - \mathcal{D}(\mathcal{E}(x))\|_2^2] &= \int_{\mathbb{R}^n} \|x - \mathcal{D}(\mathcal{E}(x))\|_2^2 \hat{P}_N(dx) \\ &= \int_{\mathbb{R}^n} \|x - \mathcal{D}(\mathcal{E}(x))\|_2^2 \left( \frac{1}{N} \sum_{i=1}^N \delta_{x_i}(dx) \right) \\ &= \frac{1}{N} \sum_{i=1}^N \int_{\mathbb{R}^n} \|x - \mathcal{D}(\mathcal{E}(x))\|_2^2 \delta_{x_i}(dx). \end{aligned}$$

Here we made use of the linearity of the integral with respect to measures. Using the defining property of the Dirac measure yields,

$$\mathbb{E}_{x \sim \hat{P}_N} [\|x - \mathcal{D}(\mathcal{E}(x))\|_2^2] = \frac{1}{N} \sum_{i=1}^N \|x_i - \mathcal{D}(\mathcal{E}(x_i))\|_2^2.$$

which is the standard squared mean square error reconstruction loss used in practice. Thus, the autoencoder training can be interpreted as minimizing the expected reconstruction error with respect to the empirical probability measure supported on the training samples.

In many practical settings, where the underlying data manifold may not be homeomorphic

to a subset of  $\mathbb{R}^m$ , such as if  $m$  is chosen too small in comparison to the dimension of  $\mathcal{M}$ , autoencoders still exhibit surprising performance as the set of  $x \in \mathcal{M}$  such that  $x \neq \mathcal{D}(\mathcal{E}(x))$  can be made arbitrarily small with respect to the intrinsic measures on  $\mathcal{M}$  [KS23]. In fluid mechanics, this proves to be a valuable fact as it enables us to represent highly complex flows in much fewer dimensions than linear methods with typically higher accuracy.

It is important to note that the learned latent representation is inherently non-unique. Given an optimal encoder–decoder pair  $(\mathcal{E}, \mathcal{D})$ , one may compose the encoder with any invertible transformation of the latent space and compensate with its inverse in the decoder. Specifically, for any diffeomorphism  $T : \mathbb{R}^m \rightarrow \mathbb{R}^m$ , the transformed pair

$$\hat{\mathcal{E}} = T \circ \mathcal{E}, \quad \hat{\mathcal{D}} = \mathcal{D} \circ T^{-1}$$

achieves identical reconstruction error, since

$$\hat{\mathcal{D}}(\hat{\mathcal{E}}(x)) = \mathcal{D}(\mathcal{E}(x)).$$

As a consequence, the latent coordinates are defined only up to such reparameterizations, and do not possess an intrinsic geometric or physical interpretation without additional constraints or structure imposed on the model.

Despite this, autoencoders have demonstrated strong empirical performance across a range of fluid mechanics applications, including flow field analysis [MFF20, VB22, FT23, SFS24], flow control [ZLG22, LZG23], state estimation [LG22, RDM23, ME25], and design optimization [TFI24, VEN25].

## CHAPTER 2

# Data-Driven Dimensionality Reduction for Aerodynamic Design Optimization of Automobiles

In this chapter, we discuss the use of an autoencoder for the aerodynamic design optimization of industry vehicle geometries. This chapter is adapted from a study performed in collaboration with Honda Motor Co., Ltd., Tochigi, Japan, with results published in [TFI24].

### 2.1 Background and Motivation

The shift from traditional combustion-based transportation to electric vehicles has brought about notable changes in vehicle design, particularly in terms of aerodynamics. The absence of a combustion engine has allowed for a reevaluation of design constraints and priorities, offering automotive designers greater creative freedom. For example, the elimination of the requirement for engine cooling has allowed for the transformation of the front geometry of electric vehicles from both a functional and stylistic perspective. However, the transition to electric vehicles also brings new concerns. Even as battery and motor efficiencies improve, the current performance of many electric vehicles struggles to meet the demands of consumers, with many electric vehicles operating with limited driving ranges despite long charging times [STG21]. The importance of vehicle efficiency has been exacerbated by the transition to sustainable transportation. The departure from conventional design norms is emblematic of the need for novel methods to accelerate the design and product development process of vehicles.

On a broader scope, the transportation sector plays a pivotal role in the global energy landscape. Transportation of people and goods constitutes approximately one-quarter of the world’s energy consumption by end users, contributing to nearly 40% of global emissions [iea22]. Oil accounts for 90% of energy consumption in transportation, with oil for use in road transport alone responsible for almost 45% of global oil demand [iea22, iea23]. Additionally, transportation is directly responsible for approximately 14% of total global greenhouse gas emissions, producing around 7 billion tonnes of CO<sub>2</sub> [EPS14, Kal18]. Unfortunately, the environmental burden associated with the transportation sector is only expected to increase. Already worldwide there are more than 1.4 billion passenger and commercial automobiles — the majority of which are powered by internal combustion engines — with this number only projected to go up [ZMM23]. Due to global growth in population and income, the demand for passenger and freight travel is projected to increase up to 41% from 2022 to 2050, which only serves to further exacerbate the growing climate crisis [DG99, eia23]. All of the aforementioned facts suggest that it is especially important for vehicle manufacturers to explore methods to push the boundaries of vehicle efficiency.

Transportation fundamentally relies on the conversion of chemical energy from fuel into mechanical motion. Accordingly, vehicles must waste excess energy to overcome driving resistances. This wasted energy manifests itself as a decrease in fuel efficiency, resulting in increased emissions and reduced range. A notable source of resistance comes from aerodynamic drag, which can account for about 75-80% of the total driving resistance when driving at 100 km/h [NOT15, MKO15]. In the case of automobiles, a 20% reduction of aerodynamic drag can result in a noticeable 4% decrease in fuel consumption of highway operation [ZFC16]. In the case of electric vehicles, without the associated losses due to combustion, the increased efficiency of the powertrain for electric vehicles results in a greater proportion of energy loss due to aerodynamic drag [SHS11, NYA24]. This makes aerodynamic performance a major design priority due to the direct relationship between drag and vehicle efficiency.

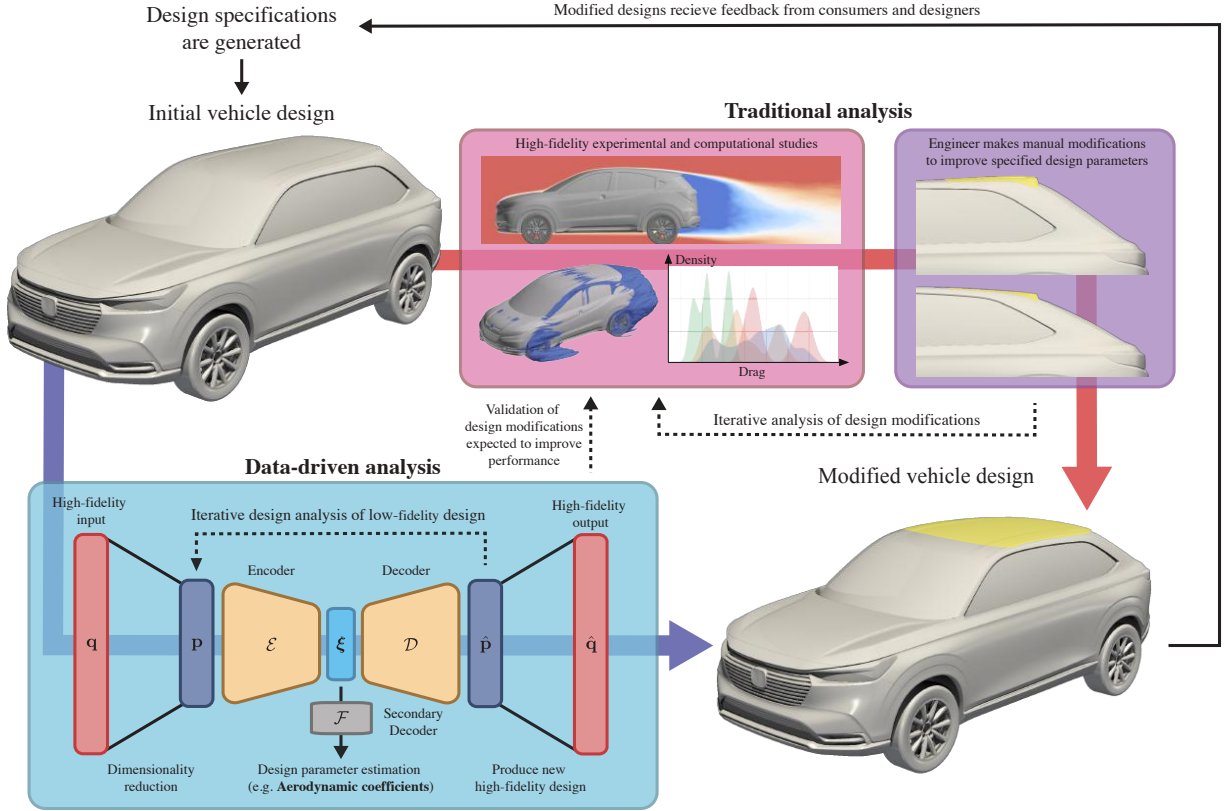


Figure 2.1: Overview of data-driven automobile design analysis.

As even small geometric changes may have large effects on the flow field around a vehicle — thereby influencing its aerodynamic performance — a careful analysis of the effect of parameter changes on aerodynamics is critical for vehicular design. Large-scale modifications can alter characteristics of the pressure distribution in the flow around a vehicle, which could considerably alter the pressure drag, which can account for up to almost 85% of the total drag on the vehicle [ARF84, LB03, HGR04]. Additionally, small-scale changes can modify the boundary layer, which can alter drag resulting from friction at the surface. However, predicting the aerodynamic performance is difficult due to the nonlinear nature of fluid flows, with even small uncertainties in estimating drag costing the US billions per year [CHS21].

Aerodynamic analysis of vehicle performance in relevant operating conditions often requires a comprehensive campaign of experimental or computational trials. Experimental studies including wind tunnel tests require a physical car model as well as expensive test-

ing facilities. High-fidelity computational fluid dynamics (CFD) studies based on direct numerical simulation or large-eddy simulations (LES) require substantial computational resources [NYA24]. To resolve the smallest turbulent scales of the flow, the number of grid points scales with the Reynolds number  $Re$ , a value that measures the ratio between inertial and viscous forces, on the order of  $Re^{9/4}$ . In the case of turbulent flow past an automobile in which the Reynolds number can well exceed millions, this task quickly becomes very expensive [KT16].

To make matters more difficult, both numerical and experimental trials need to be performed spanning an extremely large parameter space of vehicle designs with practically infinitely many possible configurations, becoming expensive campaigns in terms of both finance and time. Furthermore, the data size associated with vehicle analyses increases as experimental and computational techniques continuously improve in fidelity and accuracy. Consequently, such large amounts of high-dimensional data can make analysis extremely difficult. Along with such expensive analyses, optimization of vehicle aerodynamic performance also necessitates an extensive amount of manual modeling. These issues naturally call for data-driven approaches, such as machine learning, to learn the underlying physical relationships between vehicle geometries and aerodynamic performance to direct design analysis toward specific cases that are expected to improve aerodynamics. While trained aerodynamicists may be able to manually intuit a relationship between aerodynamic performance and geometric changes, we anticipate that deep machine-learning methods can learn such aerodynamic relationships from a large data set while offering physical insights into vehicle design, beyond that of human intuition.

The optimization of industrial designs necessitates in-depth, high-fidelity analysis of complex phenomena and complex geometries, which poses a critical bottleneck to the design process. The iterative nature of such design demands careful consideration of numerous factors and constraints, and any modifications to the design require thorough analysis. In this sense, developing a surrogate model that estimates design parameters of interest that



are associated with expensive, high-fidelity analyses can greatly improve the efficiency of the feasibility study cycle [CA21]. Performing data-driven shape optimization can help mitigate the cost burden associated with manual modeling and analysis for a naïve parameter sweep by producing modifications that are expected to improve aerodynamic performance, which can then be further studied. This problem setting is depicted in Fig. 2.1 for the case of automobile geometries.

We perform a data-driven analysis on industry-quality vehicle geometries for the purpose of drag reduction. Given that many high-dimensional data can be explained with fewer coordinates in a low-order manner [FMT88, Tem89, GT18, JG23], we seek to learn a relationship in an appropriate form for this purpose. Previous studies have demonstrated that a low-order description of vehicle geometries can be leveraged for vehicle shape optimization [RSM19, RSW21, DBW22, SYP23]. However, such studies have been limited to relatively lower-fidelity, steady-state, Reynolds-averaged Navier-Stokes (RANS) simulations, typically for simplified geometries and simulation conditions. In this work, we seek to utilize a data-driven study of vehicular aerodynamics in an industrial setting. The present analysis is performed on production designs, with aerodynamic performance obtained from high-fidelity Large-Eddy Simulations (LES) utilizing an experimentally validated computational setup [NYA24].

The current data-driven approach, as outlined in Fig. 2.1, is proposed as a method to direct traditional high-fidelity analysis in a cost-effective manner to accelerate the iterative product design process. By employing a modified nonlinear autoencoder, we learn a low-order manifold, which simultaneously contains a nonlinear relationship between vehicle geometries and their associated drag coefficients. We find that incorporating the drag coefficients in the autoencoder training process provides low-order coordinates that are suitable for design optimization. The learned manifold, which captures a nonlinear relationship between vehicle geometry and aerodynamic drag, is leveraged to obtain modifications for a given vehicle geometry to improve aerodynamic performance. The present work demonstrates that this

simple data-driven approach sufficiently captures trends in aerodynamically relevant features, which we validate through a CFD analysis of decoded geometries. While the present work primarily focuses on aerodynamic shape optimization, the framework we employ to perform this shape optimization is not specific to the choice of input geometry representations and design parameters.

## 2.2 Methods

### 2.2.1 Automobile Dataset Preparation

For this study, we consider a dataset of over 500 geometries, which consists of a variety of industry car models generated from over 20 baseline car designs. The baseline car designs consist of SUVs, hatchbacks, sedans, and box cars of varying sizes and shapes taken from industrial geometries. For each baseline model, additional cases are generated by modifications of various parameters such as the car width, height, and length. The flow fields around each vehicle are numerically simulated with a large eddy simulation using a Helmholtz flow solver with a Vreman sub-grid model [NYA24]. The free stream is specified with velocity  $U_\infty = 140 \text{ km h}^{-1}$ , density  $\rho_\infty = 1.205 \text{ kg m}^{-3}$ , and dynamic viscosity  $\mu_\infty = 1.822 \times 10^{-5} \text{ m}^2 \text{ s}^{-1}$ , resulting in a Reynolds number of up to around  $1.18 \times 10^7$  with the car length as the characteristic length scale. The flow is computed with time step  $\Delta t = 8 \times 10^{-5} \text{ s}$  and is allowed to initialize for  $0.32 \text{ s}$  (4000 steps), with statistics averaged over the next  $0.8 \text{ s}$  (10000 steps). Simulations are performed with the Fidelity CharLES flow solver [BHN17]. The domain boundary conditions are chosen to emulate the experimental wind tunnel setting. The floor geometry consists of no-slip walls and slip walls to represent the suction and non-suction regions of the wind-tunnel floor, and a moving ground condition is specified for the belt. The vehicle body is set to a wall boundary condition, and the tire and wheel geometries are set to a rotational velocity wall condition. Additionally, a moving mesh solver is used to compute the flow around the rotating wheel geometry [NYA24]. Approximately 64

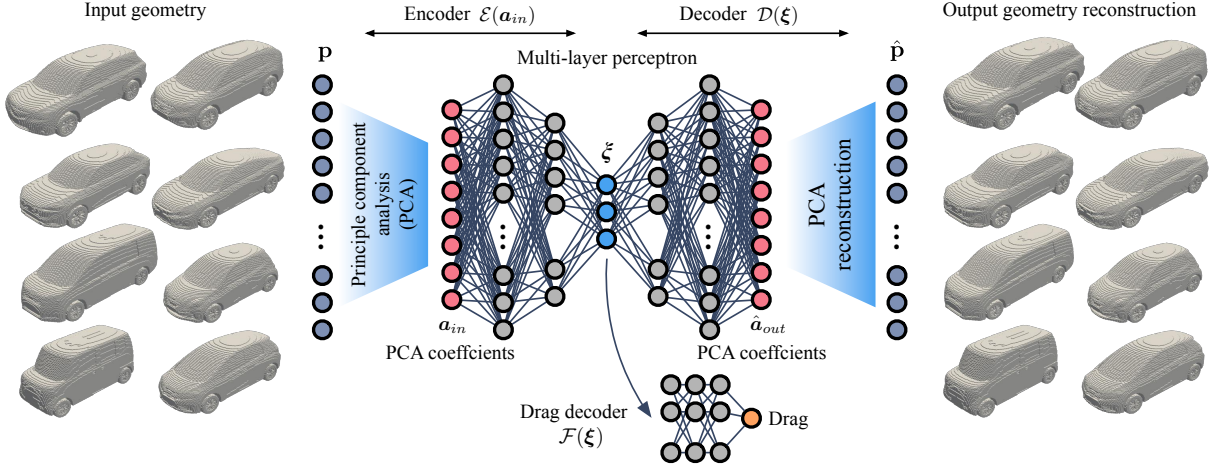


Figure 2.2: Principal component analysis (PCA) assisted observable augmented autoencoder. Voxelized vehicle geometries are taken as input and output and the drag coefficient is estimated from the encoded latent representation.

million control volumes are contained in the mesh, with around 0.6 million for the rotating wheel geometry. The total computational time for a single case is around 14950 CPU hours. This simulation setup has been validated with wind tunnel experiments to verify its accuracy in real-world conditions [NYA24]. The drag coefficient is defined as  $C_D = 2F_D/(\rho_\infty U_\infty^2 A)$  where  $F_D$  is the drag force and  $A$  is the cross-sectional area. The pressure coefficient is defined as  $C_p = 2(p - p_\infty)/(\rho_\infty U_\infty^2)$  where  $p - p_\infty$  is the local static pressure. The total pressure coefficient is  $C_{p,t} = 2p_t/(\rho_\infty U_\infty^2)$  where  $p_t = p + \frac{1}{2}\rho U^2$  is the total pressure. All coefficients present in the current work have been additionally normalized to preserve confidentiality. The vehicle geometry is converted to a uniform voxel grid with a fixed resolution of 20 mm per voxel.

### 2.2.2 PCA-Assisted Autoencoder setup

To identify a low-order manifold that captures the relationship between vehicle geometries and drag, we use an observable-augmented autoencoder consisting of a combination of principal component analysis and a multi-layer perceptron [HS06, RHW86, ASR21, FT23].

As illustrated in Fig. 2.2, the model consists of an encoder, a decoder, and a secondary drag decoder. The encoder,  $\mathcal{E} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ , takes the voxelized geometry as input  $\mathbf{p} \in \mathbb{R}^n$  and produces a low-order representation of the geometry  $\boldsymbol{\xi} \in \mathbb{R}^m$  with dimension  $m < n$ . On the other hand, the decoder,  $\mathcal{D} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ , takes the low-order state  $\boldsymbol{\xi}$  as input and produces a reconstruction of the input geometry,  $\hat{\mathbf{p}} \in \mathbb{R}^n$ . In addition, the secondary drag decoder,  $\mathcal{F} : \mathbb{R}^m \rightarrow \mathbb{R}$ , estimates the drag coefficient  $C_D \in \mathbb{R}$  from the low-order representation [FT23].

We first perform principal component analysis (PCA) [LYT67] of the voxelized body data to reduce the dimension of the geometry. Principal component analysis gives a set of orthogonal basis vectors, or modes, which capture variations in the shape designs with respect to the mean geometry. For a given dataset of geometries,  $\{\mathbf{p}_i\}_{i=1}^N$ , the PCA modes are given by the following optimization problem, which minimizes the difference between the original data and its projection onto a set of orthogonal PCA modes,

$$\{\boldsymbol{\phi}_k\}_{k=1}^r = \underset{\{\tilde{\boldsymbol{\phi}}_k\}_{k=1}^r}{\operatorname{argmin}} \sum_{i=1}^N \left\| \tilde{\mathbf{p}}_i - \sum_{k=1}^r a_{i,k} \tilde{\boldsymbol{\phi}}_k \right\|_2^2, \quad (2.1)$$

where  $\{\boldsymbol{\phi}_k\}_{k=1}^r$  is the set of PCA modes,  $\boldsymbol{\phi}_k \in \mathbb{R}^n$ , and  $\tilde{\mathbf{p}}_i = \mathbf{p}_i - \bar{\mathbf{p}}$  is the  $i$ th data input with the mean of the data set removed. We have that the PCA coefficient,  $a_{i,k} = \boldsymbol{\phi}_k^T \tilde{\mathbf{p}}_i \in \mathbb{R}$ , is the inner product of the  $i$ th data vector and the  $k$ th PCA mode. The solutions to this optimization problem correspond to the eigenvectors of the covariance matrix. Given the PCA coefficients associated with an input,  $\mathbf{p}_i$ , we can obtain the reconstructed output with

$$\hat{\mathbf{p}}_i = \bar{\mathbf{p}} + \sum_{k=1}^r a_{i,k} \boldsymbol{\phi}_k = \bar{\mathbf{p}} + \boldsymbol{\Phi} \mathbf{a}_i, \quad (2.2)$$

where  $\boldsymbol{\Phi}$  is a matrix consisting of the PCA modes  $\{\boldsymbol{\phi}_k\}_{k=1}^r$  as columns and  $\mathbf{a}_i$  is a column vector of the PCA coefficients.

The neural network portion of the autoencoder is a multi-layer perceptron (MLP) [RHW86] which inputs the PCA coefficients of a vehicle geometry and outputs a reconstruction of the

PCA coefficients and an estimate of the drag. The MLP finds a low-dimensional embedding of the PCA coefficients,  $\boldsymbol{\xi}$ , which it uses to produce a reconstruction of the PCA coefficients as well as the drag estimate. The output PCA coefficients are then used to obtain a reconstruction of the original input data using Eq. 2.2. In this sense, it can be seen as a nonlinear correction to the linear decomposition of PCA [ASR21]. The nonlinear activation functions that we use for the MLP layers are hyperbolic tangent,  $\tanh(s) = (e^s - e^{-s})/(e^s + e^{-s})$ . The network weights,  $\mathbf{w}$ , are obtained by minimizing a weighted sum of the mean squared error of the PCA coefficient reconstruction and the drag coefficient estimation:

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \left[ \frac{1}{N} \sum_{i=1}^N \left( \|\mathbf{a}_{\text{in},i} - \mathcal{D}(\mathcal{E}(\mathbf{a}_{\text{in},i}; \mathbf{w}_{\mathcal{E}}); \mathbf{w}_{\mathcal{D}})\|_2^2 + \beta \|C_{D,\text{ref},i} - \mathcal{F}(\mathcal{E}(\mathbf{a}_{\text{in},i}; \mathbf{w}_{\mathcal{E}}); \mathbf{w}_{\mathcal{F}})\|_2^2 \right) \right], \quad (2.3)$$

where  $\beta$  is a weighting parameter that adjusts the relative importance of the error terms associated with the  $C_D$  estimation and the PCA coefficient estimation. This parameter was chosen based on an L-curve analysis [HO93] and was taken to be  $\beta = 1 \times 10^5$  for our current model. The present MLP encoder takes in 400 PCA coefficients as input and consists of 8 layers with node counts of 400-512-256-128-64-32-16-3, with the MLP decoder symmetrically constructed. The drag decoder consists of 7 layers with node counts of 3-16-32-64-32-16-1. We split our dataset into training, validation, and test sets with a random 80:10:10 split. We use the Adam optimizer [KB15] for 1000 epochs with an early stopping criterion to avoid overfitting [Pre98]. During training, we save the model parameters when the loss for the validation set improves.

### 2.2.3 Shape Optimization Procedure

With the trained autoencoder model, we can perform geometry optimization to reduce the drag of a given design in an iterative manner. Consider a geometry at iteration  $j$  which we denote  $\mathbf{p}^{(j)}$ . We prepare the corresponding PCA coefficients,  $\mathbf{a}^{(j)} = \boldsymbol{\Phi}^T \mathbf{p}^{(j)}$ , and encode

to find the corresponding latent point  $\boldsymbol{\xi}^{(j)} = \mathcal{E}(\mathbf{a}^{(j)})$ , as well as the corresponding drag estimate  $C_D^{(j)} = \mathcal{F}(\boldsymbol{\xi}^{(j)})$ . By backpropagating through the drag decoder, we can obtain the sensitivity of the drag estimate with respect to the latent space point to obtain the direction of decreasing drag in the latent space,  $\mathbf{d}^{(j)} = -\nabla_{\boldsymbol{\xi}} C_D|_{\boldsymbol{\xi}=\boldsymbol{\xi}^{(j)}}$  [MFZ22]. With a user-specified step size,  $h^{(j)}$ , the modification in the latent space with respect to decreasing drag is  $\Delta\boldsymbol{\xi}^{(j)} = h^{(j)}\mathbf{d}^{(j)}$ .

To achieve a reliable geometry reconstruction and drag coefficient estimation, we also consider a penalty such that we do not move too far away from the training data. In the present work, we heuristically choose a threshold distance  $d_{\text{thresh}}$ , to be the average value of the distance between the training points. At every iteration, the distance between the latent space point,  $\boldsymbol{\xi}^{(j)}$ , and the nearest training data point in the latent space  $\boldsymbol{\xi}_{\text{train}}$  is measured. We modify the latent space perturbation to be the following:

$$\Delta\tilde{\boldsymbol{\xi}}^{(j)} = h^{(j)} \left[ \mathbf{d}^{(j)} - \alpha \frac{(\boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}_{\text{train}})}{d_{\text{thresh}}^2} (1 - \exp(-\|\boldsymbol{\xi}^{(j)} - \boldsymbol{\xi}_{\text{train}}\|_2^2 / d_{\text{thresh}}^2)) \right]. \quad (2.4)$$

The first term corresponds to the direction of decreasing drag in the latent space, while the second term penalizes moving away from the training data. The term  $\alpha$  controls the influence of the soft penalty term.

Once we obtain a perturbation in the latent space, we derive a new latent point,  $\boldsymbol{\xi}^{(j+1)} = \boldsymbol{\xi}^{(j)} + \Delta\tilde{\boldsymbol{\xi}}^{(j)}$ . Using the decoder, the corresponding geometry  $\mathbf{p}^{(j+1)} = \mathcal{D}(\boldsymbol{\xi}^{(j+1)})$  and drag estimate  $C_D^{(j+1)} = \mathcal{F}(\boldsymbol{\xi}^{(j+1)})$  can be computed. For the proof of concept shown in this work, we perform this iteration starting from a high-drag SUV geometry until the percent change of the estimated  $C_D$  drops under  $10^{-3}$  (which resulted in 40 iterations) with  $h = 0.02$  and  $\alpha = 0.125$ .

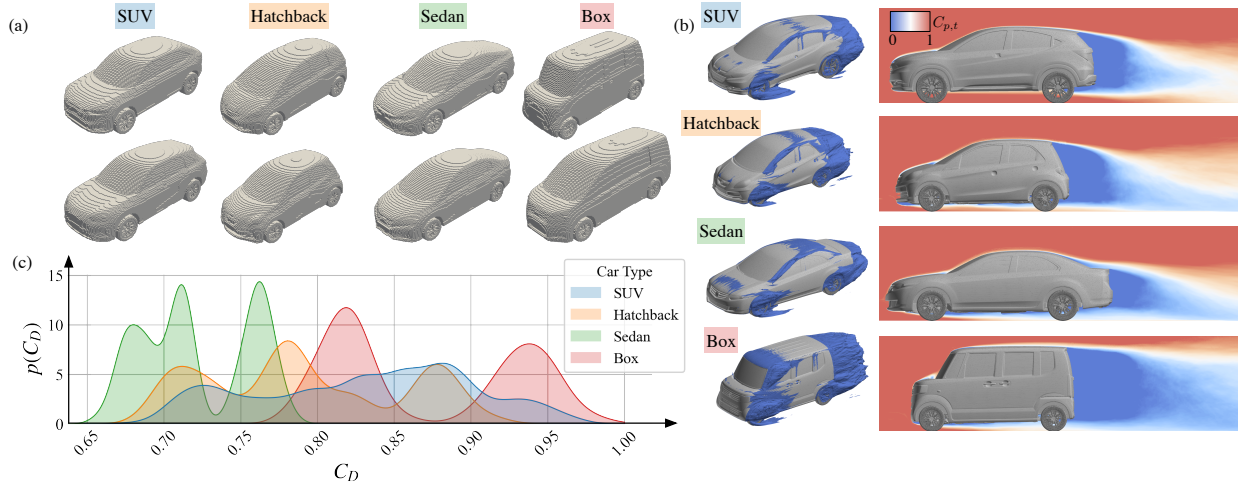


Figure 2.3: Depiction of the vehicle dataset consisting of sport utility vehicles (SUV), hatchbacks, sedans, and box geometries. (a) Representative voxelized vehicle geometries for different car types. (b) Example 3D and 2D (center plane) isocontours of normalized total pressure coefficient,  $C_{p,t}$ , for different car types.  $C_{p,t} = 0$  isocontour is shown indicating separated flow. (c) Probability density plot,  $p(C_D)$ , of normalized drag coefficient,  $C_D$ , with respect to vehicle type.

## 2.3 Results

### 2.3.1 Automobile Aerodynamic Analysis

Our data set spans a wide range of industry-quality automobile geometries which exhibit vastly different unsteady wake behavior and aerodynamic performance as shown by the voxelized geometries in Fig. 2.3(a). The baseline car designs considered in this study come from different production model car geometries including SUVs, hatchbacks, sedans, and box cars, with multiple vehicle models for each car type considered. Additionally, the features of the baseline designs for each model are parametrically modified to produce a variety of geometries. The flow is computed with a large eddy simulation utilizing a moving mesh that has been validated with industry wind tunnel experiments. Further details of the aerodynamic analysis are given in the Methods section.

The time-averaged flow around automobiles shares several salient features that are useful

in estimating the aerodynamic drag [ARF84, JMO21, GPI23, WRK23, YB21]. For even the most simplified automobile geometries, three-dimensional separated flow regions are generated in the wake behind the vehicle. The flow past most automobiles exhibits two recirculation regions that are formed when the flow separates at the top and bottom edges of the back of the vehicle. The size of the recirculation regions depends on how the shape of the rear geometry directs the trailing wake flow. A pair of trailing vortices is formed when the fluid boundary layer rolls around the sides of the vertical supports at the rear, referred to as the “C-pillars” [ARF84, LB03, HGR04]. These wake structures create a region of low pressure behind the vehicle and contribute to the induced drag. Conversely, a region of high pressure exists at the front of the vehicle where the flow is stopped by the front geometry. The difference in pressure over the surface of the vehicle results in the pressure drag, which is the primary source of drag force for automobiles [ARF84]. Fig. 2.3(b) depicts total pressure fields for the different car types in the data set. The zero total pressure isocontour is shown ( $C_{p,t} = 0$ ), which highlights where the flow may separate from the vehicle. We observe that in addition to the rear wake, we also see flow separation at other regions, including the wheels, the front roof edge, the front windshield edges (“A-pillar”), and the front bumper edges. We note that the use of the moving mesh LES solver captures physical phenomena not seen in previous analyses. One example is the wake generated from the outflow from the rotating front wheel, which produces appreciable differences in the drag estimate, but is not captured with a static mesh [NYA24]. With the variation of geometric features, the relative sizes of the aforementioned structures and the resulting aerodynamic performance can vary drastically from case to case. For example, if the transition between the front windshield and the top roof panel is abrupt, the flow may separate from the front of the roof (Box). A different flow pattern can appear in the rear where, depending on the rear windshield angle, the flow may separate over the rear spoiler only to reattach lower on the bottom of the windshield (Sedan). Fig. 2.3(c) shows a probability density plot of the normalized drag coefficients  $C_D$ , which demonstrates that the different geometries in our data set span vastly



different ranges of drag coefficients.

Although we can qualitatively analyze how fluid structures and aerodynamic performance relate to the vehicle geometry, the wide range of vehicle parameters and the nonlinear nature of fluid flow make it challenging to formulate aerodynamic prediction models, especially when comparing visually similar geometries. For example, the box car exhibits much larger separated flow regions at the front geometry, as well as a larger low-pressure wake behind the vehicle compared to the other geometries. However, we can see that in Fig. 2.3(c) some of the box cars in the current data set can exhibit a lower drag coefficient than some SUV and hatch-back designs, possibly due to differences in the front geometry. Other complexities arise in other flow interactions, such as ground effects in the wake and underbody regions, which can be profoundly altered by minor geometric changes, and require careful aerodynamic analysis. As such, the nonlinear behavior of the flow, confounded with a large number of geometric parameters, makes a reasonable prediction of the drag performance from geometries difficult to infer. However, we expect that there exists some nonlinear functional dependency between the vehicle geometry and its drag coefficient that can be captured through data-driven approaches. We aim to capture this relation utilizing a small number of latent variables that can be used as both a low-dimensional representation of a vehicle geometry as well as the estimated drag coefficient. For this objective, we use an observable augmented autoencoder assisted by principal component analysis (PCA) [FT23, LYT67, BHL93, ASR21, AW10] that reconstructs the high-dimensional car vehicle information while estimating the vehicle drag coefficient from the latent space, as illustrated in Fig. 2.2.

### 2.3.2 Latent Manifold Discovery

If we directly attempt to learn a relationship between the vehicle geometries and the drag coefficient for a given data set, we can learn many possible models that provide similar levels of accuracy. This can make shape optimization difficult, because such models may prove less reliable when generalizing to produce new geometries. With this in mind, we seek

low-dimensional coordinates that capture a relationship between input geometries and the drag coefficient, while reconstructing a geometry from the low-dimensional representation. Training both tasks simultaneously helps to mitigate overfitting as well as allows us to observe geometric similarities, which is helpful for identifying relevant features for design optimization.

The original vehicle geometry is first preprocessed to a voxelized representation. This represents the dimensionality reduction step in Fig. 2.1, as we effectively reduce our high-fidelity input from a mesh represented with a set of points connected by smooth, continuous segments to a regular grid with finite resolution. The voxelized geometries are then used for identifying a low-dimensional latent space. What this amounts to is learning a curvilinear coordinate system that parameterizes a manifold representation of our data set. We learn such a manifold by using a nonlinear autoencoder, a type of unsupervised machine learning model [HS06]. A basic autoencoder consists of an encoder, which reduces the dimension of input data into a lower-dimensional latent space, and a decoder, which reconstructs the input from the encoded representation.

For this work, we leverage an observable-augmented nonlinear autoencoder [?] that is simultaneously trained to estimate the drag coefficient from the compressed representation of the vehicle geometries, as illustrated in Fig. 2.2. Since the drag coefficient needs to be estimated from the latent variables, the identified latent representation holds a relationship between vehicle geometries and their aerodynamic performance in a low-order manner. In other words, the neural network weights are trained such that features relevant to estimations of the drag coefficient are captured. This can also reduce the problem of analyzing qualitative similarities of geometric trends in the original high-dimensional geometries to observing changes in salient features extracted in our low-dimensional space. Moreover, such an aerodynamically relevant manifold provides a desired direction to improve aerodynamic performance in a low-order manner. In other words, we can identify an optimal modification of the vehicle design with reduced computational cost. Further details of this approach are

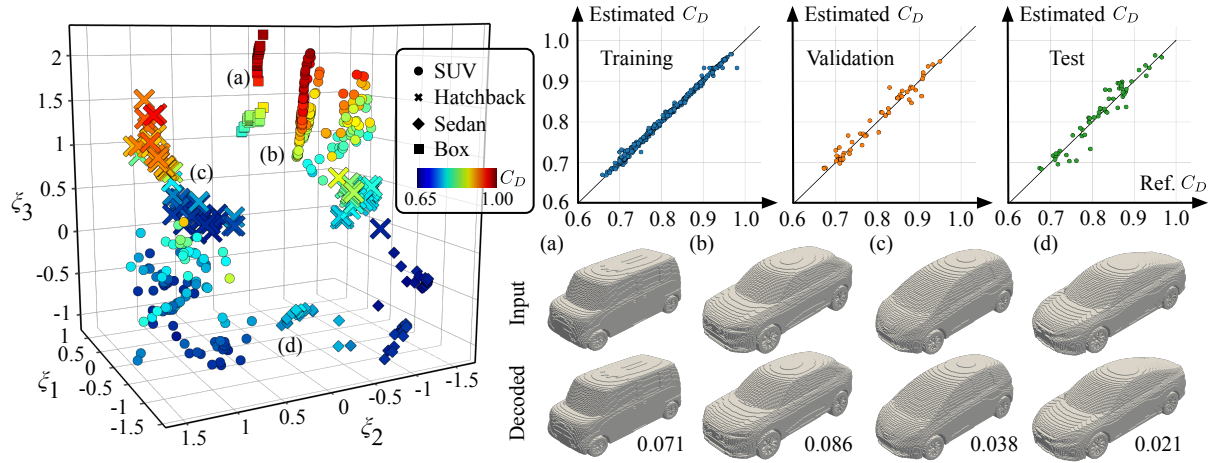


Figure 2.4: Nonlinear machine-learning-based compression of vehicle geometries. Left plot depicts latent space of sport utility vehicles (SUV), hatchbacks, sedans, and box geometries. Right parity plots show comparison of estimated and reference normalized drag coefficients  $C_D$ . Comparisons of example vehicle geometries and autoencoder reconstructions are shown. The value underneath each decoded geometry reports the percent  $L_2$  reconstruction error metric,  $\varepsilon = \|\mathbf{p} - \hat{\mathbf{p}}\|_2 / \|\mathbf{p}\|_2$ .

given in the Methods section.

Shown in Fig. 2.4 is the discovered three-dimensional latent space manifold,  $(\xi_1, \xi_2, \xi_3)$ , that is learned by the present autoencoder. We note that for our dataset, a three-dimensional latent space is enough to achieve sufficiently reasonable geometry reconstruction as well as accurate drag estimation, as we observed very little gain in accuracy without a substantial increase in the latent space dimension. Through the drag decoder, we can obtain an estimated drag coefficient corresponding to the geometry parameterized by any point in our three-dimensional latent space in Fig. 2.4. We also note that the geometry reconstruction from the decoder is qualitatively indistinguishable from the original input for a number of cases, depicted in Fig. 2.2 and Fig. 2.4. The average percent absolute error of estimated drag values lies within 2% of the reference value for the training, validation, and test sets. As seen in Fig. 2.4, each point in the low-dimensional space corresponds to a vehicle geometry. We also observe that vehicles cluster in the low-order space based on both geometric similarity

and the estimated drag coefficient, with distinct point clouds corresponding to the different car types (shown by the marker shape). Increased estimated drag approximately correlates with increasing  $\xi_1$  and  $\xi_3$  in the shown manifold.

The relative distances between latent points can be taken to be representative of the similarity between vehicle cases. For example, vehicles with very different geometries and aerodynamic performance, such as sedan and box cars, are placed far apart in the latent space. On the other hand, many of the hatchback geometries have an intermediate drag value, which is reflected by their placement in between the low-drag sedans and the high-drag SUV and box cars.

### 2.3.3 Data-Driven Shape Optimization

With the identified latent space manifold, we aim to modify existing vehicle geometries to improve aerodynamic performance. Since the present latent manifold relates car geometries to drag performance, optimization can be performed directly in the low-dimensional latent space rather than the high-dimensional space of the input data, which can accelerate computations of the gradients. Knowing this, we obtain a direction in the latent space corresponding to a decrease in the drag coefficient for a given geometry by computing the gradient of drag at a given latent space point. By modifying the latent space coordinate in the direction of reduced drag and observing changes in the decoded geometry, we can identify regions of the vehicle geometry to modify in order to reduce the drag coefficient. The produced car geometries from this iterative descent process can then be used to generate baseline designs for further validation.

However, it is important that any modified geometries produced during this optimization process should not be “extrapolation” cases. That is to say, we must ensure that the geometry produced by our optimization process still resembles a physically realizable car design. If the optimized design is too dissimilar to any cases in training, this may also put the reliability of the estimated drag into question. Since the distance in the latent space can be considered a

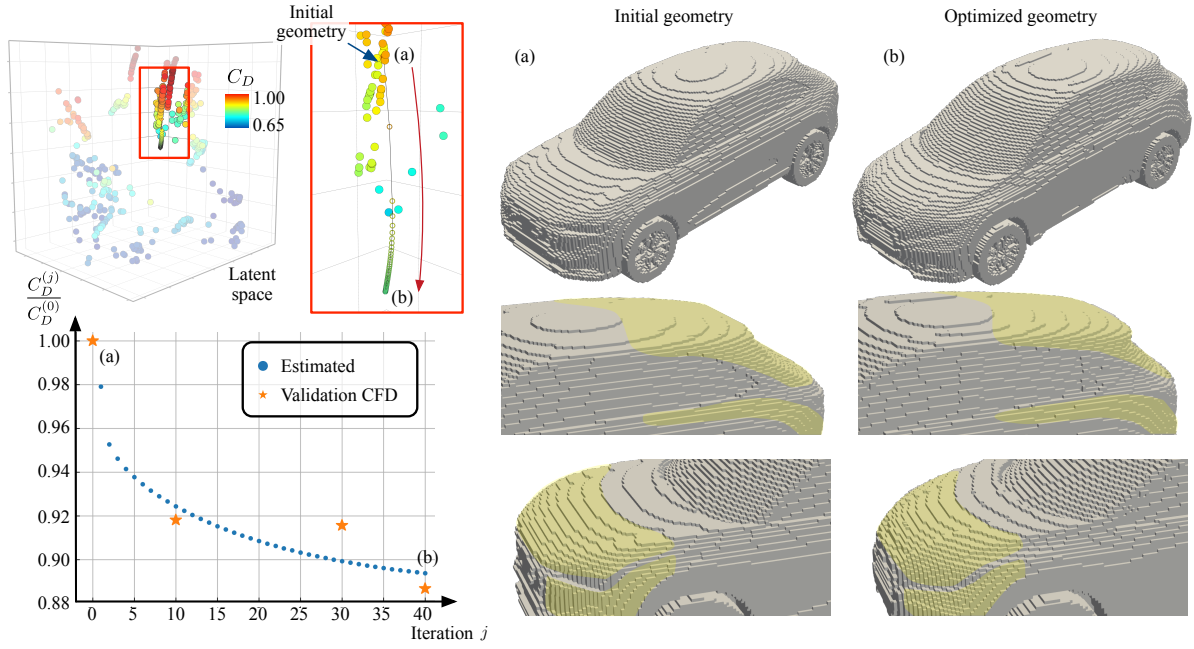


Figure 2.5: Data-driven vehicle shape optimization in the discovered latent space colored by the normalized drag coefficient. The latent optimization path with the corresponding normalized drag coefficient are shown. Orange points depict drag values obtained from validation computational fluid dynamics (CFD) simulation. (a) The given initial car and (b) optimized geometry are also depicted. The regions highlighted in yellow highlight some features that have been substantially modified.

measure of similarity, the present data-driven optimization considers a constraint based on the latent space distance between the optimized design and the training data. We consider a soft distance constraint that penalizes the optimization from moving too far from the training data. This distance constraint adds a penalty term to the cost function which grows as the distance from the training data increases. Further details of optimization formulation and the constraints are given in the Methods section.

We demonstrate our data-driven vehicle shape optimization on the discovered manifold in Fig. 2.5. We present an example case of geometry optimization starting from a high-drag SUV case (initial normalized drag coefficient,  $C_D \approx 0.86$ ). The initial geometry and the geometry after performing the shape optimization with the soft constraint are also visualized in Fig. 2.5. The model estimates a 11% reduction of the normalized drag ( $C_D \approx 0.77$ ) for the modified geometry. Along the optimization trajectory, we decode geometries and validate the drag estimate with LES performed on the smoothed voxel data. As seen in Fig. 2.5 we observe agreement between the trend of the estimated drag and the value corresponding to the CFD simulation for a few of the sampled validation cases. The validation cases demonstrate that the model has learned a trend between geometric features and aerodynamic performance rather than just memorizing the training data.

In Fig. 2.6, we show the pressure fields around the validation geometry. In the optimized geometry, we observe that the rear geometry is modified. The slope of the roof leading to the rear spoiler is lowered and the end is shifted backward. This provides a boattail-like effect as this reduces the pressure gradient where the flow separates (at the spoiler), which in turn reduces the pressure drag. Additionally, there is a smaller observed pressure gradient around the C-pillar, and we note that the  $C_{p,t} = 0$  isocontour appears elongated and tapered, indicating a longer region of recirculating flow behind the vehicle. The edges of the front geometry are smoothed, which reduces the size of the front face that the high-pressure zone (red) acts on, in addition to allowing the flow to smoothly flow around the front geometry.

Further insight into the unsteady wake dynamics can be obtained from the instantaneous

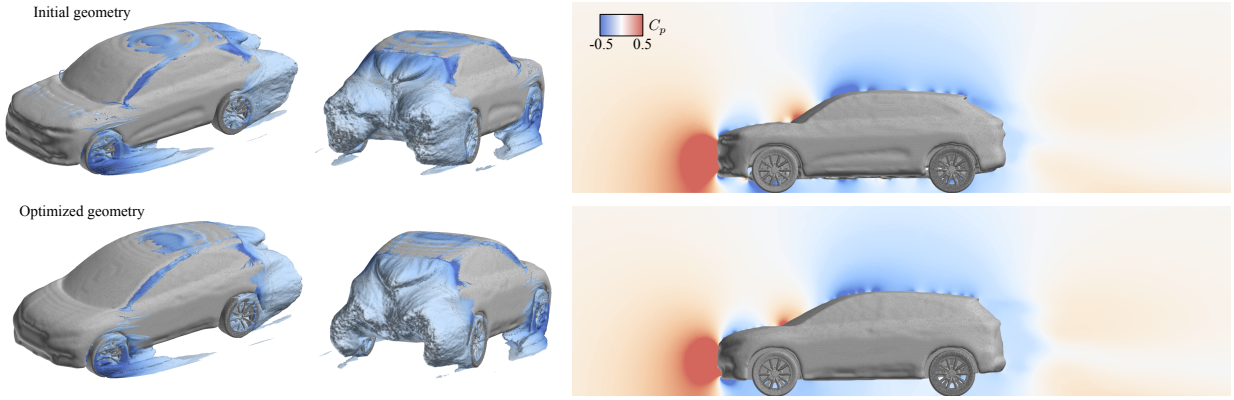
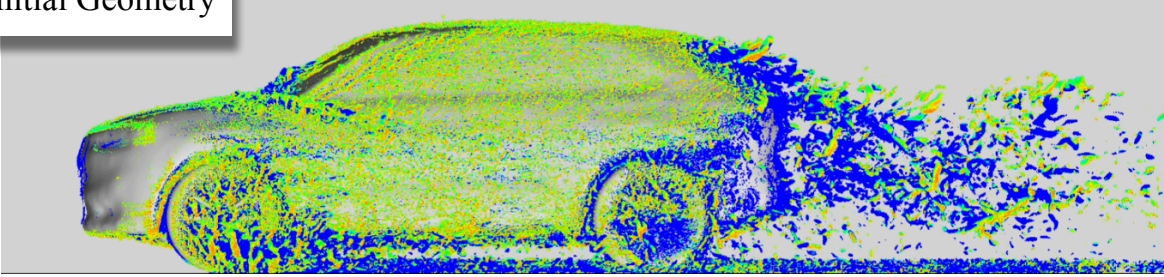


Figure 2.6: Validation large eddy simulation (LES) flow fields for smoothed voxel geometry for the initial high drag SUV case and the modified geometry obtained from the optimization procedure. (left) Zero total pressure ( $C_{p,t} = 0$ ) isocontour colored by pressure coefficient,  $C_p$ . (right) Midplane slice of pressure coefficient field,  $C_p$ . The pressure difference around the rear spoiler region is reduced.

Q-criterion isosurfaces shown in Fig. 2.7, which identify coherent vortical structures in the flow and are colored by streamwise velocity. In both geometries, a dense layer of small-scale turbulent structures is visible along the hood and roof, consistent with a turbulent boundary layer developing over the vehicle surface. We immediately note a significant mitigation of flow separation around the side edges of the rear geometry. The onset of the low-velocity (blue) region, approximately marking the boundary of the separated wake, appears immediately behind the rear wheel well on the initial geometry, but is shifted rearward on the optimized geometry towards the rear bumper.

Additionally, the large, low-velocity region immediately downstream of the rear geometry corresponds to the core of the separated wake, where large-scale, energetic vortical structures are shed from the trailing edges of the roof and C-pillar. Compared with the baseline geometry, the optimized geometry exhibits a sparser distribution of vortical structures in the wake, particularly immediately behind the rear window, consistent with the previously observed reduction in pressure drop. Together with the reduced extent of the  $C_{p,t} = 0$  isosurface noted in Fig. 2.6, this provides complementary evidence that the optimized geometry both mitigates flow separation and produces a less energetic, more gradually developing shear

Initial Geometry



Optimized Geometry

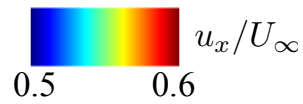
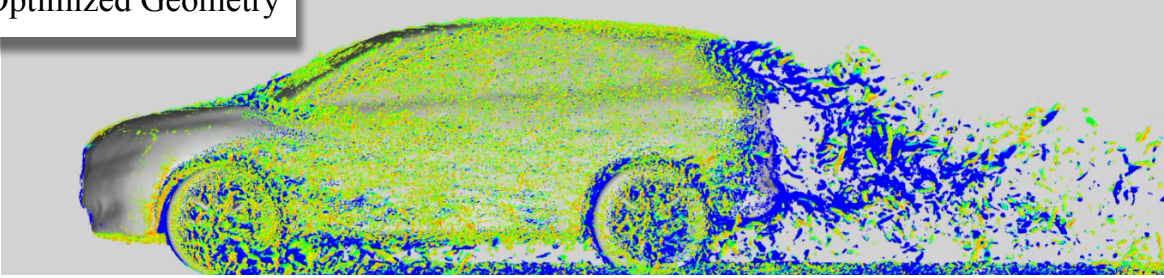


Figure 2.7: Instantaneous wake flow behind vehicle geometry simulated with LES visualized with  $Q$ -criterion ( $Q = 16 \times 10^4 \text{ s}^{-2}$ ) isosurface colored by streamwise velocity value.



layer off of the rear geometry.

## 2.4 Conclusions

In the current study, we perform a data-driven analysis of the aerodynamic performance of production vehicle geometries, which have been obtained with experimentally validated LES. We utilize a data-driven vehicle shape optimization approach leveraging an observable-augmented nonlinear autoencoder that identifies a low-dimensional latent space manifold that provides a compressed representation of automobile geometries. The use of the observable augmentation enforces that the learned low-dimensional space is also relevant to the estimation of the drag coefficient. For the present data set consisting of industry vehicle designs, the demonstrated approach effectively compresses voxelized geometries to just a three-dimensional latent space manifold while being able to sufficiently reconstruct the geometry and estimate the drag coefficient. This obtained low-dimensional space allows us to observe the relationship between different vehicle geometries and their respective aerodynamic performance, which we use to perform optimization of vehicle aerodynamics. We decode various geometries along the optimization trajectory and validate the trend of estimated drag with LES. Results of validation CFD for the decoded geometries show agreement with the trend of the estimated drag coefficients during the optimization process.

For the current autoencoder architecture, we employed a combination of PCA and a nonlinear multi-layer perceptron to improve the tractability and convergence of the neural network portion of the model by initially compressing the input voxel data with PCA. While we acknowledge the use of voxel geometry introduces cubic scaling of required storage memory with data resolution, we argue that the current approach of using voxel data with a PCA-assisted autoencoder offers a compromise between computational feasibility and geometric fidelity for use in an industrial setting, with the present model taking only around a single hour to train on a standard laptop (compared to approximately  $1.5 \times 10^4$  CPU hours

for evaluating a single LES case). The choice to use voxel data instead of the original mesh geometry is because models based on mesh data often are limited to deformation of existing control points and can produce distorted results for highly concave shapes or very sharp and thin features, especially when considering modifications like extrusion—limiting explorable shape modifications [UB18, RSM19, RSW21, DBW22]. Another difficulty in mesh- or point-based models for industrial use is that they typically require a fixed number of points at a relatively uniform density distribution, in addition to being computationally expensive during the graph construction process [EDA24]. It is for these reasons that we opt for a voxel-based representation. While the use of a PCA autoencoder approach makes the current approach computationally tractable with voxel data, it would be of interest to explore how alternative methods could be used to exploit local features to make complex geometries with minimal sacrifices of geometric information in a tractable manner.

The current study explores the potential avenue for a machine-learning analysis to expedite industrial aerodynamic design. We have demonstrated a proof-of-concept study of the applications of data-driven design to experimentally validated LES data of production vehicles for drag reduction. This analysis can also be extended to optimization of different design parameters, such as other aerodynamic coefficients or relevant metrics like mass or volume. Machine learning has the capability to streamline the production of more efficient vehicles, which is especially beneficial to the transition to electric transportation. The use of data-driven methods as a tool to direct the iterative design process exhibits promise for accelerating industrial design optimization.

## CHAPTER 3

# Learning Minimum-Distortion Optimal Transport Latent Embeddings: Theory and Applications to Separated Flow Analysis

In this chapter, we discuss the use of an optimal transport geometry to learn physically “interpretable” latent representations of flow fields. This chapter is adapted from work published in [TYT26].

### 3.1 Background and Motivation

Although autoencoders are a powerful method to compress and extract features from fluid flows, they come with a few drawbacks. While methods such as POD may perform worse in compressing highly complex nonlinear systems, the low-order coordinates and modes have intuitive physical explanations and have reproducible results [TBD17], which is not explicitly true for most machine-learning-based approaches. Fundamentally, autoencoders operate on the principle that high-dimensional datasets exist on low-dimensional manifolds [GT18]. The autoencoder learns a latent space that gives curvilinear coordinates that describe a parametric surface fit to the data [AHH18, MD22]. Often these models are prone to overfitting, requiring extensive regularization and hyperparameter tuning to improve their generalization capabilities, especially when the underlying manifold is not densely sampled [KW14, LYS22, KS23]. The problem of learning the manifold and latent space coordinates is inherently ill-posed, as any solution is only unique up to a diffeomor-

phism [SMH24]. The non-uniqueness of the learned latent space coordinates is a major contributing reason why obtaining a consistent interpretation of the latent coordinates is difficult outside of qualitative observations. This is especially an issue in the context of fluid mechanics, where it is desirable to use the learned latent space to perform downstream tasks that rely on geometric properties of the latent coordinates, such as dynamics modeling, flow field comparison, generative modeling, or interpolation.

Previous studies have sought to address some of these issues of interpretability. Studies by [MD22] and [KM24] interpret the latent space geometry of autoencoders using proper latent decomposition, which extends POD approaches to the non-Euclidean geometries learned by autoencoders. [FT23] utilizes an augmented autoencoder approach to analyze gust-airfoil interactions in which a model is simultaneously trained to reconstruct flow-field information and an estimate of the lift coefficient from the compressed latent representation. This encourages the model to learn a low-order representation where there is a qualitative relationship between the latent space coordinates and estimates of the lift. [SFS24] employed persistent homology, a method from topological data analysis, to design an autoencoder that provides a simple representation of the topology of the state dynamics of large-amplitude gust encounters in the latent space representation.

In this study, we seek an alternative data-driven approach to imposing structure in the low-order coordinates for the analysis of fluid flows by incorporating information of pairwise distances or similarities into the learned latent coordinates. This is motivated by the fact that when evaluating the effect of actuation on the dynamics of a flow, it is often necessary to quantify a degree of similarity or dissimilarity between two distinct flow fields. This typically involves the use of metrics or distances. One example is the widely adopted  $L^2$  metric (or Euclidean,  $\ell^2$ , for finite-dimensional vectors). However, while simple to compute, in some scenarios the Euclidean distance may not be the most informative measurement of dissimilarity, as it ignores the underlying structural and geometric information, only facilitating pointwise comparisons of the flow field across corresponding spatial locations. This may be

misleading in some scenarios, such as in the case where we seek to compare fields in which structures have been displaced.

Here, we utilize optimal transport (OT), which offers an intuitive framework for quantifying dissimilarities between flow fields. The class of OT distances, which includes the Wasserstein distance or earth mover’s distance, has been studied extensively for various problems, including those in economics [Vas69, Kan60], generative modeling [ACB17], image science [RTG98, PC19], and partial differential equation theory [Vil09, Mai12b]. Rather than only considering pointwise differences, OT distances effectively quantify a cost of transporting some resource or measure from one distribution to another [Vil09]. This makes OT particularly well-suited for scenarios in which the comparison of the spatial arrangement of flow features — such as vortices, coherent structures, or transported quantities — is of interest. Additionally, OT distances have been demonstrated to be robust to noise and outliers in measurements [PC19].

To demonstrate the potential of OT applied to fluid flows, we consider the analysis of separated aerodynamic flows. Separated flow is a phenomenon accompanied by often undesirable behavior, such as lift reduction and drag increase [Lis83, MD03]. Accordingly, engineers have long sought to understand and mitigate or reduce separation in aerodynamic flows through active and passive flow control techniques [Pra25, Lac61, WLD98, SP99, GW00, JM09, KMM10, YT19].

Flow separation typically occurs due to the emergence of an adverse pressure gradient, in which the boundary layer is decelerated and detaches from the surface, forming a shear layer [Lac61, MD03, MLR13]. Separation can also appear in the presence of sharp gradients in the surface geometry [Lac61, HBA00] or when induced by nearby vortical structures [HP71]. In the case of laminar separation, the separated shear layer is unstable, resulting in flow unsteadiness and/or a transition to turbulent flow [DKM94]. For high Reynolds number flows, the separated shear layer becomes more receptive to instabilities and becomes highly unsteady, eventually fully transitioning to turbulence around

which the Reynolds number is on the order of  $10^4 - 10^5$ . In such scenarios, the Kelvin-Helmholtz instabilities in the shear layer lead to the roll-up of spanwise coherent vortical structures, which drive momentum-mixing between the free-stream and the reverse flow region [Tan64, YSK09, KMM10, YT19]. Unsteady mixing facilitates entrainment of high-momentum fluid from the free stream to enable the separated shear layer to reattach to the airfoil surface further downstream, provided the boundary layer is sufficiently energized [Lac61, Lis83, MD03, JFV23].

When the flow reattaches, the time-averaged flow exhibits a closed region of recirculating flow, forming a separation bubble. Depending on various factors, such as chord-based Reynolds number or the angle of attack, the characteristics of this separated flow can vary substantially. The size, shape, and position of the separation bubble directly influence the performance of the airfoil, including stall characteristics and lift-to-drag ratio [Tan64, Lis83, MD03, KSS25]. At Reynolds numbers on the order of  $10^5$ , we may observe a long separation bubble that occupies over 20-30% of the chord, resulting in noticeable changes in the pressure distribution over the suction side [Lis83], while at even higher Reynolds numbers, a short separation bubble may be observed. These shorter bubbles are generally characterized by a linear increase of lift with the angle of attack, with the bubble bursting at the onset of stall [Lis83, MD03]. However, a long separation bubble can also lead to stall without bursting if it spans a sufficient amount of the airfoil surface.

The unsteady separation bubble has been described as a self-excited flow structure maintained by a feedback loop that occurs due to interactions between the Kelvin-Helmholtz instability in the shear layer and the vortex shedding in the wake [KSM97, ZMR89, GW00, YSK09, KMM10]. Various studies have been performed to investigate how to leverage these instability mechanisms that contribute to the sustainment of the separation bubble. In particular, periodic forcing has been demonstrated to be an effective method of unsteady boundary layer control, as it can reach similar control authority as steady blowing or suction while requiring orders of magnitude smaller momentum input [Wyg97, GW00]. [AG02]

performed an experimental parametric study using surface-mounted fluidic actuators to observe the flow response over a range of forcing frequencies, finding that unsteady periodic actuation can support complete flow reattachment. [YT19] explored the use of resolvent analysis to guide the design of a periodic thermal flux actuator for the suppression of flow separation over a NACA 0012 airfoil. The resolvent formulation was used to gain insights into the effects of variation of the spatial and temporal actuation frequencies of a heat flux actuator on the separation dynamics.

In what follows, we analyze the effect of unsteady thermal actuation on separated flow over a NACA 0012 airfoil in a data-driven manner from the perspective of OT. We leverage an OT distance-based embedding learned by an autoencoder to analyze the controlled flows in a low-dimensional manner. We align pairwise Euclidean distances in the latent space with dissimilarities computed with OT, which attempts to associate the geometric structure of the latent space with the spatial distribution of fluid structures according to the OT distance. This restriction on the latent space coordinates allows relative distances in the latent space to encode information of similarity in terms of the spatial distribution of structures in the flow response. We find that the distribution of flow responses to periodic thermal actuation is reducible to an interpretable two-dimensional latent representation that encodes the behavior of the separation bubble in the time-averaged flows. This demonstrates the potential utility of OT in the analysis of fluid flows.

## 3.2 Optimal Transport for Comparison of Fluid Flows

In this section, we give a general description of the OT distance and its application to computing dissimilarities between the distribution of structures in flow fields. Since the formulation of OT is naturally expressed in the language of measure theory and probability, we briefly recall the basic concepts and notations needed for the present implementation in Appendix A. For a more comprehensive review, we refer the reader to standard texts such

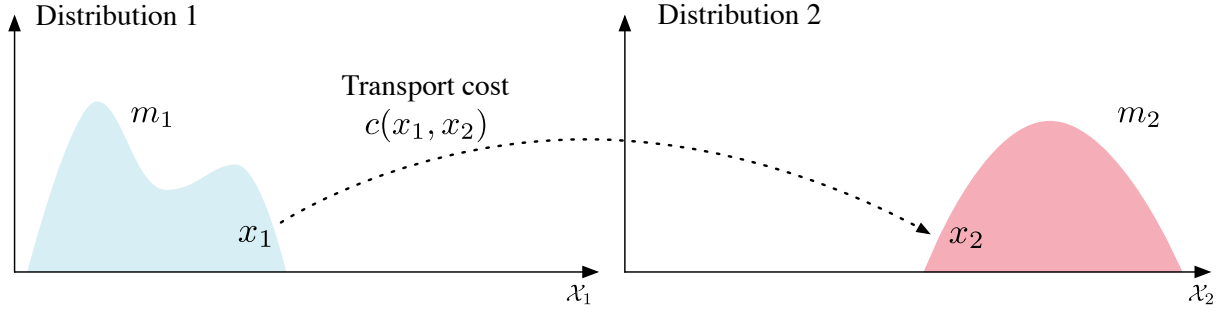


Figure 3.1: Pedagogical schematic of optimal transport between a supply and demand distribution representative of two different flow fields. The optimal transport distance is the minimum total cost to move the supply to the demand distribution.

as [Wil91, Vil09, Tao11].

### 3.2.1 Physical Intuition for Optimal Transport

OT distances provide a mathematical framework for comparing distributions of quantities while accounting for spatial structure. Originally developed to address problems of efficient resource allocation, the original optimal transport problem seeks the most cost-effective way to redistribute a given resource from one configuration to another [Vas69, Kan60]. The classical formulation considers a set of supply (or source) locations over which resources are distributed, and a set of demand (or target) locations, each requesting a specified amount of resources. The cost of transportation between locations depends on both the ground cost, in this case, the distance traveled, and the quantity of resources transported. The OT distance then corresponds to the minimum cost required to transport the resources from the supply distribution to the demand distribution. For these cases in which the total supply and demand are the same (balanced), the source and target distributions can be represented as probability measures, and OT can be used to define a metric between them. An illustration of OT for this pedagogical problem is shown in figure 3.1(a).

Following these examples, one can intuitively understand OT distances as a standard way to lift metrics (or distances) defined on some ground measurable space to a distance between



distributions of measures [Vil09, PC19, CPS18]. OT enables us to leverage the underlying geometric structure to compare distributions. In the context of fluid flows, the compared distributions may represent different flow fields, with each distribution describing the spatial variation of physical quantities such as density, velocity, or vorticity.

To build further physical intuition, consider Figure 3.2(a), which shows selected time steps of the solution to the advection–diffusion equation with a Gaussian initial condition. Figure 3.2(b) compares the distance from the initial condition measured using the  $L^2(\mathcal{X})$  metric and the unbalanced OT distance  $d_{\text{UOT}}$ . Note that, for this particular example, one could also use the balanced OT distance since the transport process is conservative. The  $(L^2(\mathcal{X}))$  distance between the solution at time (t) and the initial condition is

$$|f(\cdot, t) - f(\cdot, 0)|_{L^2(\mathcal{X})} = \left( \int_{\mathcal{X}} [f(x, t) - f(x, 0)]^2, dx \right)^{1/2}, \quad (3.1)$$

where  $(\mathcal{X} \subseteq \mathbb{R}^d)$  denotes the spatial domain. This metric captures pointwise dissimilarities between the two functions. As a result, it increases rapidly even for small translations but then saturates, classifying most subsequent time steps as equally dissimilar once there is little overlap with the initial condition. In contrast, the unbalanced OT distance continues to correlate with the displacement of the Gaussian pulse even when its support no longer overlaps with the initial condition. Consequently, the unbalanced OT distance provides a characterization of dissimilarity that better aligns with our intuitive understanding of how the distribution evolves over time.

### 3.2.2 Balanced Optimal Transport Distances

To define the OT distance between two measures, we start with spaces corresponding to the spatial domain of the two distributions, which are depicted in figure 3.1(a) by  $\mathcal{X}_1$  and  $\mathcal{X}_2$ . We then choose a lower semi-continuous cost function that maps between the product space of  $\mathcal{X}_1$  and  $\mathcal{X}_2$  to the non-negative extended reals  $C : \mathcal{X}_1 \times \mathcal{X}_2 \rightarrow \mathbb{R}_{\geq 0} \cup \{+\infty\}$ . This cost

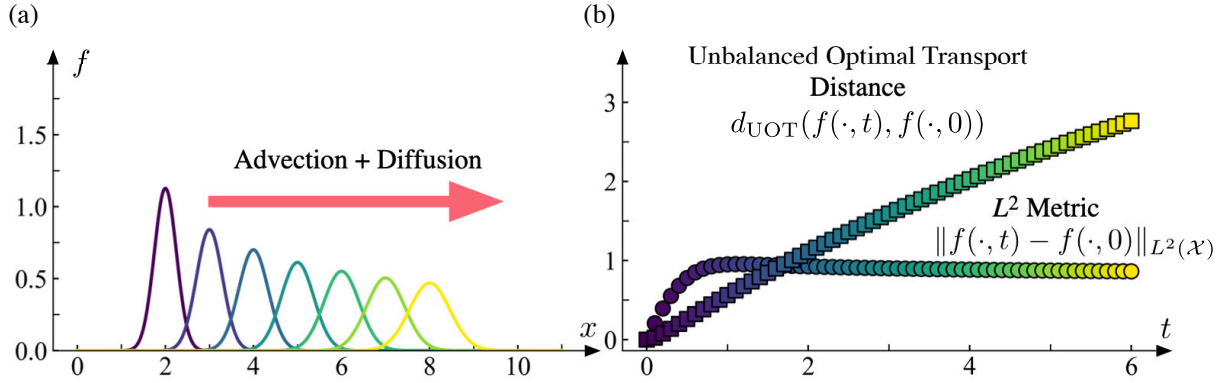


Figure 3.2: Comparison of unbalanced optimal transport distance  $d_{\text{UOT}}$  with the  $L^2$  metric. (a) Gaussian pulse undergoing advection and diffusion. (b) Comparison of distances between the evolving pulse and the initial condition.

function  $C$  describes the cost of transport from point  $x_1 \in \mathcal{X}_1$  to another point  $x_2 \in \mathcal{X}_2$ . For this work, we choose  $C$  to be the Euclidean distance between points in  $\mathcal{X}_1$  and  $\mathcal{X}_2$ . However, the relevant choice of the cost function is problem-dependent. We note that the inclusion of  $+\infty$  in the range is to account for possible obstacles that may be present in the flow domain (e.g., a solid submerged body). Given  $\mathcal{M}_+(\cdot)$ , the set of all nonnegative measures over some measurable space, we have that for positive measures  $m_1 \in \mathcal{M}_+(\mathcal{X}_1)$  and  $m_2 \in \mathcal{M}_+(\mathcal{X}_2)$ , the optimal transport (OT) cost is defined as:

$$\mathcal{J}(\Gamma) = \int_{\mathcal{X}_1 \times \mathcal{X}_2} C(x_1, x_2) d\Gamma(x_1, x_2). \quad (3.2)$$

With this cost, the balanced OT distance is computed with the following constrained minimization:

$$d_{\text{OT}}(m_1, m_2) = \inf_{\Gamma \in \mathcal{M}_+(\mathcal{X}_1 \times \mathcal{X}_2)} \left\{ \mathcal{J}(\Gamma) : P_{\#}^{\mathcal{X}_1} \Gamma = m_1, P_{\#}^{\mathcal{X}_2} \Gamma = m_2 \right\}. \quad (3.3)$$

Here  $\Gamma$ , called the coupling (or transport plan), is a nonnegative measure over the product space  $\mathcal{X}_1 \times \mathcal{X}_2$  and describes the transport of resources between the distributions of  $m_1$  and  $m_2$ . The joint distribution of  $\Gamma$  over  $\mathcal{X}_1 \times \mathcal{X}_2$  quantifies how much resource is moved between the points  $x_1$  and  $x_2$ . The measures  $P_{\#}^{\mathcal{X}_1} \Gamma$  and  $P_{\#}^{\mathcal{X}_2} \Gamma$  denote the first and second marginals

of  $\Gamma$ , respectively. These can be understood identically to the familiar notion of marginal probability, given by  $P_{\#}^{\mathcal{X}_1}\Gamma = \int_{x_2 \in \mathcal{X}_2} d\Gamma(\cdot, x_2)$  and  $P_{\#}^{\mathcal{X}_2}\Gamma = \int_{x_1 \in \mathcal{X}_1} d\Gamma(x_1, \cdot)$ . The constraints enforce that the total measure must be the same for both distributions.

### 3.2.3 Definition of Csiszár Divergence Functionals

Naturally, the distributions of measures in different fluid flow fields need not integrate to the same amount. However, the traditional balanced OT formulation requires both distributions to have the same total measure. To account for this imbalance, we utilize a modification to the OT cost using Csiszár divergence functionals. These divergences are functionals that compare two distributions of measures in a pointwise manner (like the Kullback-Leibler divergence) and can be formally built with entropy functions in the context of probability and information theory.

Here we provide a high-level summary of Csiszár divergence functionals as described in [CPS18], [LMS18], and [Ngu25]. To compute a Csiszár divergence functional, one must start with an entropy function. An admissible entropy function is a function  $\varphi : \mathbb{R} \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$  which is convex, lower semicontinuous,  $\text{dom}(\varphi) \subset [0, \infty)$ ,  $\text{dom}(\varphi) \cap (0, \infty) \neq \emptyset$ , and  $\varphi(0) = 1$ .<sup>1</sup>

From the entropy function, the Csiszár divergence functional can be defined for two measures  $m_1, m_2 \in \mathcal{M}(\mathcal{X})$  as:

$$\mathcal{D}_{\varphi}(m_1 \mid m_2) = \int_{\mathcal{X}} \varphi\left(\frac{dm_1}{dm_2}\right) dm_2 + \varphi'_{\infty} m_1^{\perp}(\mathcal{X}), \quad (3.4)$$

when  $m_1$  and  $m_2$  are nonnegative ( $\infty$  otherwise). This expression relies on the Lebesgue decomposition, which decomposes  $m_1$  into a part absolutely continuous with respect to  $m_2$ ,

---

<sup>1</sup>Here,  $\text{dom}(\varphi)$  denotes the *effective domain* of  $\varphi$  in the sense of convex analysis, namely

$$\text{dom}(\varphi) = \{x \in \mathbb{R} : \varphi(x) < \infty\}.$$

denoted  $m_1^{\text{ac}}$ , and a singular part  $m_1^\perp$  such that

$$m_1 = m_1^{\text{ac}} + m_1^\perp, \quad m_1^{\text{ac}} \ll m_2, \quad m_1^\perp \perp m_2.$$

The Radon–Nikodym derivative  $dm_1/dm_2$  is defined for the absolutely continuous part  $m_1^{\text{ac}}$ . The recession constant  $\varphi'_\infty$  is given by:

$$\varphi'_\infty = \lim_{s \rightarrow \infty} \frac{\varphi(s)}{s} \in \mathbb{R} \cup \{\infty\}. \quad (3.5)$$

In the present work, we compute the divergence using the Kullback-Leibler (KL) entropy  $\varphi_{\text{KL}}$  given by:

$$\varphi_{\text{KL}}(s) = \begin{cases} s \log(s) - s + 1 & \text{for } s > 0, \\ 1 & \text{for } s = 0, \\ +\infty & \text{otherwise.} \end{cases} \quad (3.6)$$

Note that when the total measures are equal (e.g., probability measures) and  $m_1^\perp(\mathcal{X}) = 0$ ,  $\mathcal{D}_\varphi(m_1 \mid m_2)$  with  $\varphi_{\text{KL}}$  reduces to the familiar KL divergence from statistics.

### 3.2.4 Unbalanced Optimal Transport Distances

In the case of unbalanced measures, the total cost of  $\mathcal{J}(\Gamma)$  is modified to include an extra divergence functional term, replacing the equality constraint on the marginals in the optimization. In other words, instead of strictly enforcing equality of the total measures, the divergences only softly penalize imbalance between the distributions, giving us an unbalanced OT distance.

Thus, to allow for a possible imbalance between the two flow fields, we additionally consider  $\mathcal{D}_{\varphi_1}$  and  $\mathcal{D}_{\varphi_2}$ , which are Csiszár divergences over  $\mathcal{X}_1$  and  $\mathcal{X}_2$ . Given  $\mathcal{M}_+(\cdot)$ , the set of all nonnegative measures over some measurable space, we have that for positive measures  $m_1 \in \mathcal{M}_+(\mathcal{X}_1)$  and  $m_2 \in \mathcal{M}_+(\mathcal{X}_2)$ , the unbalanced optimal transport (UOT) cost is defined

as:

$$\mathcal{J}(\Gamma) = \int_{\mathcal{X}_1 \times \mathcal{X}_2} C(x_1, x_2) d\Gamma(x_1, x_2) + \rho \left[ \mathcal{D}_{\varphi_1}(P_{\#}^{\mathcal{X}_1} \Gamma | m_1) + \mathcal{D}_{\varphi_2}(P_{\#}^{\mathcal{X}_2} \Gamma | m_2) \right]. \quad (3.7)$$

Here, the term  $\rho > 0$  is described as a characteristic radius of transport which balances the contribution of the transport and divergence terms [SFV19]. The UOT distance is then defined as the infimum of this cost over  $\Gamma \in \mathcal{M}_+(\mathcal{X}_1 \times \mathcal{X}_2)$ :

$$d_{\text{UOT}}(m_1, m_2) = \inf_{\Gamma \in \mathcal{M}_+(\mathcal{X}_1 \times \mathcal{X}_2)} \mathcal{J}(\Gamma). \quad (3.8)$$

The first integral term describes the cost associated with a change in the spatial distribution. For example, this term would capture if a flow structure, such as a vortex or a shear layer, is at a different position in two flow fields. The second term that includes the divergences penalizes how different the marginals of  $\Gamma$  are from the input measures  $m_1$  and  $m_2$ , which allows for imbalance in the total amount of the two input measures.

To compute the divergence functionals, we use the Kullback-Leibler entropy and we set the characteristic transport radius  $\rho = 1$  [Vil09, PC19, CPS18].

### 3.2.5 Numerical Computation of the Unbalanced Optimal Transport Based Flow Field Dissimilarity

As mentioned previously, the unbalanced optimal transport distance  $d_{\text{UOT}}$  is defined as the infimum of the optimal transport cost  $\mathcal{J}$  over all possible transport plans  $\Gamma \in \mathcal{M}_+(\mathcal{X}_1 \times \mathcal{X}_2)$ :

$$\mathcal{J}(\Gamma) = \int_{\mathcal{X}_1 \times \mathcal{X}_2} C(x_1, x_2) d\Gamma(x_1, x_2) + \rho \left[ \mathcal{D}_{\varphi_1}(P_{\#}^{\mathcal{X}_1} \Gamma | m_1) + \mathcal{D}_{\varphi_2}(P_{\#}^{\mathcal{X}_2} \Gamma | m_2) \right], \quad (3.9)$$

$$d_{\text{UOT}}(m_1, m_2) = \inf_{\Gamma \in \mathcal{M}_+(\mathcal{X}_1 \times \mathcal{X}_2)} \mathcal{J}(\Gamma). \quad (3.10)$$

However, this optimization problem normally exhibits poor tractability. To compute the OT distance for histograms of dimension  $N_h$ , the computational cost scales at least in  $\mathcal{O}(N_h^3 \log(N_h))$  in the general case where no restrictions are placed upon the metric used to define the OT cost [PW09, Cut13]. Consequently, rather than directly solving this constrained optimization problem, it is common to replace the non-negativity constraint on  $\Gamma$  with an entropic regularization term to obtain an approximate solution. The entropic regularization changes the original linear programming problem to a strictly convex problem, which can be solved more efficiently using the Sinkhorn-Knopp matrix scaling algorithm [Cut13, CPS18].

To define the entropically regularized problem, we consider the entropy of the transport plan. The negative entropy is given by

$$\mathcal{H}(\Gamma) = \int_{\mathcal{X}_1 \times \mathcal{X}_2} r(\log r - 1) d(x_1, x_2), \quad (3.11)$$

where the coupling  $\Gamma$  is assumed to admit a density  $r$  with respect to the product measure on  $\mathcal{X}_1 \times \mathcal{X}_2$ .

The entropically regularized unbalanced OT distance between measures  $m_1$  and  $m_2$  is the following convex optimization problem:

$$d_{\text{UOT}}^\varepsilon(m_1, m_2) = \min_{\Gamma \in \mathcal{M}(\mathcal{X}_1 \times \mathcal{X}_2)} (\mathcal{J}(\Gamma) + \varepsilon \mathcal{H}(\Gamma)), \quad (3.12)$$

where  $\varepsilon > 0$  is a small regularization constant. Here, the inclusion of the entropic regularization term  $\mathcal{H}(\Gamma)$  follows the maximum-entropy principle and serves as a relaxation of the nonnegativity constraint on  $\Gamma$  [Cut13, CPS18]. For this, we utilize the Python Optimal Transport library, which efficiently implements the Sinkhorn-Knopp matrix scaling algorithm [FCG21].

The Sinkhorn-Knopp algorithm computes the entropically regularized transport plan

through iterative matrix scaling. After discretizing the measures, the kernel is defined by

$$K_{ij} = \exp\left(-\frac{C_{ij}}{\varepsilon}\right),$$

where  $C_{ij}$  denotes the pairwise cost between source and target points. In the balanced case, for  $\varepsilon > 0$  the numerically computed entropically regularized optimal transport plan takes the form

$$\Gamma = \text{diag}(u) K \text{diag}(v), \quad (3.13)$$

where the scaling vectors  $u$  and  $v$  are updated alternately to satisfy the marginal constraints. In the unbalanced setting, these updates are modified to account for the relaxed marginals [CPS18]. The iterations continue until convergence, yielding the entropically regularized transport plan from which the transport cost is evaluated. The algorithm has a computational complexity of  $\mathcal{O}(N_h^2)$  per iteration and is significantly more efficient than solving the corresponding linear program for large-scale problems.

It is important to note that introducing entropic regularization induces a non-zero bias even when the input measures coincide [GCB19, SFV19]. To correct for this effect, the following normalization is applied, leading to the so-called Sinkhorn divergence:

$$\begin{aligned} S^\varepsilon(m_1, m_2) &= d_{\text{UOT}}^\varepsilon(m_1, m_2) - \frac{1}{2} \left[ d_{\text{UOT}}^\varepsilon(m_1, m_1) + d_{\text{UOT}}^\varepsilon(m_2, m_2) \right] \\ &\quad + \frac{\varepsilon}{2} \left( \int_{\mathcal{X}_1} dm_1 - \int_{\mathcal{X}_2} dm_2 \right)^2. \end{aligned} \quad (3.14)$$

We consider the cost  $C(x, y) = \|x - y\|_2$  on a compact domain  $\mathcal{X} \subset \mathbb{R}^d$ . In this setting, we find that  $C$  is symmetric, bounded, and Lipschitz continuous with respect to the  $L^\infty$  norm (essential supremum norm) on  $\mathcal{X} \times \mathcal{X}$ . Moreover, the associated kernel  $k_\varepsilon = e^{(-C/\varepsilon)}$  is positive definite, continuous, and universal on  $\mathcal{X} \times \mathcal{X}$ . Thus it follows from Theorem 5 in [SFV19] that for any  $\varepsilon > 0$ , the Sinkhorn divergence for our choice of cost function is positive definite, symmetric, and convex in each argument, and it satisfies  $S^\varepsilon(m_1, m_2) = 0$  if

and only if  $m_1 = m_2$ . A sketch of the proof of this is given in Appendix C

To compute a flow field dissimilarity for signed quantities (such as vorticity), we represent each scalar field as a signed measure and compare its positive and negative parts separately. Let  $\mathcal{X} \subset \mathbb{R}^d$  be a compact measurable domain equipped with the Lebesgue measure, and let  $V : \mathcal{X} \rightarrow \mathbb{R}$  be a measurable function satisfying  $V \in L^1(\mathcal{X})$ . We define the signed measure  $m$  by

$$dm = V(x) dx.$$

This definition is well-posed since  $V \in L^1(\mathcal{X})$  implies that  $m$  has finite total variation. Consequently,  $m$  admits a unique Jordan decomposition

$$m = m^+ - m^-,$$

where  $m^+$  and  $m^-$  are mutually singular finite positive measures.

By the Radon–Nikodym theorem, both measures are absolutely continuous with respect to the Lebesgue measure, and admit densities given by the positive and negative parts of  $V$ :

$$dm^+ = V^+(x) dx, \tag{3.15}$$

$$dm^- = V^-(x) dx, \tag{3.16}$$

where  $V^+(x) = \max\{V(x), 0\}$  and  $V^-(x) = \max\{-V(x), 0\}$ . We interpret  $m^+$  and  $m^-$  as the distributions of positive and negative contributions of the flow field over  $\mathcal{X}$ .

Given two fields  $V_1, V_2 \in L^1(\mathcal{X})$ , we obtain corresponding decompositions  $m_1^\pm$  and  $m_2^\pm$ . We define a field dissimilarity by comparing positive and negative parts separately using the Sinkhorn divergence:

$$d_{\text{field}}(V_1, V_2) = S^\varepsilon(m_1^+, m_2^+) + S^\varepsilon(m_1^-, m_2^-).$$



This construction is inspired by [Mai12b, Mai12a].

Now consider two sets of fields consisting of  $N$  variables,  $\mathbf{V}_1 = \{V_1^{(i)}\}_{i=1}^N$  and  $\mathbf{V}_2 = \{V_2^{(i)}\}_{i=1}^N$ . We compute the OT-based dissimilarity for each variable and define an overall field-level discrepancy via

$$\tilde{d}_{\text{field}}(\mathbf{V}_1, \mathbf{V}_2) = \bigoplus_{i=1}^N d_{\text{field}}(V_1^{(i)}, V_2^{(i)}), \quad (3.17)$$

where  $\bigoplus$  denotes a permutation-invariant aggregation operator, such as a (possibly weighted) sum, a mean, or an  $\ell^2$  norm over the vector of componentwise dissimilarities. In this work, we choose the  $\ell^2$  norm applied to the vector of field dissimilarities across variables.

### 3.3 Separated Flow Configuration and Control Setup

As a demonstrative example, we consider the effects of active flow control on separated flows over a NACA 0012 airfoil at angles of attack  $\alpha \in \{6^\circ, 9^\circ\}$  with chord-based Reynolds number  $Re_{L_c} \equiv u_\infty L_c / \nu_\infty = 23,000$  and free-stream Mach number  $M_\infty \equiv u_\infty / a_\infty = 0.3$ . Here  $u_\infty$  is the free-stream velocity,  $L_c$  is the chord length,  $a_\infty$  is the free-stream speed of sound, and  $\nu_\infty$  is the kinematic viscosity.

The flow fields were obtained via large eddy simulation (LES) using the *CharLES* finite-volume compressible flow solver [KHN11, BHN17] with the Vreman subgrid model [Vre04]. We used a C-grid mesh following the setup in [YT19], which has been examined for convergence in the flow field and aerodynamic forces with refinements in the near-field. The computational domain is within  $(x/L_c, y/L_c, z/L_c) \in [-19, 26] \times [-20, 20] \times [-0.1, 0.1]$  with the leading edge of the airfoil being placed at the origin. The solution was computed using a constant timestep of  $\Delta t u_\infty / L_c = 4.14 \times 10^{-5}$  corresponding to a maximum Courant-Friedrichs-Lewy (CFL) number of 0.84. The statistics of the aerodynamic forces were computed using over 80 convective time units. Additional details of the computational setup

and grid are provided in [YMT17a] and [YT19].

The specific heat ratio was specified as  $\gamma = 1.4$  with a Prandtl number  $Pr \equiv \nu/\kappa = 0.7$ , where  $\kappa$  is the thermal diffusivity. The dynamic viscosity was specified as a function of temperature in the form of a power law, given by  $\mu(T) = \mu_\infty(T/T_\infty)^{0.76}$  for  $T/T_\infty \in [0.5, 1.7]$  with  $\mu_\infty$  and  $T_\infty$  denoting the free-stream dynamic viscosity and temperature, respectively [GAS09]. For all of the flows in this study, we observed that the maximum temperature fluctuation in the flow is within 42% of  $T_\infty$ , falling within the allowable range of the viscosity model [YT19].

For the far-field boundary, a Dirichlet boundary condition was specified as  $(\rho, u_x, u_y, u_z, T) = (\rho_\infty, u_\infty, 0, 0, T_\infty)$ . A sponge layer [Fre97] was placed along the outlet boundary with a target state being the running-averaged flow over 10 acoustic time units. Over the airfoil surface, a no-slip boundary condition was prescribed. The airfoil surface was also specified to be adiabatic except where the periodic heat-flux actuator was placed at the leading edge.

The thermal actuation setup employed in this study followed the methodology outlined in [YT19], where a thermal actuator was placed across the span near the leading edge with prescribed frequency and spanwise profile. The actuator was implemented in the energy equation as a nonhomogeneous, time-dependent Neumann boundary condition. The actuator model was expressed using a Hann window with compact spatial support given by

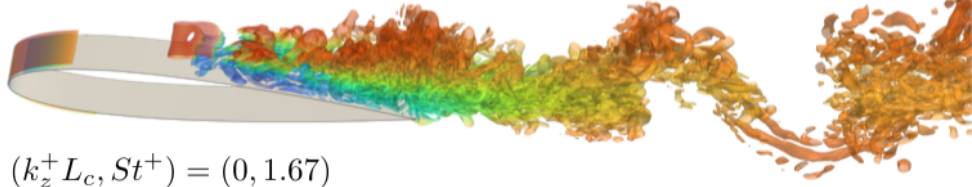
$$\phi^+(\omega^+, k_z^+) = \frac{1}{4}\hat{\phi}\sin(\omega^+t) [1 + \cos(k_z^+z)] \left\{ 1 + \cos\left[\frac{2\pi}{\sigma_a}(x - x_a)\right] \right\}, \quad (3.18)$$

where  $(x - x_a)/\sigma_a \in [-0.5, 0.5]$ . The actuator was centered at  $x_a/L_c = 0.03$  on the suction surface with width  $\sigma_a/L_c = 0.04$ . The amplitude  $\hat{\phi}$  was fixed according to the normalized total actuation power,

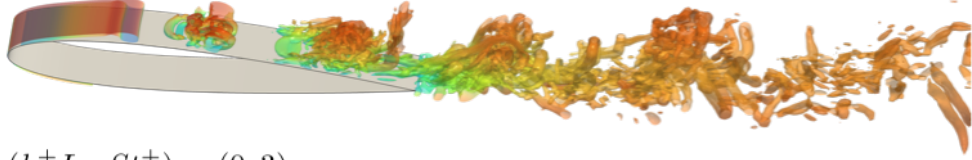
$$E^+ = \frac{\frac{1}{4}\hat{\phi}\sigma_a}{\frac{1}{2}\rho_\infty v_\infty^3 (L_c \sin \alpha)} = 0.0902, \quad (3.19)$$

which is consistent with the setup of previous studies [CEW10, SAK12, YT19]. The thermal input from the actuator manifests in an oscillatory surface vorticity flux and baroclinic

Baseline



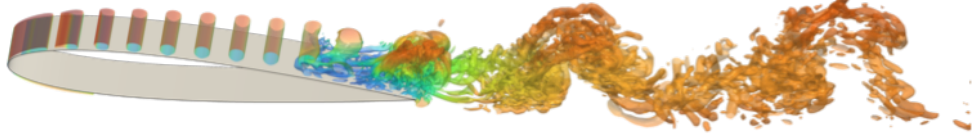
$$(k_z^+ L_c, St^+) = (0, 1.67)$$



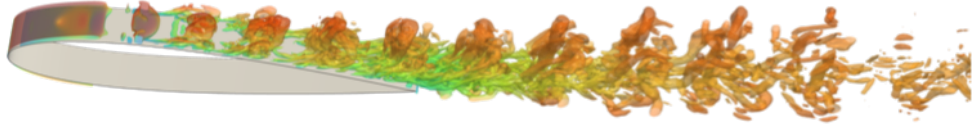
$$(k_z^+ L_c, St^+) = (0, 3)$$



$$(k_z^+ L_c, St^+) = (0, 7.33)$$



$$(k_z^+ L_c, St^+) = (10\pi, 4)$$



$$(k_z^+ L_c, St^+) = (10\pi, 8.66)$$

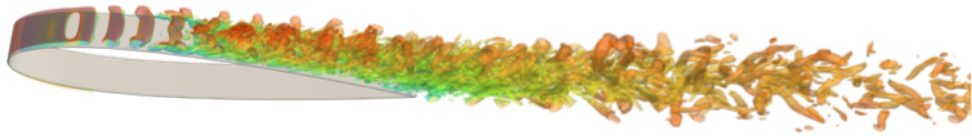


Figure 3.3: Examples of baseline flow and various responses of a separated wake past a NACA 0012 airfoil at  $\alpha = 6^\circ$  to a heat flux actuator input at the leading edge [YT19]. The  $QL_c^2/u_\infty^2 = 50$  isosurface colored by the normalized streamwise velocity is shown.

torque [YMT17b, YT19].

From this actuator, we have a two-dimensional parameter space of control inputs  $(k_z^+, \omega^+) \in 10\pi\mathbb{Z} \times \mathbb{R}_{\geq 0}$ . The control parameter pairs are characterized by a dimensionless wavenumber  $k_z^+ L_c$  and chord-based actuation Strouhal number  $St^+ = \omega^+ L_c / (2\pi u_\infty)$ . For each angle of attack, we considered actuation wavenumbers of  $k_z^+ L_c \in \{0, 10\pi, 20\pi, 40\pi\}$  with around 30 cases varying  $St^+ \in [0, 18]$  for a total of around 120 cases per angle of attack.

An assortment of wake responses resulting from changes in the input parameter  $St^+$  are shown in figure 3.3. Despite varying only two parameters in the shown cases, the resulting flow fields exhibit remarkably complex and diverse behavior. These differences manifest in several key aspects of the wake dynamics. For example, the extent and nature of flow separation deviate greatly, with the size and position of the separation bubble differing significantly across different controlled cases. Additionally, the degree of laminarization and turbulent mixing over the airfoil surface and in the downstream wake also changes noticeably between cases, with some cases exhibiting partial or global laminarization. The wide array of flow field responses motivates the use of a data-driven analysis to extract relevant features in the flow response that relate to the control performance.

### 3.4 OT Based Autoencoder for Learning Minimum-Distortion Embeddings

We seek to utilize an autoencoder to learn a low-dimensional representation of fluid flows while preserving dissimilarity between inputs in the learned latent representation. In other words, the goal is to learn not just a compressed representation of our flow fields, but to learn a representation with some geometric structure informed by OT that allows the latent coordinates to be interpreted as providing relative similarities between flow fields.

Consider some instance of discretized flowfield data  $\mathbf{q} \in \mathbb{R}^{N \times N_s}$  where  $N$  refers to the number of variables and  $N_s$  refers to the size of the spatial discretization. A typical au-

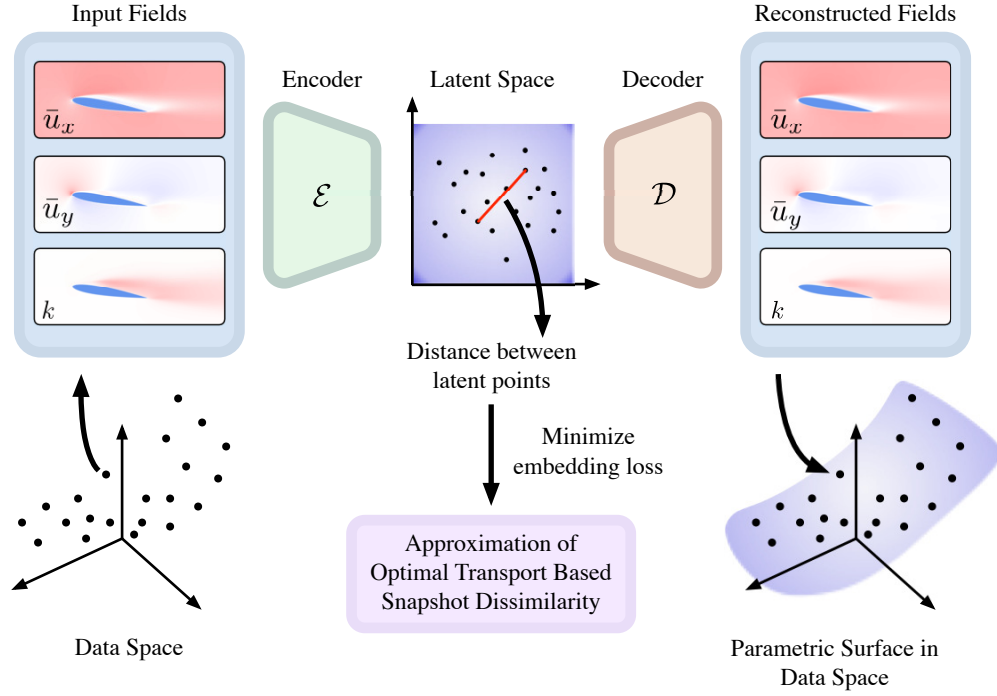


Figure 3.4: Illustration of the augmented autoencoder problem setup. During training, the model learns to associate Euclidean distances in the latent space (red line) with flow field dissimilarities computed with optimal transport.

toencoder consists of an encoder-decoder pair  $(\mathcal{E}, \mathcal{D})$  which can be constructed via neural networks [HS06]. The encoder  $\mathcal{E} : \mathbb{R}^{N \times N_s} \rightarrow \mathbb{R}^m$  maps the input from a physical space to a latent space representation, with coordinates  $\boldsymbol{\xi} \in \mathbb{R}^m$ . The decoder  $\mathcal{D} : \mathbb{R}^m \rightarrow \mathbb{R}^{N \times N_s}$  takes a latent space coordinate and produces a reconstruction of the input  $\tilde{\mathbf{q}} = \mathcal{D} \circ \mathcal{E}(\mathbf{q}) \in \mathbb{R}^{N \times N_s}$ . In this study, the encoder network consists of convolutional layers followed by dense layers. Each layer is followed by batch normalization and a ReLU activation function. The decoder is constructed in a reverse fashion with dense layers followed by convolutional layers. Residual skip connections are placed around each convolutional block to stabilize training and reduce overfitting [HZR16, IS15]. The parameters for the autoencoder architecture are given in Table 3.1. In the present work, the number of filters is increased with depth [LBB02]. All variables have been min-max normalized prior to training.

Here, we consider the time- and spanwise-averaged streamwise velocity, vertical velocity, and the turbulent kinetic energy,  $k$ , as the input and output, i.e.,  $\mathbf{q} = (\bar{u}_x, \bar{u}_y, k)$ . We note that since  $\bar{\omega}_z = \partial \bar{u}_y / \partial x - \partial \bar{u}_x / \partial y$ , we can also reconstruct  $\bar{\omega}_z$  using the reconstructed velocities for assessment of the reconstruction performance. For the autoencoder input and output as well as the flow field OT-dissimilarity computation, we consider a two-dimensional spatial region around the airfoil given by  $\mathcal{X} = [-0.5L_c, 2L_c] \times [-0.5L_c, 0.5L_c]$  with  $n_x = 250$  and  $n_y = 100$ .

The model's weights  $\theta$  are optimized via the following objective function:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \quad (\mathcal{L}_1 + \lambda \mathcal{L}_2) \quad (3.20)$$

$$\mathcal{L}_1 = \frac{1}{MN N_s} \sum_{i=1}^M \|\mathbf{q}_i - \mathcal{D} \circ \mathcal{E}(\mathbf{q}_i)\|_F^2 \quad (3.21)$$

$$\mathcal{L}_2 = \frac{2}{M(M-1)} \sum_{i=1}^M \sum_{j=i+1}^M \left( \|\mathcal{E}(\mathbf{q}_i) - \mathcal{E}(\mathbf{q}_j)\|_2 - \tilde{d}_{\text{field}}(\mathbf{q}_i, \mathbf{q}_j) \right)^2, \quad (3.22)$$

where  $M$  is the dataset size. The first term in the objective  $\mathcal{L}_1$  is the mean square error of the flow field reconstruction, which seeks to make the autoencoder approximate the identity

| Encoder                           |                | Decoder                            |               |
|-----------------------------------|----------------|------------------------------------|---------------|
| Layer                             | Data Size      | Layer                              | Data Size     |
| Input $(\bar{u}_x, \bar{u}_y, k)$ | (250, 100, 3)  | Dense                              | (64)          |
| Conv2D + BN (3,3,16)              | (250, 100, 16) | Dense                              | (128)         |
| MaxPooling2D                      | (125, 50, 16)  | Dense                              | (256)         |
| Conv2D + BN (3,3,32)              | (125, 50, 32)  | Dense                              | (1536)        |
| MaxPooling2D                      | (63, 25, 32)   | Reshape                            | (8, 3, 64)    |
| Conv2D + BN (3,3,32)              | (63, 25, 32)   | Conv2D + BN (3,3,32)               | (8, 3, 32)    |
| MaxPooling2D                      | (32, 13, 32)   | UpSampling2D                       | (16, 6, 32)   |
| Conv2D + BN (3,3,64)              | (32, 13, 64)   | Conv2D + BN (3,3,32)               | (16, 6, 32)   |
| MaxPooling2D                      | (16, 6, 64)    | Conv2D + BN (3,3,32)               | (16, 6, 32)   |
| Conv2D + BN (3,3,64)              | (16, 6, 64)    | UpSampling2D                       | (32, 12, 32)  |
| MaxPooling2D                      | (8, 3, 64)     | Conv2D + BN (3,3,16)               | (32, 12, 16)  |
| Conv2D (3,3,64)                   | (8, 3, 64)     | Conv2D + BN (3,3,16)               | (32, 12, 16)  |
| Reshape                           | (1536)         | UpSampling2D                       | (64, 24, 16)  |
| Dense                             | (256)          | Conv2D + BN (3,3,16)               | (64, 24, 16)  |
| Dense                             | (128)          | Conv2D + BN (3,3,16)               | (64, 24, 16)  |
| Dense                             | (64)           | UpSampling2D                       | (128, 48, 16) |
| Dense (Latent vector)             | (2)            | Conv2D + BN (3,3,8)                | (128, 48, 8)  |
| -                                 | -              | Conv2D + BN (3,3,8)                | (128, 48, 8)  |
| -                                 | -              | UpSampling2D                       | (250, 100, 8) |
| -                                 | -              | Output $(\bar{u}_x, \bar{u}_y, k)$ | (250, 100, 3) |

Table 3.1: Network architecture of the encoder and decoder models. The activation function used is ReLU.

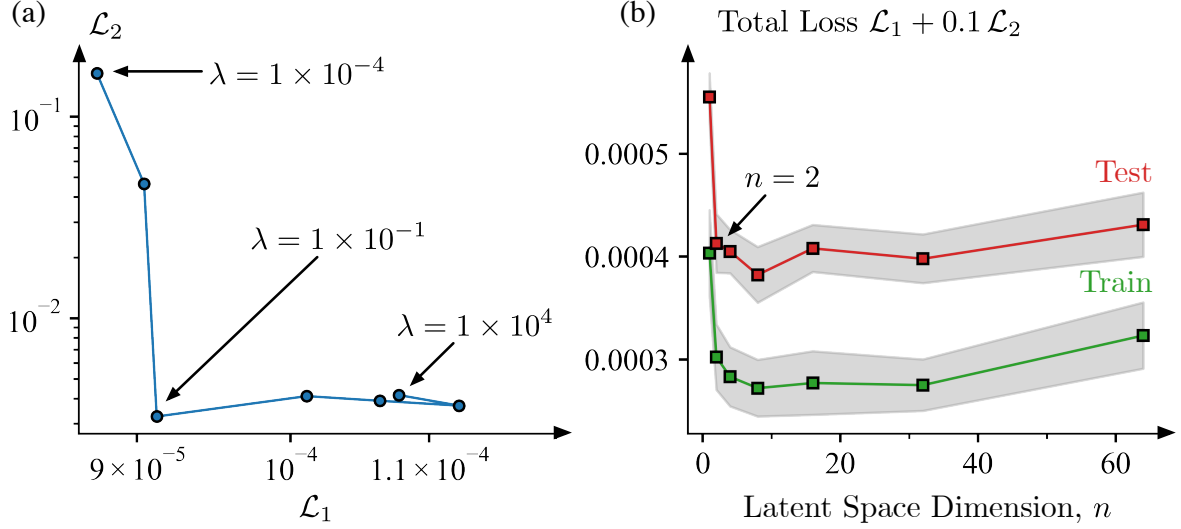


Figure 3.5: Example parameter study of the OT-based autoencoder for  $\alpha = 9^\circ$ . (a) L-curve showing the trade-off between  $\mathcal{L}_1$  and  $\mathcal{L}_2$  for the test set with  $10^{-4} \leq \lambda \leq 10^4$ . (b) Variation of the total loss  $\mathcal{L}_1 + \lambda \mathcal{L}_2$  with respect to the latent space dimension for  $\lambda = 0.1$ . Standard deviation for the last 500 epochs is colored in gray.

function such that  $\mathcal{D} \approx \mathcal{E}^{-1}$ .

We introduce the OT-based embedding loss term  $\mathcal{L}_2$ , defined as the squared difference comparing pairwise Euclidean distances of latent points with the OT-based flow field dissimilarities of their corresponding flow fields. With this term, we seek to impose a geometric structure on the embedded flow fields in which straight lines in the latent space should correspond to an approximate OT geodesic between inputs. If structures in the flow field are moved between two different inputs, resulting in a higher flow field dissimilarity, the model would essentially learn to embed these fields further apart in the latent space. Conversely, if two inputs have a small flow field dissimilarity, the model should learn to embed them close to one another in the latent space. This prevents the autoencoder from placing latent points arbitrarily relative to one another. With such an embedding, geometric properties of the latent space can be used to intuit physical trends or relationships between different controlled cases. We note that if necessary for downstream tasks, we can also include a sec-



ondary decoder  $\mathcal{F} : \mathbb{R}^m \rightarrow \mathbb{R}^d$  which maps from latent coordinates to estimates of physical parameters [FT23, TFI24]. However, for our particular demonstration, we found that using the relevant performance metrics  $\mathbf{p} = (\bar{C}_D, \bar{C}_L) \in \mathbb{R}^2$  as an auxiliary output did not have a significant impact on the overall results.

We utilize the AdamW optimizer [KB15] with an initial learning rate of  $1 \times 10^{-4}$  and weight decay parameter of  $10^{-2}$  for up to 4000 total epochs. An 80 : 20 split is performed between training and test data. The hyperparameter  $\lambda$  determines the relative influence of the embedding loss term. We choose this parameter by observing the tradeoff between the two loss terms as we vary  $\lambda$ . The tradeoff takes the form of an  $L$ -curve, or Pareto front, and we choose  $\lambda$  to be around the elbow of this curve [HO93].

### 3.5 Results

We examine the effectiveness of the present OT-based approach in identifying low-dimensional coordinates that succinctly capture the effects of the open-loop thermal control on the separated flow for controlled cases with  $\alpha = 6^\circ$  and  $9^\circ$ . In the following analyses, we consider separate autoencoders trained only on each angle of attack. For the choice of the  $\mathcal{L}_2$  loss weight  $\lambda$ , an L-curve analysis is shown in figure 3.5(a) with  $10^{-4} \leq \lambda \leq 10^4$ . We choose  $\lambda$  to be the value around the elbow of the curve, which occurs around  $\lambda = 0.1$ . In figure 3.5(b), we report the total loss for both training and test sets with respect to the choice of the latent space dimension with the chosen weight of  $\lambda = 0.1$ . The mean and standard deviation of the loss are computed using the last 500 epochs of training, where the loss has stabilized. For both angles of attack, we find that the loss sharply drops as we go from  $n = 1$  to  $n = 2$ , however, the loss for the test set seems to vary little after this point. We show a representative reconstruction from the OT autoencoder using a two-dimensional latent space for both angles of attack in figure 3.6. We report the reconstruction error for a given flow field image  $f$  with  $\varepsilon = 100 \times \|f_{\text{rec.}} - f_{\text{original}}\|_F / \|f_{\text{original}}\|_F$  for each reconstructed field. To

additionally assess the autoencoder performance, we compare  $\omega_z$  computed from the input and reconstructed velocities. We observe that the OT autoencoder is capable of qualitatively reconstructing the flow field from just a two-dimensional latent space, successfully recovering the general profile of the separation bubble and shear layer.

The performance of the two models (with  $n = 2$ ) for both angles of attack is summarized in table 3.2. The average errors for the flow fields  $f$  are reported using the Frobenius norm error metric. Note that if the discretized flow fields are flattened from matrices into vectors, this is equivalent to the vector  $\ell^2$  norm, which is usually reported. For the OT embedding term, error is measured using a normalized  $\ell^2$  error metric between the embedded latent distances and the OT-based dissimilarity of the corresponding flow fields:

$$\varepsilon_d = 100 \times \sqrt{\frac{\sum_{i=1}^M \sum_{j=i+1}^M \left( \|\mathcal{E}(\mathbf{q}_i) - \mathcal{E}(\mathbf{q}_j)\|_2 - \tilde{d}_{\text{field}}(\mathbf{q}_i, \mathbf{q}_j) \right)^2}{\sum_{i=1}^M \sum_{j=i+1}^M \|\mathcal{E}(\mathbf{q}_i) - \mathcal{E}(\mathbf{q}_j)\|_2^2}}, \quad (3.23)$$

which in this context is referred to as the multi-dimensional scaling (MDS) stress [KW78, DS00, BG05]. This is a measurement of how distorted the embedded distances are from the original dissimilarities, and is invariant under translation and uniform stretching of the dissimilarities and latent coordinates. An MDS stress value that is smaller than 15% is typically described as acceptable, although this threshold is heuristic and can be problem dependent [Kru64, KW78, BG05].

### 3.5.1 $\alpha = 9^\circ$ Latent Space

We begin by examining the  $\alpha = 9^\circ$  cases, which exhibit comparatively simpler flow responses than the  $\alpha = 6^\circ$  cases, primarily due to the absence of global laminarization in the flow responses [YT19]. In particular, we first observe the effect of the OT based embedding term on the learned latent space representation. We consider latent spaces learned by two representative autoencoder models as shown in figure 3.7: (a) a standard autoencoder trained

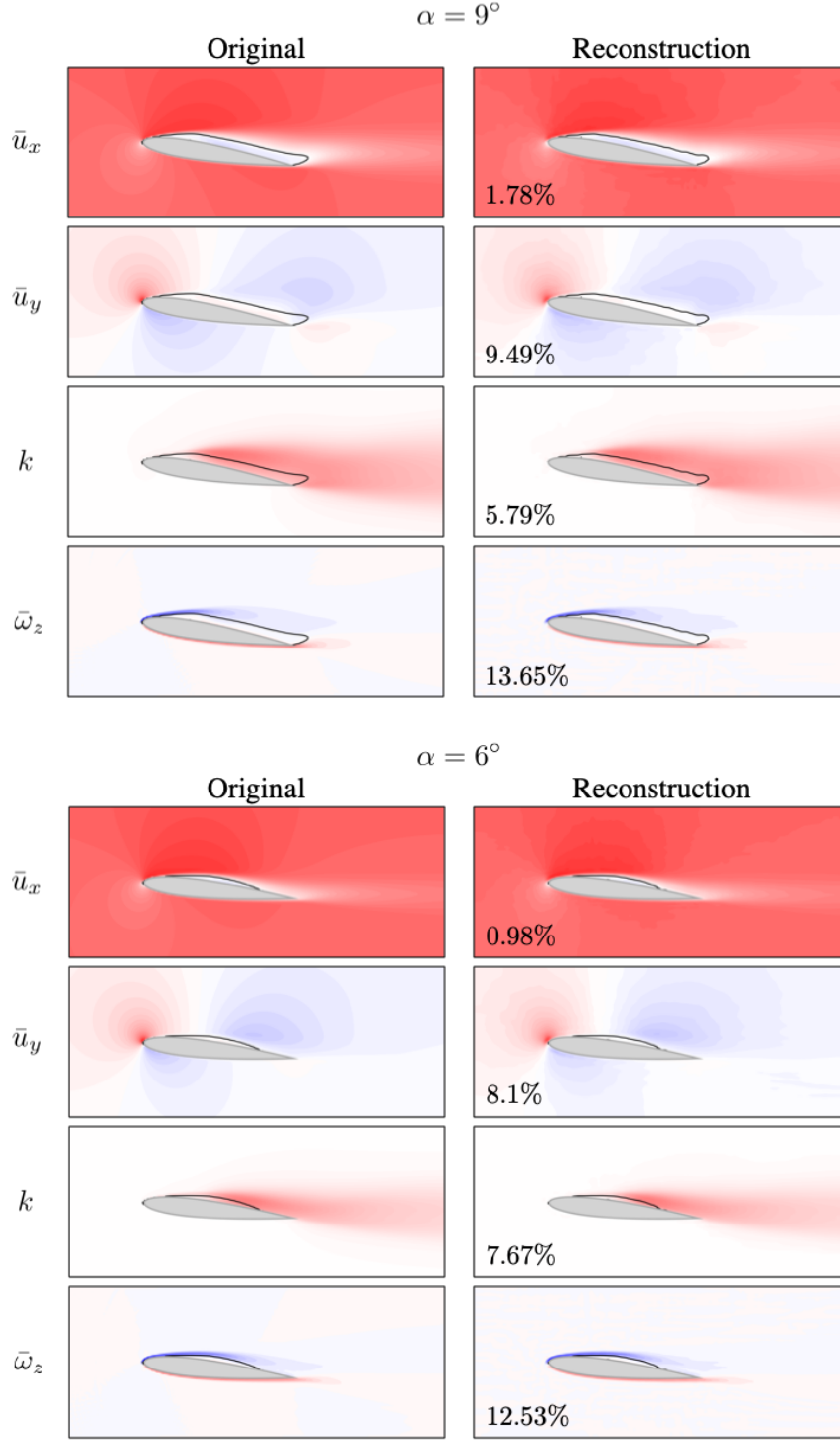


Figure 3.6: Reconstructions of baseline flow fields by OT autoencoder. The  $\bar{u}_x = 0$  isocontour is shown in black for all fields. The reconstructed vorticity is obtained by central differencing of the reconstructed velocity fields. The percent Frobenius norm reconstruction error is reported.

| AoA       | Quantity         | Reg. AE Err.<br>Train | Reg. AE Err.<br>Test | OT AE Err.<br>Train | OT AE Err.<br>Test |
|-----------|------------------|-----------------------|----------------------|---------------------|--------------------|
| $6^\circ$ | $\bar{u}_x$      | 1.11                  | 1.13                 | 1.25                | 1.37               |
|           | $\bar{u}_y$      | 9.99                  | 10.08                | 8.75                | 9.05               |
|           | $k$              | 7.86                  | 9.35                 | 8.45                | 10.36              |
|           | $\bar{\omega}_z$ | 13.26                 | 14.36                | 15.09               | 15.29              |
|           | OT Emb.          | -                     | -                    | 12.77               | 13.52              |
| $9^\circ$ | $\bar{u}_x$      | 1.38                  | 1.14                 | 1.42                | 1.47               |
|           | $\bar{u}_y$      | 8.59                  | 8.50                 | 8.50                | 8.48               |
|           | $k$              | 6.62                  | 7.18                 | 8.28                | 8.71               |
|           | $\bar{\omega}_z$ | 16.60                 | 16.48                | 14.89               | 14.76              |
|           | OT Emb.          | -                     | -                    | 13.82               | 15.00              |

Table 3.2: Comparison of average percent error for field variables using different autoencoder architectures (with two latent variables) at angles of attack  $\alpha = 6^\circ$  and  $9^\circ$ . Field variable errors are reported as percent Frobenius norm error. The embedding loss is reported using the MDS stress.

solely with reconstruction loss ( $\mathcal{L}_1$ ) and (b) the OT autoencoder, which includes the OT embedding term ( $\mathcal{L}_1$  and  $\mathcal{L}_2$ ). For ease of discussion, we rotate all latent spaces to align with their principal component axes so that we can describe how the flow fields vary along axes where the OT distances are estimated to vary the most. This transformation does not affect the loss or the interpretation of relative dissimilarity, as the reconstruction loss  $\mathcal{L}_1$  remains unchanged since the rotation can be reversed without loss, and the  $\mathcal{L}_2$  term is inherently invariant to uniform rotation of the latent space. We note that both models achieve comparable reconstruction performance, indicating that the additional loss term does not severely compromise the ability to accurately recover the flow fields.

Also depicted in figure 3.7 are the  $\bar{u}_x = 0$  isocontours for representative input cases, which exhibit the variation in the separation bubble behavior as we traverse the latent space. Case A in figure 3.7 shows the baseline flow, with a long separation bubble that spans the entire suction side. As we move from case A to F, we observe that the separation bubble shrinks towards the leading edge. Conversely, case G presents a qualitatively different flow regime.

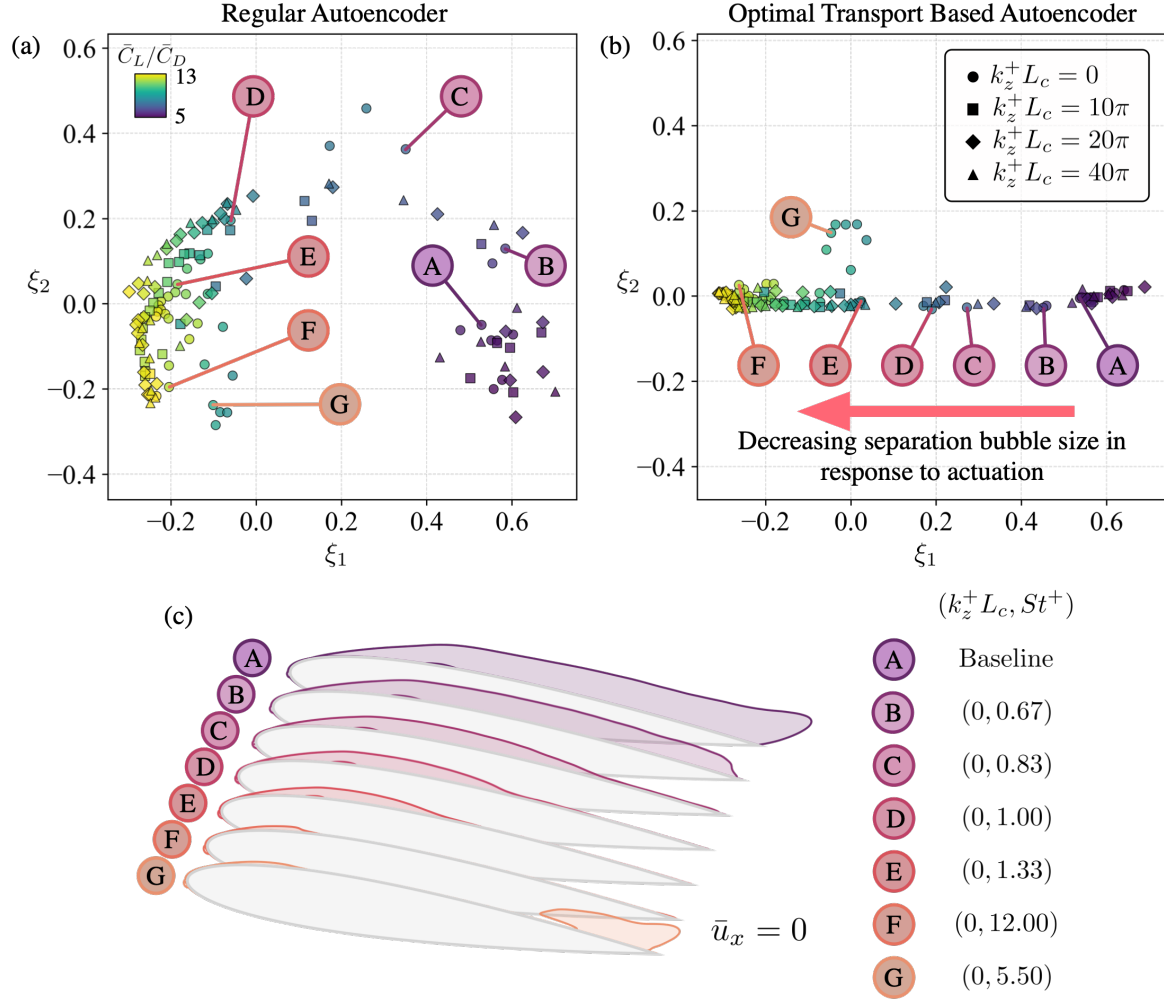


Figure 3.7: Comparison of learned latent spaces with  $\alpha = 9^\circ$  colored by aerodynamic performance  $\bar{C}_L/\bar{C}_D$  for (a) standard autoencoder ( $\mathcal{L}_1$  loss only) (b) OT autoencoder ( $\mathcal{L}_1$  and  $\mathcal{L}_2$  losses). (c) Isocontours of time-averaged streamwise velocity  $\bar{u}_x = 0$  shown for different representative cases labeled in (a) and (b).

Separation occurs near the trailing edge as spanwise rollers merge and break down near the trailing edge, forming a recirculation zone, and there is a pronounced increase in turbulent kinetic energy near the trailing edge. This is more clearly shown by case 1-2 in figure 3.8.

For both the regular autoencoder and the OT based latent space, we see a sequential progression from case A to F as the separation bubble shrinks, which correlates with the control performance in terms of the average lift-to-drag ratio. However, for the standard autoencoder in figure 3.7(a), while we can make out a qualitative trend of the lift-to-drag ratio in the latent space, the latent representations are unconstrained in geometry, and there is no inherent interpretation to the distance between latent points. We again emphasize that the relative positioning of the latent points for the standard autoencoder latent space in figure 3.7(a) is arbitrary up to diffeomorphism, which inhibits our ability to judge how similar the flow fields are based solely on their latent positions. In fact, the distinction between the outlier cases in which separation occurs near the trailing edge (case G) is not clearly made in the regular autoencoder latent space. Additionally, in the latent space learned by the regular autoencoder, the best-performing and worst-performing cases begin to approach each other, despite being the most qualitatively “dissimilar” cases.

In the OT-based latent space shown in figure 3.7(b), we find that the first latent coordinate  $\xi_1$  clearly corresponds with the variation in size of the separation bubble. As  $\xi_1$  decreases, the separation bubble size reduces and eventually vanishes, as seen in cases A-F. The second latent coordinate  $\xi_2$  captures changes in the flow field associated with the partial laminarization of the flow and the trailing-edge separation, which is specific to the  $k_z^+ = 0$  configurations, depicted by case G. Most cases are distributed along the  $\xi_1$  axis (i.e., first principal component), suggesting that the separation bubble size is the dominant mode of variation in the flow response according to the OT-based latent geometry. We once again emphasize that the ability to derive an interpretation for the principal components in the latent space arises because Euclidean distances in the latent space serve as surrogates for the OT-based snapshot dissimilarity.

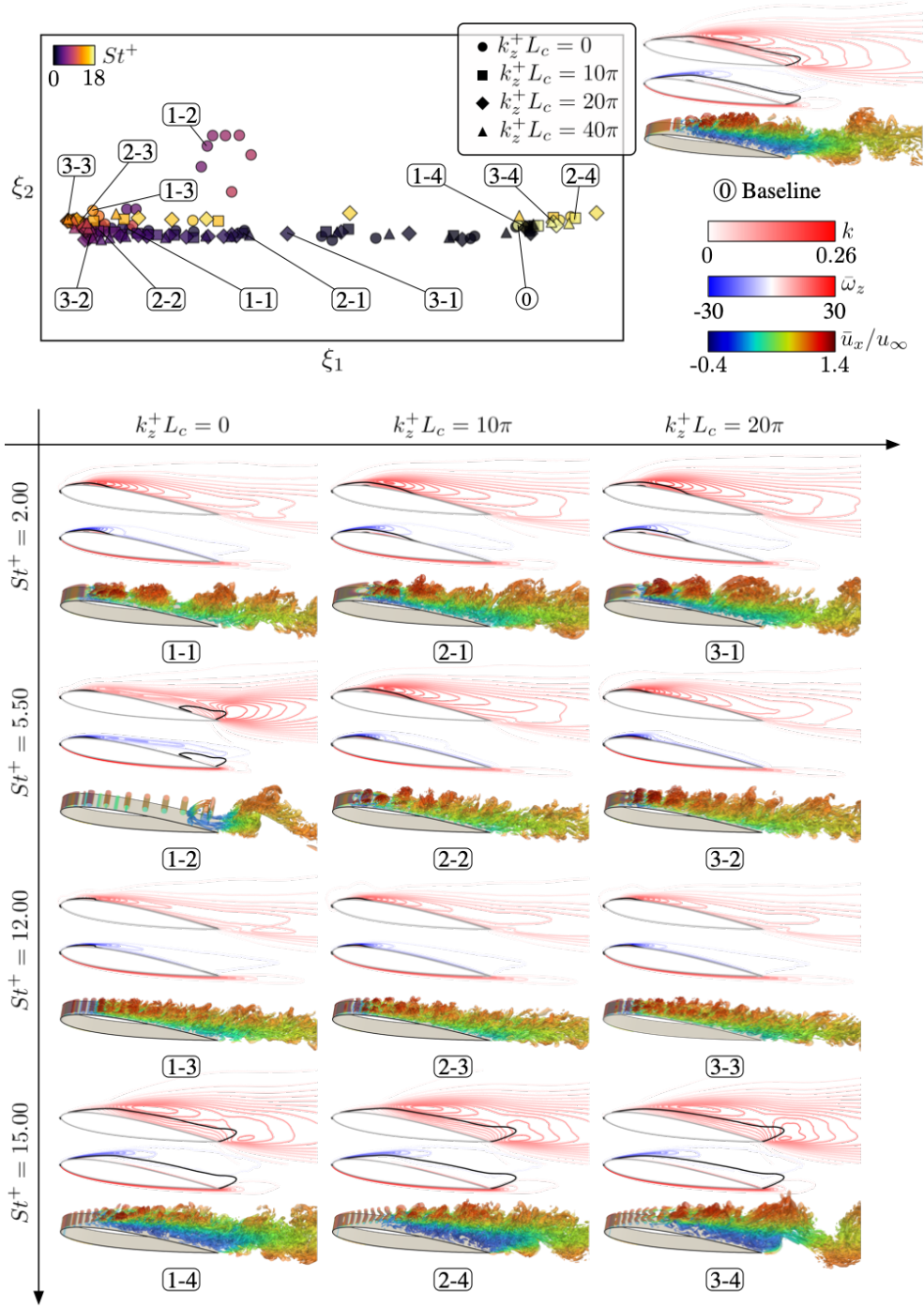


Figure 3.8: Plot of latent embeddings highlighting the influence of actuation parameters for  $\alpha = 9^\circ$ . Labeled are example actuation cases. Examples of average turbulent kinetic energy, average vorticity fields, as well as instantaneous  $Q$ -criterion ( $QL_c^2/u_\infty^2 = 50$ ) colored by streamwise velocity are shown for the labeled cases. The  $\bar{u}_x = 0$  isocontour is shown for all average fields.

From figure 3.7(b), we additionally observed that  $\xi_1$  is also strongly correlated with the control performance in terms of the average lift-to-drag ratio. Lower values of  $\xi_1$  are associated with reduced flow separation (hence a smaller separation bubble) and an increased lift-to-drag ratio. To understand the variation of the control performance in terms of the control input parameters, in figure 3.8 we show the OT-based latent space colored by the actuation frequency with marker shapes corresponding to the spanwise wave number of actuation. We find that the  $\xi_1$  coordinate or separation bubble size primarily depends on  $St^+$ , having a weaker dependence on  $k_z^+ > 0$ .

We also depict several representative cases from both two and three-dimensional actuation at various frequencies in figure 3.8, with their corresponding latent embeddings labeled. Starting from the baseline (case 0), increasing the forcing frequency within the range  $0 \lesssim St^+ \lesssim 4.5$  (e.g., cases 1-1, 2-1, and 3-1) leads to a rapid decrease of  $\xi_1$ , corresponding to an improved lift-to-drag ratio as was seen in figure 3.7(b). In this regime, flow separation is effectively suppressed, and the turbulent kinetic energy in the wake becomes lower, reflecting a reduction in the size of the separation bubble and the associated turbulent mixing region. We see that the two-dimensional actuation initially sees a faster rate of reduction of the separation bubble size with respect to increasing the actuation frequency, as seen in both the flow fields. This is also reflected in the relative placement of cases 1-1, 2-1, and 3-1 along  $\xi_1$ .

For  $5 \lesssim St^+ \lesssim 8$ , when  $k_z^+ = 0$ , there is an increase of  $\xi_1$  and  $\xi_2$  toward the cluster of outlier points (represented by case 1-2). Here, performance deteriorates for the two-dimensional actuation due to the reappearance of separation near the trailing edge, leading to a reduction in the lift-to-drag ratio (increase of  $\xi_1$ ). In this range, two-dimensional actuation induces partial laminarization over the suction surface, characterized by spanwise vortical structures that originate near the leading edge and break down further downstream, causing the trailing edge separation. For  $k_z^+ > 0$ , trailing edge separation is not observed, and there is no substantial increase of  $\xi_1$  during this intermediate range of forcing frequencies.



We see that the OT-based embedding clearly distinguishes the  $k_z^+ = 0$  cases with partial laminarization from the other control cases.

As the forcing frequency increases further to  $9 \lesssim St^+ \lesssim 13$ , the lift-to-drag ratio continues to improve, with the trailing edge separation suppressed for  $k_z^+ = 0$  (case 1-3), and we return to the left side of the  $\xi_1$  axis in latent space. It is during this range of forcing frequencies that we are the furthest in the negative  $\xi_1$  direction, and we observe the local optimal lift-to-drag ratio. This occurs at around  $St^+ \approx 12$  for  $k_z^+ = 10\pi$  and  $20\pi$  (cases 2-3 and 3-3). For  $St^+ \gtrsim 12$ ,  $\xi_1$  increases sharply, indicating a loss of actuator effectiveness. As a result, flow separation reappears, and lift-to-drag ratio declines, returning to baseline levels (cases 1-4, 2-4, 3-4).

### 3.5.2 $\alpha = 6^\circ$ Latent Space

Next, we analyze the latent embedding for the  $\alpha = 6^\circ$  cases, which exhibit a more diverse range of behaviors compared to  $\alpha = 9^\circ$ . To visualize the effect of the OT embedding for  $\alpha = 6^\circ$ , in figure 3.9 we show the latent embeddings for the  $\alpha = 6^\circ$  cases corresponding to the regular and OT-based autoencoders in addition to separation profiles for representative cases. Just as in the  $\alpha = 9^\circ$  case, while we see that we can interpret a trend of the lift-to-drag ratio for the regular autoencoder in figure 3.9(a), we again emphasize that the relative placement of cases is arbitrary, and thus the cases are scattered in the latent space without any regard to any notion of similarity between the flow fields. We see that the OT-based embedding shown in figure 3.9(b) more succinctly captures the separation bubble behavior (cases A-E) while distinguishing cases with laminarization and trailing edge separation (e.g., the branches containing cases F and G). In contrast to the  $\alpha = 9^\circ$  latent space, while we see that the lift-to-drag ratio visibly correlates with  $\xi_1$ , we also note that there is some dependence on  $\xi_2$  as seen with the performance of some of these laminarized cases.

Again, as shown in figure 3.9(b), the  $\xi_1$ -coordinate captures the dominant mode of variation in the latent space, with most cases being positioned along this axis. As was the case for

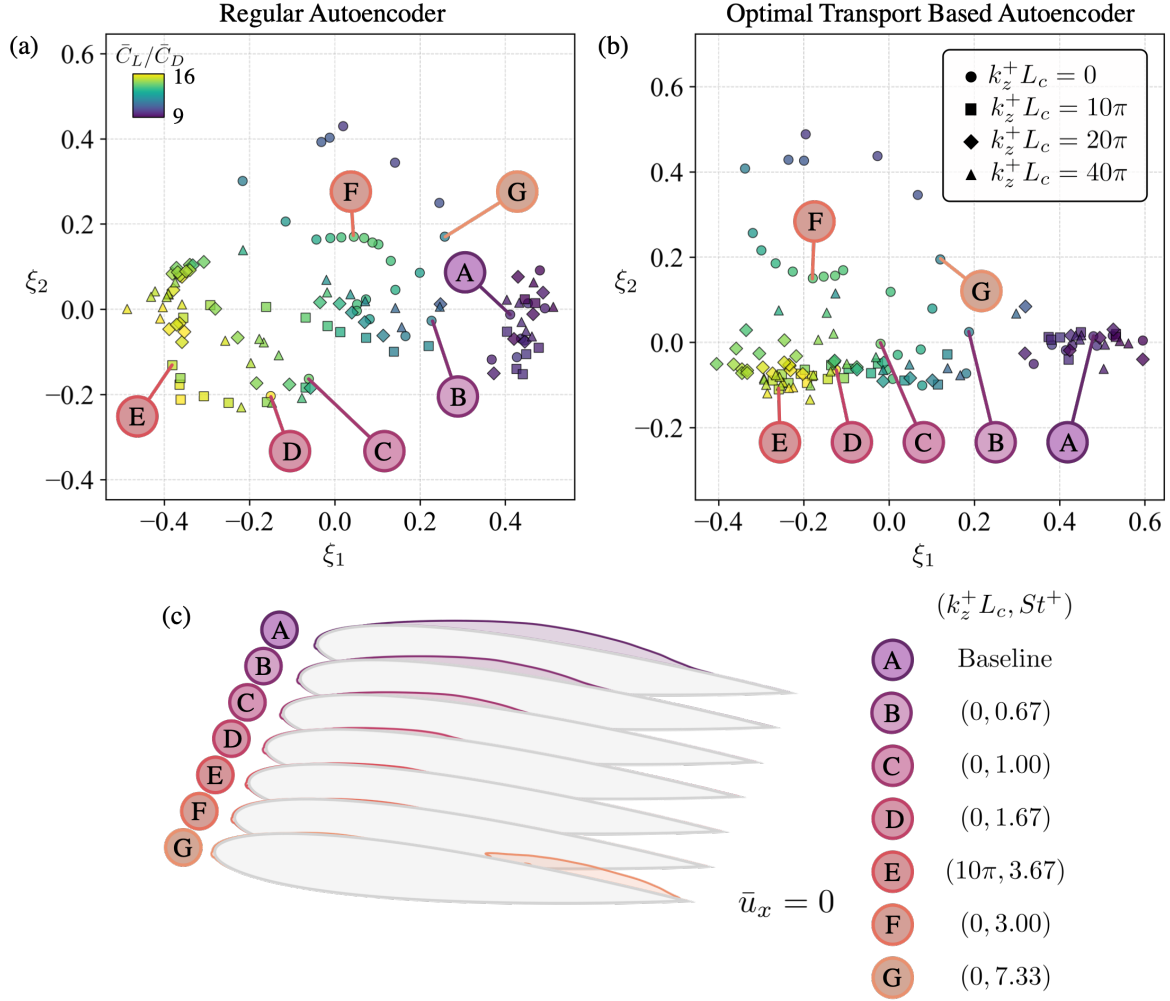


Figure 3.9: Comparison of learned latent spaces with  $\alpha = 6^\circ$  colored by control performance  $\bar{C}_L/\bar{C}_D$  for (a) standard autoencoder ( $\mathcal{L}_1$  loss only) (b) OT autoencoder ( $\mathcal{L}_1$  and  $\mathcal{L}_2$  losses). (c) Isocontours of time-averaged streamwise velocity  $\bar{u}_x = 0$  shown for different representative cases labeled in (a) and (b).

the  $\alpha = 9^\circ$  cases, a decrease in  $\xi_1$  generally corresponds to a shrinking bubble that moves toward the leading edge, before disappearing completely. Likewise, the  $\xi_2$ -coordinate identifies control cases in which the flow exhibits different behavior in the wake due to laminarization. However, a major distinguishing feature of the  $\alpha = 6^\circ$  latent space compared to the  $\alpha = 9^\circ$  one is that for the cases with leading edge separation, the separation bubble size alone does not as effectively characterize distinct cases. For example, cases E and F have very similar separation bubble profiles; however, they clearly exhibit different control performance and placement in the latent space. To understand why this is the case, we note that unlike the  $\alpha = 9^\circ$  cases, which only experience partial laminarization, the  $\alpha = 6^\circ$  latent space includes cases in which separation is suppressed, and the flow field is globally laminarized (case F). To make this point clearer, in figure 3.8, we again show the variation of the latent space with respect to the control parameters in addition to example average and instantaneous flow fields corresponding to labeled cases. For cases with global laminarization, spanwise vortical structures that originate from the leading edge indicate the suppression of three-dimensional flow structures as they advect downstream. These cases are distinguished from partially laminarized fields that have trailing-edge separation due to the breakdown of these spanwise structures (e.g., case G in figure 3.9(c) or case 1-3 in figure 3.8). This manifests in cases with very similar separation profiles, but different wake behavior. Additional effects associated with the laminarization of the flow captured in the  $\xi_2$  coordinate are seen in the variation of the turbulent kinetic energy fields, which also play a significant role in the estimation of the lift-to-drag ratio for  $\alpha = 6^\circ$ .

Despite the aforementioned differences, the overall behavior within the latent space when the forcing frequency is varied parallels that of the  $\alpha = 9^\circ$  cases. If we start at the baseline flow (case 0) in figure 3.10, increasing the forcing frequency within  $0 \lesssim St^+ \lesssim 2$  results in an initial decrease of  $\xi_1$  corresponding to the suppression of separation and improvement of the control performance (e.g., cases 1-1, 2-1, 3-1).

For cases with  $k_z^+ > 0$ , increasing  $St^+$  past this range would result in moving further left

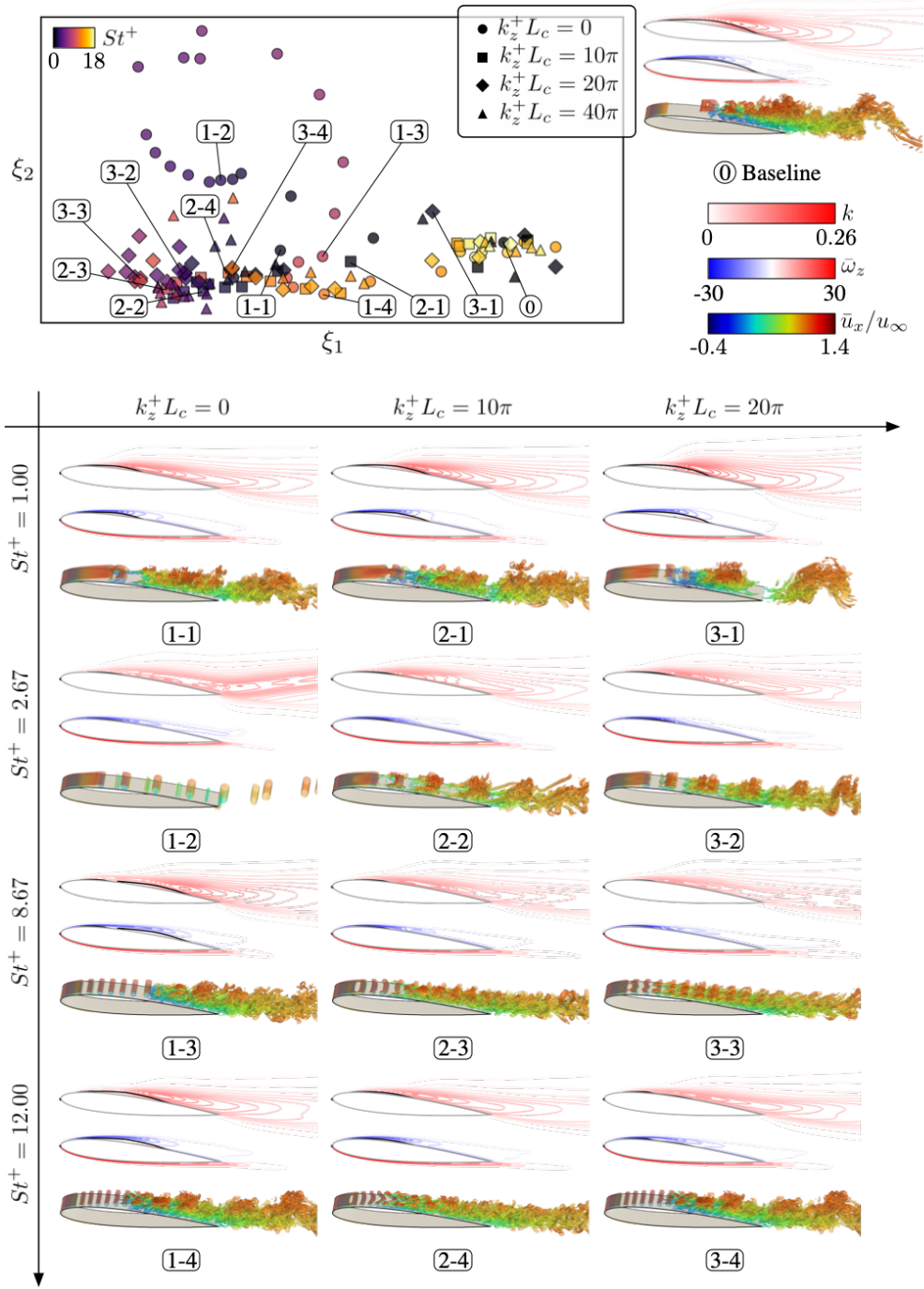


Figure 3.10: Plot of latent embeddings highlighting the influence of actuation parameters for  $\alpha = 6^\circ$ . Labeled are example actuation cases. Examples of average turbulent kinetic energy, average vorticity fields, as well as instantaneous  $Q$ -criterion ( $QL_c^2/u_\infty^2 = 50$ ) colored by streamwise velocity are shown for the labeled cases. The  $\bar{u}_x = 0$  isocontour is shown for all average fields.

along  $\xi_1$  until the local maxima in the lift-to-drag ratio is reached in the range of  $2 \lesssim St^+ \lesssim 12$  as separation is eventually completely suppressed (e.g., cases 2-2, 3-2). For  $k_z^+ = 0$ , while  $\xi_1$  is decreased, reflecting the suppression of separation, around  $St^+ \approx 2$  ( $\xi_1 \approx -0.1$ ), there is a vertical jump in  $\xi_2$  corresponding to cases experiencing global laminarization (case 1-2). Here we see that the turbulent kinetic energy field shows a distinctly different appearance in the separated flow. The wake is directed upwards and is elongated towards the trailing edge, reflecting the advection of the pairs of spanwise rollers downstream. In this globally laminarized regime, even though there is a reattachment of the wake, there is a decrease in the lift-to-drag ratio when compared to cases with similar separation bubble sizes (similar  $\xi_1$  position).

When  $k_z^+ = 0$  and  $4 \lesssim St^+ \lesssim 8$ , the vortical structures merge and break down near the trailing edge (e.g., case G in figure 3.9), resulting in separation and a further decrease in performance. Eventually, the separation bubble begins to move back up towards the leading edge and rejoins the main distribution of cases along the  $\xi_1$  axis (cases 1-3). Moving between cases 1-2 and 1-3, we see that  $\xi_2$  primarily characterizes differences in the trailing edge behavior. These cases observed strengthened turbulent kinetic energy mixing near the trailing edge and the wake, while cases along the  $\xi_1$  axis primarily have the strongest turbulent kinetic energy at the rear edge of the separation bubble over the mid-chord and leading edge. For  $St^+ \gtrsim 12$ ,  $\xi_1$  increases for all  $k_z^+$  as we again return to baseline performance as the controller has diminished effectiveness (e.g., cases 1-4, 2-4, 3-4).

In summary, these findings suggest that with the use of OT, the wide array of complex flow responses to thermal actuation can be effectively characterized by a low-dimensional representation that is consistent across the two angles of attack studied. For both  $\alpha = 6^\circ$  and  $9^\circ$ , the dominant mode of variation in response to control  $\xi_1$  captures variations in separation bubble size and shows a strong correlation with the lift-to-drag ratio. Secondary effects are included in  $\xi_2$ , which encodes the behavior of the turbulent kinetic energy wake response corresponding to laminarization of the flow and the position of separation.

### 3.6 Conclusions

We introduced an autoencoder framework that incorporates unbalanced OT distances to learn latent representations of flow fields that encode a physically interpretable notion of similarity. Unlike standard autoencoders, which may organize latent variables arbitrarily, provided reconstruction accuracy is obtained, the OT-based approach imposes a notion of geometric structure by attempting to align pairwise distances in latent space with OT geodesics in the input space. When applied to flows over a NACA 0012 airfoil subject to unsteady thermal actuation, we are capable of obtaining an interpretable representation of the wide array of responses to control in a low-dimensional space. The use of OT as a dissimilarity metric in this problem setting allows us to capture displacements of flow structures and aerodynamic performance trends in response to different control inputs.

The current results demonstrate that the flow response to thermal actuation can be succinctly represented by two latent variables with the use of the OT-based embedding. The first latent coordinate captures the primary response of the flow to control. Along this coordinate, we explicitly observed a correlation between the separation bubble size and the control performance in terms of the lift-to-drag ratio. The second coordinate was found to contain information on changes in the wake due to laminarization of the flow and trailing edge separation. This interpretation of the learned latent coordinates in the OT-based approach was consistent across the two angles of attack studied. This consistency was not observed with the standard autoencoder formulation, suggesting the OT-based autoencoder uncovered shared physical relationships across the two sets of cases despite being trained separately for each angle of attack.

While this study employed a relatively simple scheme based on matching Euclidean distances in the latent space to OT-based dissimilarities in the input space, one may incorporate the intrinsic non-Euclidean geometry of the learned manifold. Although our analysis focused on a two-dimensional control parameter space,  $(k_z^+, St^+)$ , the present framework can be ap-

plied for higher-dimensional parameter spaces where interpretability becomes increasingly difficult and a naive exploration of parameters is expensive. Additionally, while the current study analyzed time-averaged flow fields, an extension to time-resolved or subsampled spatial domains [FT24] presents a promising direction for the utilization of the OT-based approach for analyzing flow structures and control mechanisms. Additionally, it may be of interest to consider OT distances between sets of states, rather than individual flow fields, using ground cost metrics such as the Hausdorff distance between attractors [IKM19], the Fréchet distance [AG95], or dynamic time warping [BC94]. Given that we have also learned a relationship between the latent coordinates and the control performance, future efforts can also address design optimization in addition to uncertainty quantification. Additionally, the present approach may assist in other downstream tasks such as generative modeling, given that interpolants in the latent space are trained to coincide with OT-distances, which may improve the quality of interpolation. With this in mind, the current results demonstrate the utility of OT in the representation and interpretation of complicated fluid physics.

## CHAPTER 4

# Reduced-Order State Estimation of Fluid Flows via Diffusion Model Augmented Particle Filtering

### 4.1 Introduction

In many fluid mechanics applications, state estimation and data assimilation are challenging due to the high dimensionality of flow fields, strongly nonlinear dynamics, and non-Gaussian uncertainty structures [AJS08]. Reconstructing flow states from sparse sensor measurements is often an ill-posed inverse problem, as multiple flow realizations may be consistent with the same observations, particularly in the presence of noise. Deterministic estimation methods typically provide a single average solution but fail to characterize the uncertainty and multimodality of the posterior distribution. A probabilistic framework is therefore desirable, enabling the representation of multiple plausible flow states together with their associated probabilities.

Recent advances in diffusion models have provided a powerful framework for learning complex high-dimensional probability distributions without restrictive parametric assumptions [SSK20, GV25, DCF26, DPF24]. Their ability to represent highly non-Gaussian and multimodal distributions makes them particularly attractive for inverse problems and state estimation. In this work, we develop a reduced-order Bayesian state estimation framework that combines autoencoder-based manifold learning, score-based diffusion models, and particle filtering. Building on recent latent-space sequential filtering methodologies [MJE25], we replace Gaussian latent-state assumptions with learned nonparametric representations of



the posterior distribution.

The framework operates on a low-dimensional latent manifold identified through an autoencoder. A conditional score-based diffusion model is trained to approximate the posterior distribution of latent states conditioned on sparse sensor observations, while a neural ordinary differential equation (neural ODE) surrogate models latent dynamics and propagates trajectories forward in time.

For sequential data assimilation, particle filtering is performed in latent space. The neural ODE provides predictive dynamics for particle propagation, while the diffusion model generates observation-consistent latent samples for the proposal distribution. Particle weights are updated using the observation likelihood, enabling recursive approximation of the posterior as new measurements become available. Performing inference and propagation within the reduced latent representation yields a tractable framework for Bayesian state estimation of nonlinear fluid systems with strongly non-Gaussian uncertainty.

As a representative test case, we consider gust–airfoil interactions, which exhibit flow separation, unsteadiness, and nonlinear dynamics. This problem provides a challenging benchmark for probabilistic state reconstruction from sparse measurements and for evaluating the ability of the proposed framework to capture multimodal posterior distributions during sequential Bayesian data assimilation.

## 4.2 Probabilistic Autoencoder-Based Forward Model

Let  $x \in \mathbb{R}^n$  denote the discretized flow-field state and let  $y \in \mathbb{R}^r$  denote the corresponding noisy sensor observations. Our objective is to characterize the posterior distribution of the state conditioned on the observations,  $p(x \mid y)$ . Although the physical flow field is in actuality an infinite-dimensional object, throughout this work we assume a finite-dimensional discretization (e.g., via a computational mesh or finite-dimensional basis expansion), so that  $x$  represents the discretized state. This assumption allows us to work with probability

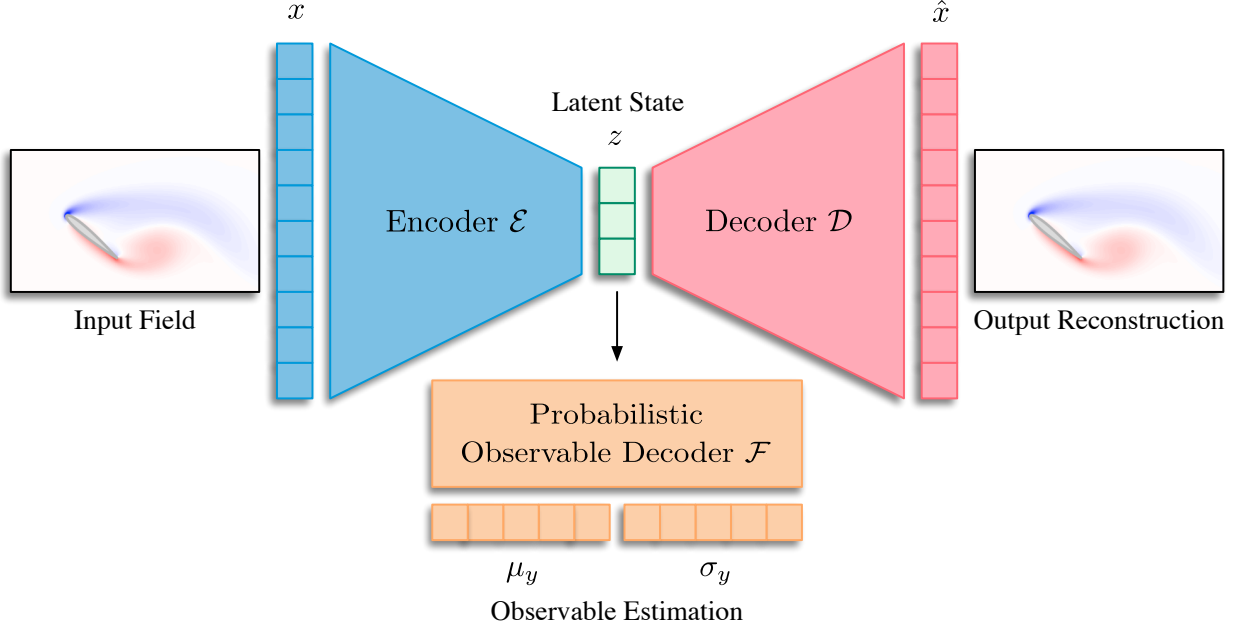


Figure 4.1: Autoencoder with probabilistic observable decoder.

densities with respect to the Lebesgue measure on  $\mathbb{R}^n$ , avoiding the complications that arise in infinite-dimensional function spaces, where no analogue of the Lebesgue measure exists,<sup>1</sup> at the expense of foregoing guarantees of discretization independence.

We employ a variational autoencoder (VAE) consisting of an encoder–decoder pair  $(\mathcal{E}, \mathcal{D})$ . Rather than mapping the input to a deterministic latent representation, in this case the encoder  $\mathcal{E} : \mathbb{R}^n \rightarrow \mathbb{R}^m \times \mathbb{R}^m$  predicts the parameters of a Gaussian approximate posterior,  $(\mu_z, \log \sigma_z^2) = \mathcal{E}(x)$ , which defines a distribution  $q_\phi(z | x) = \mathcal{N}(z; \mu_z(x), \text{diag}(\sigma_z^2(x)))$ , where  $\phi$  denotes the encoder parameters. A latent sample is obtained using the reparameterization trick,

$$z = \mu + \sigma \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I),$$

---

<sup>1</sup>The Lebesgue measure on  $\mathbb{R}^n$  is a nontrivial, nonnegative, translation-invariant, and locally finite Borel measure that provides a canonical notion of volume with respect to which probability densities are defined. No such measure exists on infinite-dimensional Banach spaces, so there is no infinite-dimensional analogue of the Lebesgue measure. Consequently, probability distributions in function spaces cannot, in general, be represented by densities with respect to a canonical reference measure. Infinite-dimensional probabilistic inference instead commonly defines distributions using Gaussian measures, but such formulations are beyond the scope of this work.

thereby allowing gradients to propagate through the stochastic sampling operation during backpropagation. The decoder then reconstructs the flow field,  $\hat{x} = \mathcal{D}(z)$ , where  $m \ll n$ . The latent variable  $z$  provides a low-dimensional representation of the flow field, while the decoder reconstructs an approximation of the original state. Rather than approximating the high-dimensional posterior density  $p(x | y)$  directly, we instead seek to learn the latent posterior density  $p(z | y)$ , from which samples of the physical state can subsequently be recovered through the decoder.

To relate the latent representation to the observations, we introduce a probabilistic observation model

$$\mathcal{F} : \mathbb{R}^m \rightarrow \mathbb{R}^r \times \mathbb{R}^r,$$

which predicts the parameters of a Gaussian conditional density,  $(\mu_y, \sigma_y) = \mathcal{F}(z)$ . The conditional density of the observations is therefore modeled as  $p_\theta(y | z)$ , where the neural network parameters are collectively denoted by  $\theta$ . We train the autoencoder and probabilistic observation model jointly by minimizing

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{x \sim P_x} [\|x - \hat{x}\|_2^2] \tag{4.1}$$

$$+ \mathbb{E}_{x \sim P_x} [D_{\text{KL}}(q_\phi(z | x) \parallel p(z))] \tag{4.2}$$

$$+ \lambda \mathbb{E}_{(x,y) \sim P_{\text{data}}} [-\log p_\theta(y | z)], \tag{4.3}$$

where  $P_{\text{data}}$  denotes the unknown joint probability measure of  $(x, y)$  and

$$P_x = \int_{\mathbb{R}^r} P_{\text{data}}(\cdot, dy)$$

is its marginal on the state space.

The second term is the Kullback–Leibler divergence between the approximate posterior  $q_\phi(z | x)$  and the latent prior

$$p(z) = \mathcal{N}(0, I),$$

which regularizes the latent representation toward a standard Gaussian distribution. One can opt to weigh this term to control the strength of the latent regularization. However, for this study, we opt for a weight of unity, as is the case in the standard VAE objective.

Since  $P_{\text{data}}$  is unknown, the expectations are approximated using the empirical probability measure

$$\hat{P}_{\text{data}} = \frac{1}{N} \sum_{k=1}^N \delta_{(x_k, y_k)},$$

where  $\delta_{(x_k, y_k)}$  denotes the Dirac probability measure on the product space concentrated at  $(x_k, y_k)$ . The expression for the empirical marginal follows since, for every Borel set  $A \subseteq \mathbb{R}^n$ ,<sup>2</sup>

$$\begin{aligned} \hat{P}_x(A) &= \int_{\mathbb{R}^r} \hat{P}_{\text{data}}(A, dy) = \int_{\mathbb{R}^r} \frac{1}{N} \sum_{k=1}^N \delta_{(x_k, y_k)}(A, dy) \\ &= \frac{1}{N} \sum_{k=1}^N \delta_{(x_k, y_k)}(A \times \mathbb{R}^r) = \frac{1}{N} \sum_{k=1}^N (\delta_{x_k} \otimes \delta_{y_k})(A \times \mathbb{R}^r) \\ &= \frac{1}{N} \sum_{k=1}^N \delta_{x_k}(A) \delta_{y_k}(\mathbb{R}^r) = \frac{1}{N} \sum_{k=1}^N \delta_{x_k}(A). \end{aligned}$$

Substituting the empirical marginal measure into the reconstruction objective gives

$$\begin{aligned} \mathbb{E}_{x \sim \hat{P}_x} [\|x - \hat{x}\|_2^2] &= \int_{\mathbb{R}^n} \|x - \hat{x}\|_2^2 \hat{P}_x(dx) \\ &= \frac{1}{N} \sum_{k=1}^N \int_{\mathbb{R}^n} \|x - \hat{x}\|_2^2 \delta_{x_k}(dx) \\ &= \frac{1}{N} \sum_{k=1}^N \|x_k - \mathcal{D}(\mathcal{E}(x_k))\|_2^2, \end{aligned}$$

which is precisely the standard mean-squared reconstruction loss used for autoencoder train-

---

<sup>2</sup>The symbol  $\otimes$  denotes the product of measures. For Dirac measures,  $\delta_{(x_k, y_k)} = \delta_{x_k} \otimes \delta_{y_k}$ , so that  $(\delta_{x_k} \otimes \delta_{y_k})(A \times B) = \delta_{x_k}(A) \delta_{y_k}(B)$ .

ing. Similarly, the empirical approximation of the KL divergence term is

$$\mathbb{E}_{x \sim \hat{P}_x} [D_{\text{KL}}(q_\phi(z | x) \parallel p(z))] = \frac{1}{N} \sum_{k=1}^N D_{\text{KL}}(q_\phi(z | x_k) \parallel \mathcal{N}(0, I)),$$

which, under the Gaussian parameterization adopted here, admits the closed-form expression

$$D_{\text{KL}}(q_\phi(z | x_k) \parallel \mathcal{N}(0, I)) = \frac{1}{2} \sum_{i=1}^m (\mu_{z,i}^2 + \sigma_{z,i}^2 - \log \sigma_{z,i}^2 - 1).$$

In our implementation, the encoder predicts the log-variance, which is clamped to the interval  $\log \sigma_{z,i}^2 \in [-10, 10]$  prior to exponentiation. This prevents extremely small or large variance values, thereby improving numerical stability during optimization while leaving the theoretical form of the KL divergence unchanged.

Finally, substituting the empirical joint measure into the likelihood term yields

$$\begin{aligned} \mathbb{E}_{(x,y) \sim \hat{P}_{\text{data}}} [-\log p_\theta(y | z)] &= \int_{\mathbb{R}^n \times \mathbb{R}^r} -\log p_\theta(y | \mathcal{E}(x)) \hat{P}_{\text{data}}(d(x, y)) \\ &= \frac{1}{N} \sum_{k=1}^N -\log p_\theta(y_k | \mathcal{E}(x_k)). \end{aligned}$$

For the Gaussian observation model used in this work, the negative log-likelihood can be evaluated in closed form (up to an additive constant) as

$$\frac{1}{2N} \sum_{k=1}^N \sum_{j=1}^{N_{\text{sens}}} \left[ \frac{(\mu_{y,k,j} - y_{k,j})^2}{\sigma_{y,k,j}^2} + \log \sigma_{y,k,j}^2 \right].$$

For numerical stability, the predicted variance is lower-bounded by a small constant  $\varepsilon > 0$ , yielding the implemented loss

$$\frac{1}{2N} \sum_{k=1}^N \sum_{j=1}^{N_{\text{sens}}} \left[ \frac{(\mu_{y,k,j} - y_{k,j})^2}{\max(\sigma_{y,k,j}^2, \varepsilon)} + \log(\max(\sigma_{y,k,j}^2, \varepsilon)) \right].$$

The lower bound prevents division by zero and ensures that the logarithm remains well-defined, thereby improving the robustness of optimization under finite-precision arithmetic.

Consequently, minimizing  $\mathcal{L}(\theta, \phi)$  corresponds to empirical risk minimization with respect to the empirical probability measures induced by the training data. The reconstruction term encourages accurate recovery of the flow field, the KL divergence regularizes the latent representation toward the Gaussian prior, thereby promoting a smooth and well-structured latent manifold, and the negative log-likelihood term trains the probabilistic observation model to estimate the conditional density of the sensor measurements given the latent state. Once trained, the model provides a low-dimensional probabilistic forward model that is well suited for Bayesian inference in the latent space.

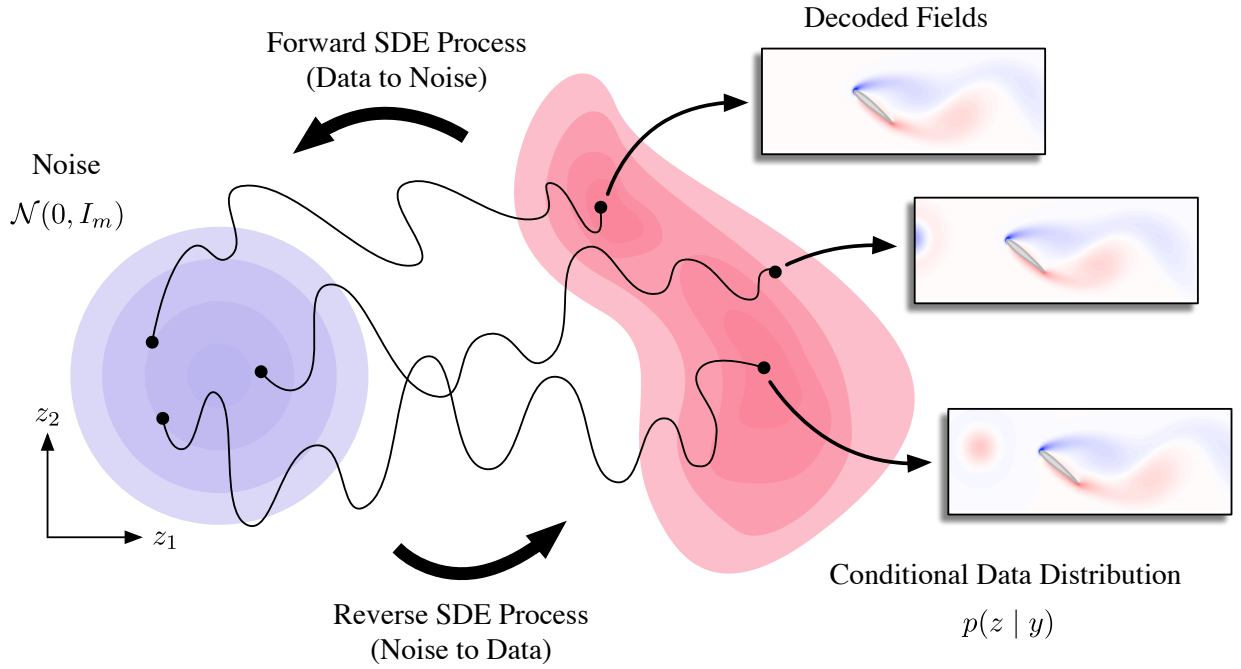


Figure 4.2: Illustration of the latent diffusion process.

### 4.3 Probabilistic Latent State Estimation from Sensor Measurements with Conditional Score-Based Diffusion Models

Once the autoencoder has been trained, we aim to model the conditional distribution of latent states given sensor measurements, i.e.,

$$p(z \mid y),$$

where  $z \in \mathbb{R}^m$  denotes the latent state and  $y$  denotes the corresponding observation.

Let  $P_{\text{data}}$  denote the probability measure on the original state space. Since the encoder  $\mathcal{E}$  defines a deterministic mapping from the state space to the latent space, the induced latent distribution is given by the pushforward measure  $P_0 = \mathcal{E}_\# P_{\text{data}}$ .

This formulation implies that sampling  $z \sim P_0$  is equivalent to sampling  $x \sim P_{\text{data}}$  and computing  $z = \mathcal{E}(x)$ . The advantage of this construction is that the latent diffusion model operates in a lower-dimensional space that preserves the “essential” structure of the original data distribution while reducing its computational cost.

The conditional diffusion model consists of a generative procedure that transforms a simple reference prior distribution into the unknown conditional data distribution. This is achieved by constructing a forward noising process that gradually perturbs data into noise, and learning a reverse-time process that inverts this transformation conditioned on observations.

These processes are formulated in continuous time using stochastic differential equations (SDEs), which provide a mathematically grounded framework for defining both the forward corruption process and the learned generative dynamics. We describe the basic details of the implementation of the diffusion model in the present context, but refer the reader to [SSK20, SE20, DCF26] for further specifics.

#### 4.3.1 Forward Diffusion Process

We define a continuous-time stochastic process  $z_\tau$ , with pseudo-time  $\tau \in [0, T]$ , such that

$$z_0 \sim P_0, \quad z_T \sim P_T,$$

where  $P_T$  is chosen to be a tractable reference distribution, typically  $\mathcal{N}(0, I)$ . The forward diffusion process is governed by the Itô SDE

$$dz_\tau = f(z_\tau, \tau) d\tau + g(\tau) dW_\tau, \quad (4.4)$$

where  $f$  is a drift function,  $g$  is a scalar diffusion coefficient, and  $W_\tau$  is an  $m$ -dimensional Wiener process.

The drift term determines the deterministic evolution of the latent variables over time, while the diffusion term injects stochastic noise that progressively corrupts information about the initial condition. Together, these terms define a family of marginal distributions  $\{P_\tau\}_{\tau \in [0, T]}$  that smoothly interpolates between the data distribution  $P_0$  and the reference distribution  $P_T$ . In our work, we experimented with both the variance-exploding (VE) and variance-preserving (VP) versions of the forward process.

#### 4.3.2 Variance-Exploding (VE) and Variance-Preserving (VP) Cases

The general SDE formulation induces the evolution of the marginal probability density  $p_\tau(z)$  governed by the Fokker–Planck equation

$$\frac{\partial p_\tau(z)}{\partial \tau} = -\nabla \cdot (f(z, \tau) p_\tau(z)) + \frac{1}{2} g^2(\tau) \nabla^2 p_\tau(z), \quad (4.5)$$

which characterizes how probability mass is transported and diffused over time.

Within this framework, the variance-preserving (VP) and variance-exploding (VE) for-



mulations correspond to different choices of drift and diffusion coefficients, leading to different evolutions of the marginal distribution  $p_\tau(z)$  while both converging to a tractable reference distribution.

**VP-SDE (variance-preserving formulation).** In the VP case, the forward process is defined by

$$f(z, \tau) = -\frac{1}{2}\beta(\tau)z, \quad g^2(\tau) = \beta(\tau), \quad (4.6)$$

which yields the Fokker–Planck equation

$$\frac{\partial p_\tau(z)}{\partial \tau} = \frac{\beta(\tau)}{2} \nabla \cdot (z p_\tau(z)) + \frac{\beta(\tau)}{2} \nabla^2 p_\tau(z). \quad (4.7)$$

This form shows that probability mass is simultaneously: (i) contracted toward the origin by the divergence term, and (ii) diffused outward by the Laplacian term.

These two effects balance such that the signal is gradually attenuated while the variance of the process remains bounded. The corresponding marginal distribution admits the closed-form representation

$$z_\tau = \alpha(\tau)z_0 + \sigma(\tau)\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad (4.8)$$

where  $\alpha(\tau)$  is a deterministic decay function induced by  $\beta(\tau)$ , and  $\sigma(\tau)$  is the accumulated noise scale.

**VE-SDE (variance-exploding formulation).** In contrast, the VE formulation removes deterministic drift and directly controls the growth of variance via the diffusion coefficient:

$$f(z, \tau) = 0, \quad g^2(\tau) = \frac{d}{d\tau}\sigma^2(\tau), \quad (4.9)$$

which leads to the simplified Fokker–Planck equation

$$\frac{\partial p_\tau(z)}{\partial \tau} = \frac{1}{2} \frac{d}{d\tau} \sigma^2(\tau) \nabla^2 p_\tau(z). \quad (4.10)$$

In this case, probability mass evolves purely through diffusion, with no deterministic transport term. This results in a monotonically increasing variance, and the marginal distribution takes the form

$$z_\tau = z_0 + \sigma(\tau)\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \quad (4.11)$$

which explicitly shows that uncertainty is injected over time without attenuation of the signal component.

### 4.3.3 Reverse-Time Generative Process

The forward diffusion process admits a corresponding reverse-time SDE, which reconstructs samples from  $P_0$  starting from noise distributed according to  $P_T$  [And82, SSK20]. For a given realization sampled from the reference distribution  $z_T \sim P_T$ , the reverse-time dynamics are given by

$$dz_\tau = [f(z_\tau, \tau) - g^2(\tau) \nabla_z \log p_\tau(z_\tau | y)] d\tau + g(\tau) d\bar{W}_\tau, \quad (4.12)$$

where  $\bar{W}_\tau$  denotes a Wiener process evolving backward in time, and  $p_\tau(z | y)$  is the conditional density of the latent state at time  $\tau$ .

The key quantity in this expression is the *score function*,  $\nabla_z \log p_\tau(z | y)$ , which defines the direction of steepest ascent of the log-probability density. Intuitively, this term guides the reverse process toward regions of higher posterior likelihood, ensuring that samples generated from noise are consistent with both the learned data distribution and the conditioning observation  $y$ . In practice, this score function is unknown and must be approximated from data.

#### 4.3.3.1 Learning the conditional score function.

To approximate the conditional score, we introduce a neural network  $s_\theta(z, \sigma, y)$ , parameterized by  $\theta$ , which is trained using denoising score matching. Given training pairs  $(x, y) \sim P_{\text{data}}$ , latent representations are computed as  $z = \mathcal{E}(x)$ . We then construct a noisy version of the latent variable via

$$\tilde{z} = z + \sigma_i \epsilon, \quad \epsilon \sim \mathcal{N}(0, I),$$

which implies the conditional distribution

$$\tilde{z} \mid z \sim \mathcal{N}(z, \sigma_i^2 I).$$

The network is trained to predict the score of this perturbed distribution. This is achieved by minimizing the denoising score-matching objective

$$\theta^* = \arg \min_{\theta} \frac{1}{2L} \sum_{i=1}^L \mathbb{E}_{(x,y) \sim P_{\text{data}}} \mathbb{E}_{\epsilon \sim \mathcal{N}(0,I)} [\|\sigma_i s_\theta(\tilde{z}, \sigma_i, y) + \epsilon\|_2^2], \quad (4.13)$$

where  $L$  denotes the number of discrete noise scales.

The objective can be understood by noting that the optimal denoising direction for Gaussian corruption is proportional to the added noise. In particular, for the perturbation model  $\tilde{z} = z + \sigma \epsilon$ , the score function satisfies

$$\nabla_{\tilde{z}} \log p(\tilde{z} \mid z) = -\frac{1}{\sigma^2}(\tilde{z} - z) = -\frac{1}{\sigma} \epsilon.$$

Therefore, the network is trained to recover the noise vector (up to scaling), which is equivalent to learning the score of the conditional distribution.

Training over multiple noise scales  $\{\sigma_i\}_{i=1}^L$  ensures that the model learns a consistent score field across different levels of corruption. Small values of  $\sigma$  correspond to fine-grained local structure in the latent space, while large values capture large-scale geometric properties of

the distribution. This multi-scale formulation is essential for stable and accurate reverse-time generation.

Overall, the diffusion model transforms probabilistic state estimation into a stochastic dynamical system in latent space. Starting from Gaussian noise, the learned reverse SDE progressively transports samples toward regions of high posterior probability under  $p(z \mid y)$ , thereby enabling principled conditional generation and uncertainty-aware state estimation.

#### 4.3.3.2 Numerical solution of the reverse-time SDE

The reverse-time SDE generally does not admit a closed-form analytical solution. Consequently, samples are generated by numerically integrating the reverse dynamics from the terminal time  $T$  to 0. In this work, the reverse SDE is solved using the Euler–Maruyama method, which is the stochastic analogue of the forward Euler method for ordinary differential equations.

Recall that the reverse-time SDE is given by

$$dz_\tau = [f(z_\tau, \tau) - g^2(\tau)\nabla_z \log p_\tau(z_\tau \mid y)] d\tau + g(\tau) d\bar{W}_\tau,$$

where the score function is approximated by the trained neural network

$$\nabla_z \log p_\tau(z_\tau \mid y) \approx s_\theta(z_\tau, \tau, y).$$

To numerically integrate the reverse process, the pseudo-time interval  $[0, T]$  is partitioned into  $N$  equally spaced time steps,

$$T = \tau_N > \tau_{N-1} > \cdots > \tau_0 = 0,$$

with time step

$$\Delta\tau = \frac{T}{N}.$$

Using the Euler–Maruyama discretization [KP77], we approximate the reverse SDE is with the recursive update

$$z_{k-1} = z_k - [f(z_k, \tau_k) - g^2(\tau_k)s_\theta(z_k, \tau_k, y)] \Delta\tau + g(\tau_k)\sqrt{\Delta\tau} \varepsilon_k, \quad (4.14)$$

where

$$\varepsilon_k \sim \mathcal{N}(0, I)$$

is an independent Gaussian random vector sampled at each time step.

The reverse-time integration is initialized by sampling  $z_N \sim P_T$ , where the terminal distribution is chosen as  $P_T = \mathcal{N}(0, I)$ . The Euler–Maruyama update is then applied sequentially for  $k = N, N-1, \dots, 1$  until  $\tau = 0$ , producing a sample

$$z_0 \sim p(z \mid y).$$

Finally, the decoder of the trained autoencoder maps the latent sample back to the original state space,

$$\hat{x} = \mathcal{D}(z_0),$$

yielding a probabilistic estimate of the system state. By repeating the reverse-time integration from different Gaussian initializations, multiple samples can be generated from the posterior distribution, enabling uncertainty quantification in the reconstructed state.

## 4.4 Ensemble Propagation and Assimilation of Measurements

Given a sensor measurement  $y^{(k)}$  at time  $t = t^{(k)}$ , the diffusion model provides samples from the conditional latent distribution  $p(z^{(k)} \mid y^{(k)})$ . While these samples capture the instant-

neous posterior associated with the current observation, they do not incorporate information from previous measurements. To address this limitation, we introduce a particle filtering framework in latent space that sequentially assimilates observations while leveraging the diffusion model during resampling to mitigate particle degeneracy.

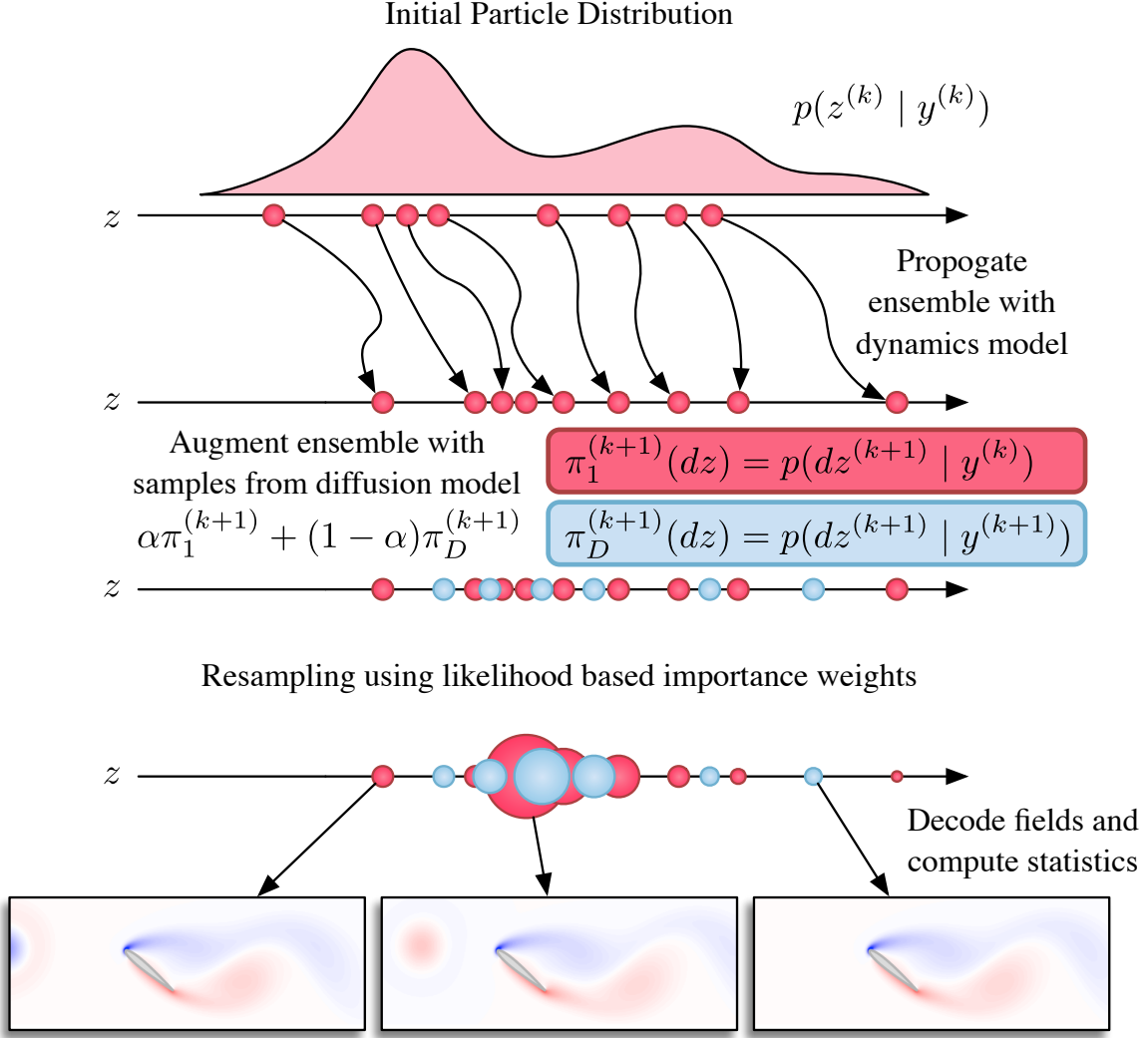


Figure 4.3: Illustration of the latent particle filter procedure. Red particles indicate samples from the previous measurement ensemble. Blue particles indicate samples from the diffusion model conditioned on the current measurement.

Suppose that at time step  $k$  the filtering distribution is represented by an ensemble of  $N$

particles with associated empirical measure,

$$\pi_2^{(k)} = \frac{1}{N} \sum_{i=1}^N \delta_{z_i^{(k)}}, \quad (4.15)$$

where  $\delta_z$  denotes the Dirac measure concentrated at  $z$ . This ensemble approximates the filtering distribution

$$\pi_2^{(k)}(dz^{(k)}) \approx P(dz^{(k)} \mid y^{(1:k)}), \quad (4.16)$$

where  $y^{(1:k)}$  denotes the collection of all observations up to time step  $k$ .

#### 4.4.1 Neural ODE Prediction Step

Each particle is propagated forward using a latent dynamics model. In this work we utilize a neural ODE to provide the forward prediction map [LG22, LZG23],

$$\frac{dz}{dt} = f_\theta(z). \quad (4.17)$$

The neural ODE is trained by fitting trajectories of latent states observed at discrete time instances. Given training data consisting of latent trajectories  $\{z(t_k)\}_{k=0}^K$ , the objective is to learn parameters  $\theta$  such that the induced flow map reproduces the observed evolution.

Direct backpropagation through the numerical ODE solver is computationally expensive in both memory and time, since it requires storing the full forward trajectory. Instead, we employ the adjoint sensitivity method [CRB18], which computes gradients by solving an augmented backward-in-time differential equation. Let the Neural ODE loss functional be defined as

$$\mathcal{L} = \sum_{k=0}^K \ell(z(t_k)),$$

where  $\ell(\cdot)$  is the reconstruction error. We define the adjoint variable

$$a(t) = \frac{\partial \mathcal{L}}{\partial z(t)}.$$

The evolution of the adjoint state is governed by the continuous-time ODE

$$\frac{da}{dt} = -a(t)^\top \frac{\partial f_\theta(z(t))}{\partial z}, \quad (4.18)$$

which is integrated backward in time from  $t = T$  to  $t = 0$ , with terminal condition determined by the gradient of the loss at observed data points. The gradient of the loss with respect to the model parameters  $\theta$  is given by

$$\frac{d\mathcal{L}}{d\theta} = - \int_T^0 a(t)^\top \frac{\partial f_\theta(z(t))}{\partial \theta} dt. \quad (4.19)$$

This expression shows that parameter updates depend on the sensitivity of the vector field  $f_\theta$  along the entire trajectory, weighted by the adjoint state.

When constructing the neural network for the right-hand side, we must be careful with our choice of architecture to ensure existence and uniqueness of solutions. We require that the learned vector field of the Neural ODE,

$$f_\theta : \mathbb{R}^m \rightarrow \mathbb{R}^m,$$

is continuously differentiable, which can be ensured by choosing smooth activation functions in the neural network. Common choices that we experimented with are the SiLU, Tanh, and LipSwish activations.

Since every continuously differentiable function is locally Lipschitz, the assumptions of the Picard–Lindelöf theorem are satisfied by  $f_\theta$ . Hence, for every initial condition  $z_0 \in \mathbb{R}^m$ ,



the initial value problem corresponding to the Neural ODE

$$\dot{z}(t) = f_\theta(z(t)), \quad z(t_0) = z_0,$$

admits a unique maximal solution on an interval containing  $t_0$ . Throughout this work, we assume that the Neural ODE architecture is constructed such that this maximal solution is defined on the integration interval  $[t_0, t_f]$ . A sufficient condition is that  $f_\theta$  be globally Lipschitz. This can be achieved by combining globally Lipschitz activation functions with appropriate constraints on the weight matrices, such as spectral normalization [MKK18]. However, the resulting Lipschitz constant may still be large, potentially leading to instabilities.

The solution defines a flow map

$$\Phi^{\Delta t} : \mathbb{R}^m \rightarrow \mathbb{R}^m, \quad \Phi^{\Delta t}(z_0) = z(t_0 + \Delta t),$$

where  $z(t)$  is the unique solution initialized at  $z(t_0) = z_0$ . Since the vector field is autonomous, the flow depends only on the elapsed time  $\Delta t$ . The flow satisfies the semigroup property

$$\Phi^{\Delta t_2} \circ \Phi^{\Delta t_1} = \Phi^{\Delta t_1 + \Delta t_2},$$

whenever the compositions are defined, together with

$$\Phi^0 = \text{Id}.$$

If the solution exists for both positive and negative times, then the flow is invertible and

$$(\Phi^{\Delta t})^{-1} = \Phi^{-\Delta t},$$

so that  $\Phi^{\Delta t}$  is a diffeomorphism.<sup>3</sup>

The latent dynamics are therefore represented by the discrete-time prediction model

$$z_i^{(k+1|-)} = \Phi^{\Delta t} \left( z_i^{(k)} \right) + \eta_i^{(k)},$$

where  $\eta_i^{(k)}$  denotes additive process noise. The predicted ensemble is therefore described by the empirical measure

$$\pi_1^{(k+1)} = \frac{1}{N} \sum_{i=1}^N \delta_{z_i^{(k+1|-)}}. \quad (4.20)$$

This measure approximates the predictive distribution

$$\pi_1^{(k+1)}(dz^{(k+1)}) \approx P(dz^{(k+1)} \mid y^{(1:k)}). \quad (4.21)$$

#### 4.4.2 Diffusion Augmentation

To mitigate particle degeneracy during sequential assimilation, we augment the predicted ensemble using samples drawn from the diffusion model conditioned on the newest observation  $y^{(k+1)}$ . Particle filters are known to suffer from sample impoverishment where repeated weighting and resampling can cause particles to collapse onto a low-dimensional sub-manifold of the state space [GSE95]. The diffusion-generated samples act as a rejuvenation mechanism, restoring particle diversity while remaining consistent with the current observation. Consider an  $M$  sample ensemble generated from the conditional diffusion model at time  $t^{(k+1)}$ :

$$\tilde{z}_j^{(k+1)} \sim p(z^{(k+1)} \mid y^{(k+1)}). \quad (4.22)$$

---

<sup>3</sup>Although the flow of a deterministic ODE is invertible whenever solutions exist in both time directions, this should not be confused with physical time-reversal symmetry. Dissipative systems remain mathematically invertible despite contracting trajectories, but the inverse flow is often numerically unstable. Moreover, genuinely irreversible dynamics, such as stochastic diffusion or other processes with information loss, require more careful treatment.

The corresponding empirical measure is

$$\pi_D^{(k+1)} = \frac{1}{M} \sum_{j=1}^M \delta_{\tilde{z}_j^{(k+1)}}. \quad (4.23)$$

We then construct a proposal distribution as a convex combination of the predictive ensemble and the diffusion ensemble,

$$Q^{(k+1)} = \alpha \pi_1^{(k+1)} + (1 - \alpha) \pi_D^{(k+1)}, \quad (4.24)$$

where  $\alpha \in [0, 1]$  controls the balance between dynamical consistency and diffusion-based rejuvenation.

#### 4.4.3 Importance Sampling

Since the predictive and proposal distributions are represented by empirical probability measures, they are finite sums of Dirac measures. Consequently, they are singular with respect to the Lebesgue measure, and thus the corresponding Radon–Nikodym derivatives with respect to the Lebesgue measure do not exist. To obtain continuous approximations suitable for importance sampling, we construct kernel density estimates (KDEs) from the particle ensembles. Given an empirical measure

$$\pi(dz) = \frac{1}{N} \sum_{i=1}^N \delta_{z_i}(dz), \quad (4.25)$$

the corresponding density estimate is defined as

$$\hat{p}(z) = \frac{1}{N} \sum_{i=1}^N K_h(z - z_i), \quad (4.26)$$

where  $K_h$  is a Gaussian kernel with bandwidth  $h$ . In this work, the bandwidth is chosen according to Silverman’s rule of thumb, although one may experiment with more elaborate

methods [Sil18]. Applying this procedure to the predictive measure  $\pi_1^{(k+1)}$  and the proposal measure  $Q^{(k+1)}$  yields continuous density approximations  $\hat{p}_{\text{pred}}^{(k+1)}$  and  $\hat{q}^{(k+1)}$ , respectively.

We assign particle sampled from the proposal distribution importance weights according to

$$w_i^{(k+1)} \propto p_\theta(y^{(k+1)} \mid z_i^{(k+1)}) \left[ \frac{\hat{p}_{\text{pred}}^{(k+1)}(z_i^{(k+1)})}{\hat{q}^{(k+1)}(z_i^{(k+1)})} \right]^\beta, \quad (4.27)$$

where  $p_\theta(y \mid z)$  denotes the Gaussian likelihood defined by the probabilistic decoder of the autoencoder and  $\beta \in [0, 1]$  is a tempering parameter. When  $\beta = 1$ , this corresponds to sequential importance sampling using the KDE-based approximations of the predictive and proposal densities. After normalization the filtered distribution is resampled from the weighted empirical measure

$$\pi_2^{(k+1)} = \sum_{i=1}^N \frac{w_i^{(k+1)}}{\sum_{\ell=1}^N w_\ell^{(k+1)}} \delta_{z_i^{(k+1)}}. \quad (4.28)$$

This measure approximates the posterior distribution

$$\pi_2^{(k+1)}(dz^{(k+1)}) \approx P(dz^{(k+1)} \mid y^{(1:k+1)}). \quad (4.29)$$

When  $\alpha = 1$ , the proposal measure reduces to the predictive ensemble alone, recovering the standard bootstrap particle filter [GSE95].

Finally, the latent particles are mapped back to the physical state space using the decoder,

$$x_i^{(k+1)} = \mathcal{D}(z_i^{(k+1)}).$$

The resulting ensemble can then be used to estimate posterior means, variances, confidence intervals, and other quantities of interest in the physical flow field.

## 4.5 Application to Laminar Gust-Airfoil Interaction

We consider the application of the proposed methodology to laminar vortex-gust interactions past a NACA 0012 airfoil at a chord-based Reynolds number,  $Re = u_\infty c / \nu = 100$ , where  $u_\infty$  is the free-stream velocity,  $c$  is the chord length, and  $\nu$  is the kinematic viscosity. The simulations are performed at a high-incidence angle of attack  $\alpha = 40^\circ$ . All simulations are performed using an incompressible flow solver [HI04, HMI06] over a computational domain that lies in  $(x/c, y/c) \in [-15, 30] \times [-20, 20]$  with the wing leading edge specified as the origin.

To generate the vortex disturbances, we parameterize the gusts by  $(x_0/c, y_0/c, R/c, G)$ , where  $x_0$  and  $y_0$  denote the gust initial center from the leading edge,  $R$  is the gust radius, and  $G = u_{\text{gust}}/u_\infty$  is the gust ratio. We generate 60 cases at random initial shedding phase sampled from the baseline trajectory and gust parameters in the set  $(x_0/c, y_0/c, R/c, G) \in \{-3\} \times [-0.5, 0.5] \times [0.25, 1] \times [-4, 4]$  with 1500 snapshots per case with a spacing of  $c\Delta t/u_\infty = 0.005$ . The gust takes the form of a Taylor vortex:

$$u_\theta = u_{\text{gust}} \frac{r}{R} \exp \left[ \frac{1}{2} \left( 1 - \frac{r^2}{R^2} \right) \right], \quad (4.30)$$

where  $r$  is the radius from the initial vortex center and  $u_{\text{gust}}$  is the maximum rotational velocity of the initialized gust disturbance. We place 8 equispaced pressure sensors along the entire airfoil surface starting at the leading edge, with additive Gaussian noise at 2% to 5% of the free-stream dynamic pressure value. An illustration of the simulation setup is shown in Figure 4.4(a).

The general autoencoder architecture and training procedure follow [TYT26], with the network architecture summarized in Table 4.1. The autoencoder comprises a convolutional encoder that maps the input flow field to the parameters of a Gaussian latent distribution, from which latent samples are generated using the reparameterization trick. A convolutional decoder reconstructs the flow field, while a secondary fully connected probabilistic decoder

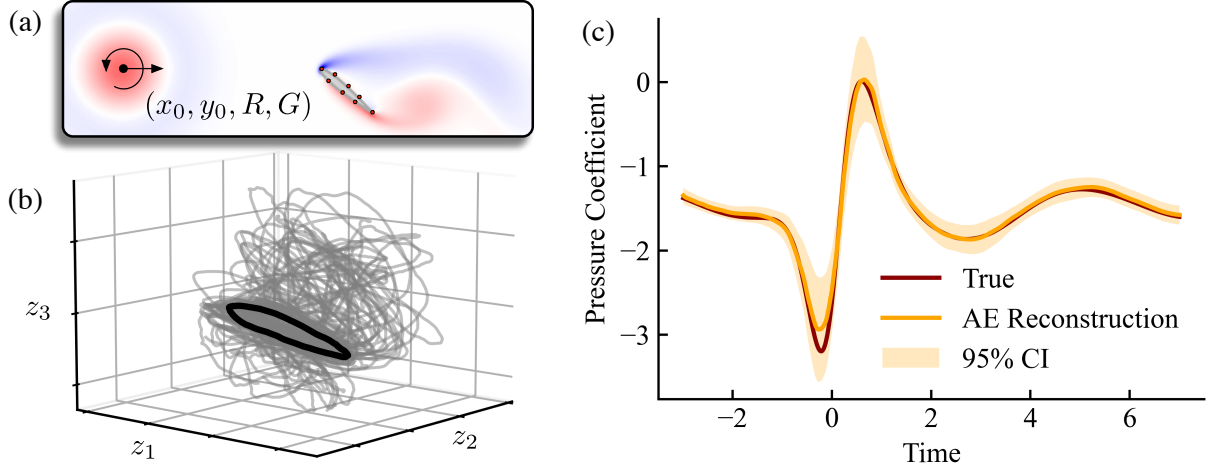


Figure 4.4: (a) Example gust-airfoil interaction. Pressure sensors marked in red. (b) Probabilistic Obs.-AE latent space. Baseline trajectory highlighted in black. (c) Example maximum likelihood estimation of leading edge sensor pressure coefficient from latent space.

predicts the mean and log-variance of the pressure sensor measurements. The states  $x \in \mathbb{R}^{400 \times 120}$  correspond to the discretized out-of-plane vorticity field  $\omega = \partial_{x_1} u_{x_2} - \partial_{x_2} u_{x_1}$ , while  $y \in \mathbb{R}^8$  denotes the noisy pressure sensor measurements.<sup>4</sup> Using an L-curve criterion, we selected a latent dimension of  $m = 8$  and rotated the latent space to align with its principal components.

The conditional latent distribution is subsequently modeled using a score-based diffusion model parameterized by a residual multilayer perceptron. The perturbed latent state is conditioned on both sinusoidal time embeddings and learned sensor embeddings before being processed through a stack of residual fully connected blocks with Layer Normalization and SiLU activation functions. The parameters for the SDM are shown in Table 4.2.

The latent Neural ODE is parameterized by a fully connected multilayer perceptron, each employing the SiLU activation function, which has a bounded first derivative. A learned linear shortcut is added to the nonlinear network output, allowing the model to efficiently

---

<sup>4</sup>Here we use  $(x_1, x_2, x_3)$  for the three spatial directions to distinguish them from  $x$  and  $y$  being the state and measurement variables.

capture both linear and nonlinear latent dynamics [LBT23]. The network architecture is summarized in Table 4.3. The autoencoder, SDM, and NODE are optimized using the Adam optimizer with weight decay [KB15] with automatic mixed-precision [MNA18] training via gradient scaling. During training and evaluation, we add noise to the pressure sensors at 5% of the free-stream pressure value.

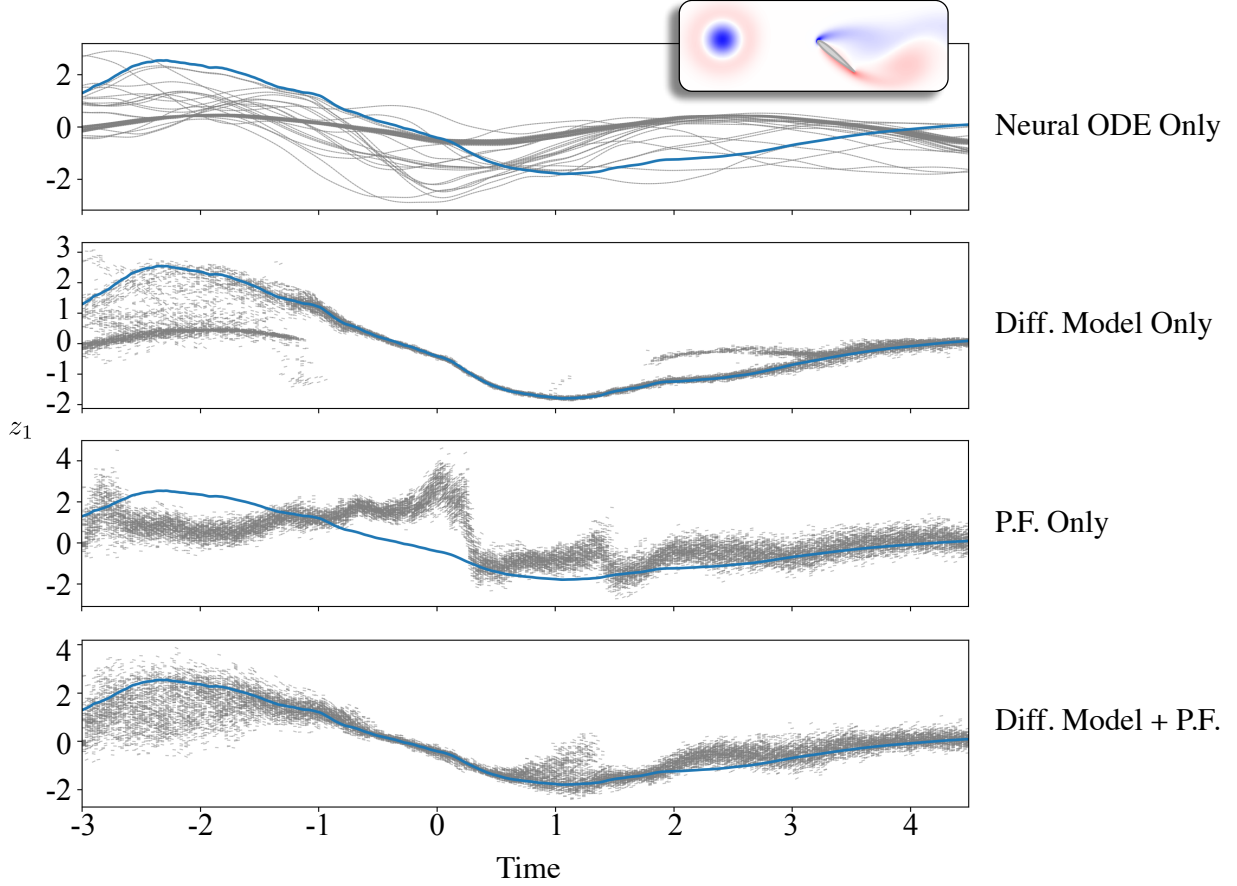


Figure 4.5: State estimation for the first latent coordinate  $z_1$  from noisy pressure sensor information for an example gust encounter. Blue lines show the true latent state trajectory and the gray lines show ensemble particles. The time  $t = 0$  corresponds to when the gust has convected a distance of  $|x_0|/c$ . The gust parameters are  $(x_0, y_0, R/c, G) = (-3, 0, 0.25, -2.5)$

Figure 4.4(b) shows the first three latent coordinates, with the undisturbed limit cycle (black) surrounded by gust-disturbed cases (gray). The latent representation also enables probabilistic estimation of the pressure measurements  $y$ . An example prediction of the

leading-edge pressure coefficient,

$$C_P = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty u_\infty^2},$$

against the convective time is shown in Figure 4.4(c). The autoencoder exhibits a normalized  $\ell_2$  estimation error around 20% for the vorticity field.

Figure 4.5 shows state estimation ensembles for the first latent coordinate during an example gust encounter. The initial ensemble is sampled from the diffusion model estimate of  $p(z^{(0)} | y^{(0)})$ , with noisy pressure measurements collected every  $(c\Delta t/u_\infty = 0.02)$  convective time. The initial distribution is skewed heavily towards the baseline flow, as seen with the sharp mode in trajectories starting near  $z_1 = 0$ , reflecting ambiguity between the undisturbed flow state and gust realizations located sufficiently far upstream, which are indistinguishable from measurement noise. In this sense, the incoming gust disturbances can be seen as a low-probability extreme event.

Propagating the initial ensemble with the Neural ODE only does not incorporate new sensor measurements, so every trajectory from the initial ensemble is simply evolved forward. As a result, the ensemble continues to represent both the undisturbed baseline flow and multiple possible incoming gust realizations, leading to large uncertainty throughout the trajectory.

The diffusion-only approach generates an instantaneous estimate of the latent state distribution directly from the incoming pressure measurements. This provides accurate estimates during the gust encounter. However, because each estimate is made independently, the state distribution before and after the interaction remains bimodal, reflecting the ambiguity between a distant gust and sensor noise in the baseline flow.

Using the standard bootstrap particle filter performs poorly for this problem. The filter rapidly collapses onto an incorrect mode early in the sequence due to the deterministic nature of the neural ODE, even with injected noise using a KDE approach as discussed in [GSE95]. Additionally, there is a visible bias towards the baseline undisturbed states near  $z_1 = 0$ . This



leads to the particle filter diverging from the true latent trajectory and preventing recovery once additional measurements become available.

In contrast, the proposed diffusion-augmented particle filter combines the strengths of both approaches. The diffusion model preserves the uncertainty associated with the unknown incoming disturbance, maintaining information about possible incoming disturbances, while the particle filter sequentially assimilates measurements over time. Consequently, the uncertainty contracts gradually as evidence accumulates and the gust approaches the airfoil, rather than collapsing abruptly at the onset of the gust encounter. This produces a state estimate that remains consistent with the true latent trajectory throughout the sequence.

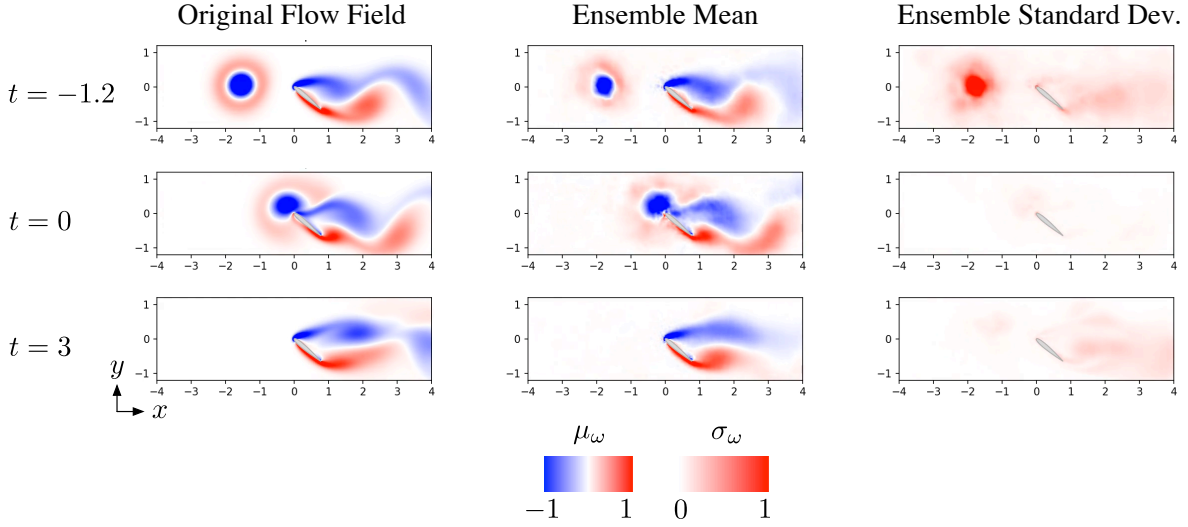


Figure 4.6: Statistics of estimated vorticity field ensemble for the diffusion-augmented particle filter compared to the original flow field.

Figure 4.6 shows the decoded vorticity field statistics at several times during the gust encounter for the diffusion-augmented particle filter ensemble. The reconstruction uncertainty directly reflects uncertainty in the latent state estimate. At early times, uncertainty is concentrated in the size and strength of the incoming gust disturbance. This is expected, as when the gust is significantly far away from the airfoil, the influence of the gust on the sensors cannot be distinguished from sensor noise. Thus, as was seen in the latent state

estimation, the conditional distribution  $p(x | y)$  corresponds to both the undisturbed flow in the presence of noise, and different potential upstream vortical disturbances. When the gust reaches the airfoil, the ensemble mean closely matches the true flow field and the uncertainty is substantially reduced in the flow field state. After the gust convects downstream, a moderate level of uncertainty remains in the wake structures.

Figure 4.7 compares the reconstructed flow fields obtained using the four estimation methods. For the neural ODE prediction shown in Figure 4.7(a), the ensemble exhibits uncertainty corresponding to the possible presence of an incoming gust. However, the ensemble mean remains strongly biased toward the baseline flow because no measurement assimilation is performed as the gust approaches. Additionally, we can see indicators in the standard deviation field of gust encounters that have been propagated from the initial ensemble, which are not consistent with the current measurement, such as the gust core at  $(x, y) \approx (-0.5, -0.5)$ . The diffusion model prediction in Figure 4.7(b) captures the incoming disturbance more accurately, although considerable uncertainty remains in both the gust position and strength. In contrast, the particle filter alone, shown in Figure 4.7(c), produces an erroneous estimate of the gust and wake. The particle filter solution rapidly diverges to an unphysical flow field, resulting in a highly distorted ensemble mean. Finally, the diffusion model-augmented particle filter in Figure 4.7(d) provides the most accurate reconstruction, with the ensemble mean closely matching the reference flow field. The remaining uncertainty is primarily concentrated near the core of the incoming gust, indicating that the estimator has successfully localized the disturbance estimate.

## 4.6 Conclusions

State estimation in fluid mechanics remains challenging due to the high dimensionality of flow fields, nonlinear dynamics, sparse observations, and the presence of strongly non-Gaussian uncertainty. In this work, we proposed a reduced-order Bayesian state estimation framework

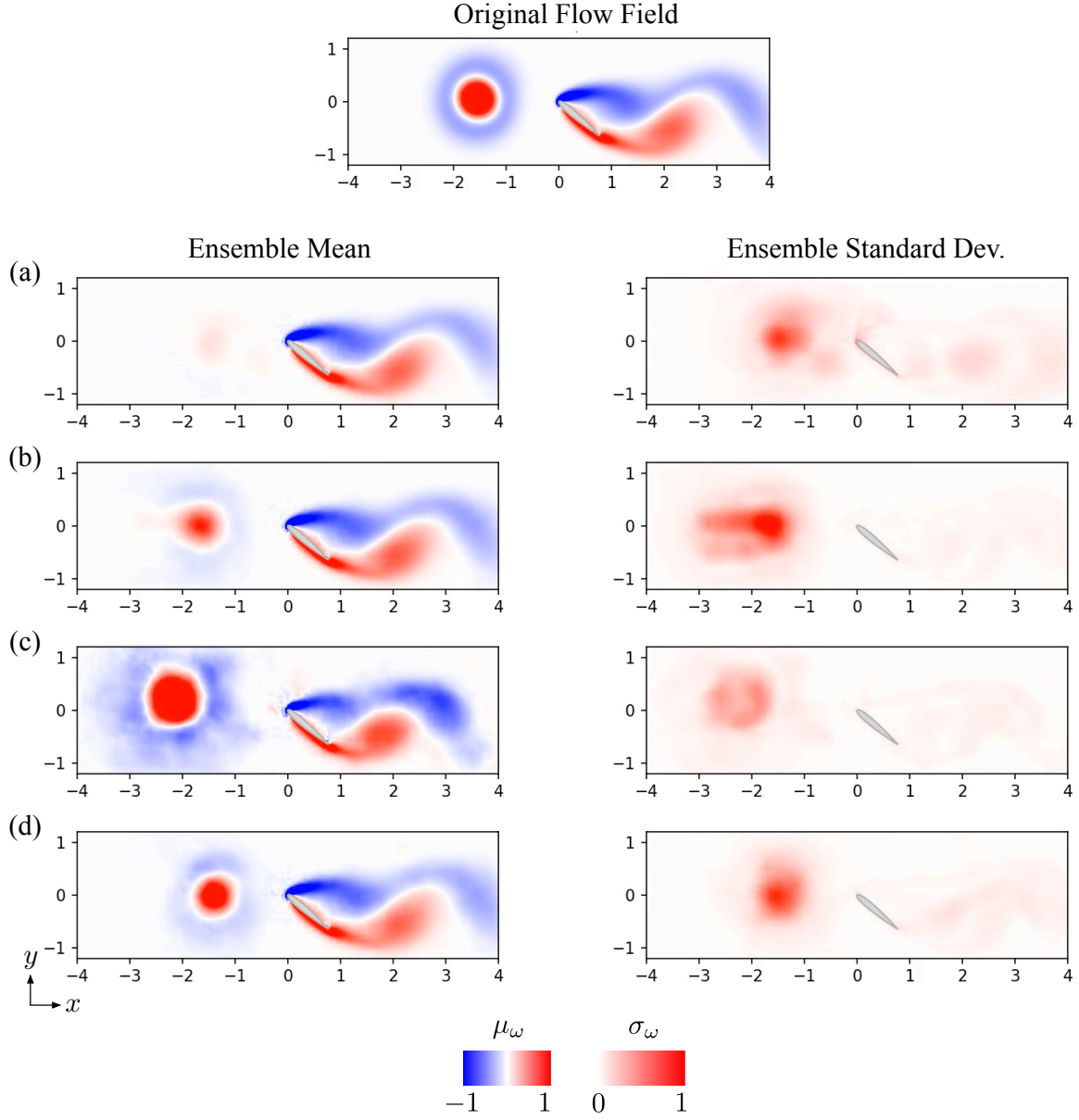


Figure 4.7: Comparison of reconstruction of different methods. Shown are ensemble statistics of the estimated vorticity field predicted by (a) the neural ODE, (b) the diffusion model, (c) the particle filter, and (d) the diffusion model–augmented particle filter, compared against the reference flow field at convective time  $tU_\infty/c = -1.2$ . The gust parameters are  $(x_0, y_0, R/c, G) = (-3, 0, 0.25, 2.5)$ .

that combines modern generative models with the classical particle filtering paradigm. A variational autoencoder provides a compact latent representation of the flow field, a conditional score-based diffusion model learns the conditional latent distribution from noisy sensor measurements, and a latent Neural ODE provides an efficient dynamical model for propagating particles between observations. These learned models are integrated within a bootstrap particle filter, enabling sequential Bayesian inference to be performed entirely in a low-dimensional latent space.

The proposed diffusion-augmented particle filter combines the complementary strengths of generative modeling and traditional sequential Monte Carlo methods. The diffusion model provides expressive, multimodal probabilistic estimates that accurately capture uncertainty from sparse measurements, while the particle filter incorporates temporal information through recursive Bayesian updates. Unlike a conventional particle filter, which is prone to particle impoverishment and mode collapse in deterministic nonlinear systems, the diffusion model continually replenishes the particle ensemble with physically plausible hypotheses, allowing competing flow realizations to be maintained until sufficient observational evidence becomes available. This results in more robust state estimation throughout the gust encounter and improved recovery of the underlying flow field.

Although demonstrated here for gust–airfoil interactions, the proposed framework is general and may be applied to a broad class of nonlinear dynamical systems where accurate probabilistic state estimation from sparse observations is required. More broadly, this work illustrates how modern generative models can complement rather than replace established Bayesian filtering techniques, providing a practical pathway toward scalable, uncertainty-aware data assimilation for complex fluid flows. While we adopted a simple strategy for integrating the diffusion model with particle filtering, more sophisticated multiple importance sampling techniques could provide a more principled and efficient framework for their combination.

| Encoder                           |                | Field Decoder       |               |
|-----------------------------------|----------------|---------------------|---------------|
| Layer                             | Data Size      | Layer               | Data Size     |
| Input $\omega$                    | (400, 120, 1)  | Dense               | (128)         |
| Conv(3×3,16) + GN                 | (400, 120, 16) | Dense               | (5600)        |
| MaxPool                           | (200, 60, 16)  | Reshape             | (25, 7, 32)   |
| Conv(3,3,24) + GN                 | (200, 60, 24)  | Conv(3,3,32) + GN   | (25, 7, 32)   |
| Conv(3,3,24) + GN                 | (200, 60, 24)  | Conv(3,3,32) + GN   | (25, 7, 32)   |
| MaxPool                           | (100, 30, 24)  | Upsample (nearest)  | (50, 15, 32)  |
| Conv(3,3,32) + GN                 | (100, 30, 32)  | Conv(3,3,32) + GN   | (50, 15, 32)  |
| Conv(3,3,32) + GN                 | (100, 30, 32)  | Conv(3,3,24) + GN   | (50, 15, 24)  |
| MaxPool                           | (50, 15, 32)   | Upsample (bilinear) | (100, 30, 24) |
| Conv(3,3,32) + GN                 | (50, 15, 32)   | Conv(3,3,24) + GN   | (100, 30, 24) |
| Conv(3,3,32) + GN                 | (50, 15, 32)   | Conv(3,3,16) + GN   | (100, 30, 16) |
| MaxPool                           | (25, 7, 32)    | Upsample (bilinear) | (200, 60, 16) |
| Flatten                           | (5600)         | Conv(3,3,16) + GN   | (200, 60, 16) |
| Dense                             | (256)          | Conv(3,3,1)         | (200, 60, 1)  |
| Dense $(\mu_z, \log \sigma_z^2)$  | (16)           | Output $\omega$     | (400, 120, 1) |
| Probabilistic Decoder             |                |                     |               |
| Layer                             |                | Data Size           |               |
| Latent vector                     |                | (128)               |               |
| Dense + BN                        |                | (256)               |               |
| Dense + BN                        |                | (256)               |               |
| Dense + BN                        |                | (256)               |               |
| Dense + BN                        |                | (256)               |               |
| Output $(\mu_y, \log \sigma_y^2)$ |                | $(2N_{sens})$       |               |

Table 4.1: Network architecture of the probabilistic autoencoder. The encoder predicts the latent Gaussian parameters  $(\mu_z, \log \sigma_z^2)$ , from which the latent vector is sampled using the reparameterization trick. The field decoder reconstructs the input field, while the probabilistic decoder predicts the mean and log-variance of the sensor measurements. GN denotes Group Normalization [WH18] and BN denotes Batch Normalization [IS15]. All hidden layers use the SiLU activation function,  $\text{SiLU}(x) = x \sigma(x) = \frac{x}{1+e^{-x}}$ .

| Score-Based Diffusion Model                                                            |           |
|----------------------------------------------------------------------------------------|-----------|
| Layer                                                                                  | Data Size |
| <b>Pseudo-Time Embedding</b>                                                           |           |
| Input $\tau$                                                                           | (1)       |
| $[\tau - \frac{1}{2}, \cos(2\pi\tau), \sin(2\pi\tau), \cos(4\pi\tau), \sin(4\pi\tau)]$ | (5)       |
| Linear + SiLU                                                                          | (256)     |
| Linear                                                                                 | (128)     |
| <b>Sensor Embedding</b>                                                                |           |
| Input $y$                                                                              | (8)       |
| Linear + SiLU                                                                          | (256)     |
| Linear                                                                                 | (128)     |
| <b>Score Network</b>                                                                   |           |
| Input latent state $x_\tau$                                                            | (8)       |
| Concatenate $(x_\tau, \tau_{emb}, y_{emb})$                                            | (264)     |
| Linear + SiLU                                                                          | (256)     |
| Residual Block ( $\times 4$ )                                                          | (256)     |
| LayerNorm                                                                              | (256)     |
| Linear + SiLU                                                                          | (256)     |
| Linear                                                                                 | (256)     |
| Skip connection + SiLU                                                                 | (256)     |
| Concatenate $(h, \tau_{emb}, y_{emb})$                                                 | (512)     |
| Linear + SiLU                                                                          | (256)     |
| Linear                                                                                 | (8)       |
| Output (Score)                                                                         | (8)       |

Table 4.2: Network architecture of the score-based diffusion model (SDM). Independent embedding networks first encode the diffusion time and sensor observations. These embeddings are concatenated with the perturbed latent state before being processed by a residual multilayer perceptron consisting of four residual blocks. The final output is the estimated score of the perturbed latent state. All hidden layers employ the SiLU activation function.

| <b>Latent Neural ODE</b>         |           |
|----------------------------------|-----------|
| Layer                            | Data Size |
| Input latent state $h$           | (8)       |
| Linear + SiLU                    | (256)     |
| Linear + SiLU                    | (256)     |
| Linear + SiLU                    | (256)     |
| Linear + SiLU                    | (256)     |
| Linear                           | (8)       |
| Output $\dot{h} = f_{\theta}(h)$ | (8)       |

Table 4.3: Network architecture of the latent Neural ODE. A linear shortcut connection from the input layer is added to the network output to improve optimization and enable efficient learning of near-linear latent dynamics.

## CHAPTER 5

### Concluding Remarks and Future Directions

#### 5.1 Summary of Contributions

This thesis has examined the use of data-driven dimensionality reduction techniques for the modeling, analysis, and control of fluid flows, with a particular emphasis on aerodynamic applications. Motivated by the high-dimensionality and nonlinear, multiscale nature of the governing equations, we have argued throughout for the value of reduced-order representations that retain the essential dynamical and/or geometric structure of a flow while discarding redundant complexity. Chapter 1 laid the theoretical groundwork for this perspective, situating the manifold hypothesis, proper orthogonal decomposition, and autoencoder-based nonlinear embeddings within the broader context of data-driven reduced-order modeling.

Building on this foundation, the thesis developed three distinct but conceptually related applications of learned low-dimensional representations. In Chapter 2, we introduced an autoencoder-based procedure for aerodynamic design optimization of industrial automobile geometries, demonstrating that nonlinear latent representations can efficiently capture shape variation, thereby accelerating design exploration over high-dimensional geometric parameter spaces. Optimization of an example geometry showed an 11% reduction in the drag coefficient. Notably, the resulting optimized designs achieved aerodynamic performance within 2% of experimentally validated, high-fidelity large-eddy simulations (LES), demonstrating that the learned reduced-order representation was sufficiently faithful to the underlying flow physics to support reliable design decisions rather than serving as a purely qualitative sur-



rogate.

In Chapter 3, we developed an approach based on optimal transport distances to learn a latent geometry with greater physical interpretability, and applied this framework to assess control performance in separated flows over an airfoil subject to thermal actuation. In this work, we endow the latent space with a meaningful geometric structure by enforcing the autoencoder to learn a minimum distortion embedding. By associating the latent coordinates with an optimal transport structure rather than treating them as an arbitrary reparameterization of the original flow fields, distances in the latent space could be directly related to physically interpretable changes in the flow, making the representation potentially useful for analysis, control assessment, and other downstream applications rather than reconstruction alone. The proposed approach was applied to separated flows over a NACA 0012 airfoil subject to thermal actuation at the leading edge. With the optimal transport-based latent space, we were able to succinctly identify control regimes corresponding to modifications of the laminar separation bubble size, as well as control inputs that resulted in partial and global laminarization of the leading edge shear layer.

In Chapter 4, we considered the problem of probabilistic state estimation and data assimilation for gust encounters, combining a latent diffusion model with a particle filtering framework. We applied the proposed methodology to the probabilistic state estimation of a high-incidence angle of attack NACA 0012 airfoil experiencing large-amplitude discrete gust disturbances. The proposed method demonstrated that modern generative modeling techniques can be integrated with classical, interpretable Bayesian state estimation machinery, yielding a framework that combines the expressiveness and scalability of deep generative models with the mathematical rigor and probabilistic consistency of traditional importance sampling methods. The proposed method demonstrated significant improvements in the estimation of the latent space trajectory in comparison to the traditional bootstrap particle filter.

Together, these three studies illustrate the versatility of learned low-dimensional repre-

sentations across a range of tasks spanning design, control assessment, and state estimation. A unifying theme across the applications presented in this thesis is that purely data-driven methodologies become substantially more interpretable, reliable, and practically useful when they are grounded in physical insight and mathematical structure, rather than being employed as unconstrained black-box function approximators. While modern learning-based approaches are capable of achieving impressive predictive performance, their effectiveness in scientific and engineering contexts is ultimately determined not only by accuracy, but also by the extent to which they respect and encode the underlying structure of the governing system.

Across all investigated applications, a recurring conclusion is that the design of the latent representation plays a central role in downstream model performance and interpretability. In particular, latent spaces constructed with explicit or implicit awareness of the underlying physics or mathematical structure consistently yield more robust and meaningful representations than those optimized solely for reconstruction error. Such structured representations not only improve numerical stability and generalization, but also enable a clearer segregation of physically relevant flow features from spurious variability. This observation is especially significant in downstream tasks such as estimation, control, and reduced-order modeling, where the objective is not merely to reproduce observed data, but to extract representations that support meaningful physical predictions of fluid systems for downstream application. In this sense, the latent space becomes an active design choice rather than a passive intermediate layer.

Overall, the results of this thesis reinforce the view that the most effective integration of machine learning into fluid mechanics arises from hybrid approaches that combine data-driven flexibility with principled physical and mathematical constraints. Such approaches are particularly powerful in complex flow regimes, where nonlinear interactions, multiscale dynamics, and instability mechanisms must be faithfully represented to enable reliable prediction, control, and reduced-order modeling of fluid systems.

## 5.2 Future Directions

The methodologies developed in this thesis suggest several potential avenues for future research in data-driven dimensionality reduction for aerodynamic applications. Although the present work has focused primarily on developing latent representations for reduced-order modeling and state estimation, the resulting framework naturally lends itself to broader questions concerning flow prediction, control, and physically informed representation learning. The following directions outline several opportunities for extending these ideas to increasingly complex flow regimes and more general classes of dynamical systems.

### 5.2.1 State Estimation for Highly Unsteady Aerodynamic Disturbances

A natural direction would be the extension of the present frameworks toward the state estimation and mitigation of extreme gust encounters and other rare-event unsteady aerodynamic phenomena. A primary objective in this context would be the development of low-dimensional models capable of accurately characterizing the aerodynamic response of lifting surfaces in a unified manner across both discrete gust events and continuously varying disturbances. Such reduced-order representations would provide compact yet physically meaningful descriptions of dominant flow dynamics, enabling efficient prediction and estimation of transient aerodynamic loads.

Beyond the limiting cases of isolated and persistent gust disturbances, it would also be of interest to investigate intermediate flow regimes characterized by intermittent forcing. These regimes are particularly relevant in realistic atmospheric conditions, where gust structures exhibit variability across multiple spatial and temporal scales. Studying the transition between discrete and continuous forcing could reveal shared dynamical structures that may be exploited for modeling and estimation purposes, while also clarifying the extent to which latent representations generalize across different disturbance classes rather than being tailored to a specific regime.

A further extension would involve the incorporation of probabilistic latent dynamics into the reduced-order modeling framework. Rather than assuming deterministic latent evolution, neural stochastic differential equations could be employed to learn the intrinsic process noise associated with unresolved flow dynamics and external disturbances. Such formulations would enable uncertainty to be propagated directly within the latent space, providing probabilistic state estimates that are likely to be more robust under realistic operating conditions.

### 5.2.2 Latent-Space Flow Control

Following the establishment of reliable state estimation capabilities, a natural progression would involve the development of control strategies aimed at mitigating aerodynamic disturbances and improving performance objectives. The central idea would be to leverage learned low-dimensional representations to construct computationally efficient control laws. In this setting, the latent dynamics would serve as a surrogate model for the evolution of the full flow field, enabling control design to be performed in a reduced-order space rather than the original high-dimensional system. This paradigm corresponds to reduced-order control, where both estimation and actuation are mediated through a learned manifold representation of the flow.

Within a simulation environment, several classes of actuation mechanisms could be considered. These include localized blowing and suction jets, which may be represented either as momentum forcing terms in the governing equations or through boundary condition modifications. More general surface-based actuation strategies could also be incorporated, including spatially distributed thermal actuation or wall-normal stress forcing, depending on the physical configuration and modeling assumptions.

On the algorithmic side, multiple control methodologies could be explored in conjunction with learned latent representations. These include model predictive control (MPC), in which finite-horizon optimal control problems are solved directly in the reduced latent

space; reinforcement learning (RL), in which control policies are trained to optimize long-term performance through interaction with the simulated environment; and optimal control approaches based on learned dynamics models, in which adjoint-based or gradient-based optimization is performed on the latent evolution equations. Across these approaches, a central consideration would be the trade-off between control performance, analytical tractability, and computational efficiency while maintaining consistency with the underlying flow physics.

### 5.2.3 Geometric Machine Learning for Reduced-Order Modeling

Another avenue for future investigation concerns the incorporation of additional differential geometric structure into learned latent representations. In the present work, the latent space is treated as a Euclidean space, implicitly assuming that linear interpolation and Euclidean distance provide meaningful measures of similarity. However, for nonlinear dynamical systems, the latent representation may instead reside on a curved manifold, for which Euclidean metrics fail to capture the intrinsic geometry of the state space. Geometric machine learning provides a principled framework for addressing this limitation by incorporating manifold-aware concepts, including geodesic distances, curvature, and constrained optimization on nonlinear spaces, into the construction of reduced-order models. Such formulations may yield latent representations that better preserve the underlying geometry and dynamical structure of complex flow systems.

Embedding such geometric constraints into neural latent representations could enable autoencoders to learn coordinates that more faithfully preserve the underlying dynamical structure of complex flows, particularly in regimes where aerodynamic responses exhibit strongly nonlinear behavior.

In addition, the identification of coherent structures and other dynamically invariant solutions represents a complementary direction for physics-informed representation learning. Exact coherent structures have been shown to organize turbulent and transitional flows by acting as dynamically significant solutions embedded within the high-dimensional state

space. Developing geometric autoencoder frameworks capable of identifying latent variables associated with these structures could provide reduced-order representations that are not only compact but also directly connected to the fundamental mechanisms governing flow evolution. Rather than learning purely statistical correlations, such representations would seek to recover the intrinsic dynamical organization of the flow, potentially improving extrapolation, interpretability, and control performance.

#### 5.2.4 Algebraic Topology for Representation Learning of Fluids

A further research direction concerns the incorporation of computational algebraic topology or topological data analysis (TDA) into data-driven dimensionality reduction. Whereas geometric machine learning seeks to preserve local manifold structure, topological methods instead characterize the global organization of data by identifying invariant features that persist across multiple spatial or temporal scales. Persistent homology, one of the principal tools in TDA, provides a systematic framework for identifying connected components, loops, voids, and higher-dimensional topological features that remain robust under continuous deformation and are relatively insensitive to measurement noise. Such descriptors offer complementary information to conventional latent variable models by capturing qualitative structural properties of the underlying dynamical system. While persistent homology has found application in several fluid mechanics problems, there has yet to be a systematic effort to formally integrate topological descriptors into modern modeling and reduced-order representations of fluid flows.

These ideas are particularly well suited to graph-based representations of fluid flows. In random vortex methods and other Lagrangian particle formulations, the flow field is naturally represented by interacting collections of vortex particles whose relationships are more appropriately described through graphs than structured grids. Similar graph-based descriptions arise in vortex networks, swimmer interactions, and particle methods used throughout computational fluid dynamics. Preserving the evolving connectivity and interaction structure

of these systems presents a challenge for conventional dimensionality reduction techniques, which typically rely upon Euclidean vector representations.

One possible direction would therefore be the integration of persistent homology with graph neural networks and geometric autoencoders to construct topology-aware latent representations. Topological invariants could be incorporated either as additional features describing the evolving flow state or directly into the optimization objective through topology-preserving regularization. Such approaches may encourage learned representations that preserve not only local geometric relationships but also the global organization of coherent vortical structures throughout their evolution. In turn, this may improve robustness, interpretability, and generalization across flow conditions exhibiting substantial changes in topology.

Beyond providing a compact representation of coherent structures, suitably defined topological descriptors may also admit a mean-field interpretation in vortex dynamics. In the point-vortex formulation of the incompressible Euler equations, the vorticity field is approximated by a collection of interacting vortices whose empirical measure converges, under suitable assumptions, to the continuum vorticity distribution as the number of vortices tends to infinity. Such convergence does not, by itself, imply convergence of arbitrary persistence diagrams. However, when persistent homology is constructed from a filtration that depends continuously on the underlying measure in a topology compatible with the mean-field limit, it may be possible to use stability results to establish convergence of the corresponding persistence diagrams. Under these additional assumptions, persistent topological features could possibly be interpreted as multiscale descriptors of the limiting mean-field dynamics, rather than as properties specific to individual vortex realizations.

This viewpoint suggests that persistence diagrams may capture the evolution of coherent structures that survive averaging over vortex interactions. Long-lived homology classes correspond to robust vortical regions and recirculation structures, while short-lived features are more likely associated with numerical noise or transient small-scale vortices. Consequently,

topological summaries provide a natural bridge between particle-based descriptions of the flow and continuum representations, offering descriptors that remain meaningful across different discretizations and ensemble realizations.

Although these connections remain largely theoretical, they motivate the use of persistent homology as an additional state descriptor for probabilistic flow estimation. In particular, incorporating topological summaries into latent-state inference may help distinguish physically plausible vortex configurations from spurious reconstructions that possess similar pointwise errors but fundamentally different coherent structures.

More broadly, combining computational topology with geometric machine learning offers the prospect of reduced-order models that simultaneously preserve local differential structure and global topological organization. Such representations may prove particularly valuable for highly nonlinear, graph-structured dynamical systems where both geometry and connectivity play fundamental roles in governing the underlying flow physics.

### **5.2.5 Optimal Transport on Manifolds for Fluid Mechanics**

A promising extension of the framework developed in this thesis concerns the use of unbalanced optimal transport on manifolds. Two related notions of a manifold arise in this setting. First, the physical domain may possess non-Euclidean geometry, as occurs for atmospheric and oceanic flows on the sphere, flows over curved aerodynamic surfaces, turbomachinery configurations, and periodic or multiply connected domains. Second, even when the physical domain is Euclidean, the collection of dynamically realizable flow states may occupy a curved, low-dimensional manifold within the high-dimensional state space. Optimal transport provides a natural connection between these two forms of geometry by defining distances between fluid observables using the intrinsic geometry of the physical domain.

In the case where the physical domain is some Riemannian manifold, it may be advantageous to define the transport cost in terms of geodesic distances on the manifold. This



may be important for vortices propagating on spherical or curved domains, rotating waves in annular geometries, traveling structures in periodic flows, and surface-supported quantities associated with boundary layers or wall actuation.

This geometry may be particularly useful for reduced-order modeling of transport-dominated flows. Linear reduced-order models often require many modes to represent structures that translate, rotate, or convect through the domain, even when the underlying motion is governed by only a few variables. A manifold-aware transport model could instead represent the displacement of coherent structures directly while using additional coordinates to describe their deformation and intensity. This may yield compact representations of vortex shedding, separated shear layers, gust-induced structures, rotating stall, and interacting vortices on periodic or spherical domains.

Optimal transport on manifolds can also be combined naturally with autoencoder-based manifold learning. Consider an encoder and decoder

$$E : q \mapsto z, \quad D : z \mapsto \hat{q}(z), \quad (5.1)$$

where  $q$  is a high-dimensional fluid state and  $z \in \mathbb{R}^r$  contains its reduced latent coordinates. A nonnegative fluid observable may be constructed through a map  $R$ ,

$$\mu(z) = R(D(z)) \in \mathcal{M}_+(M), \quad (5.2)$$

where  $\mathcal{M}_+(M)$  denotes the space of finite nonnegative measures on  $M$ . Depending on the application,  $\mu$  could represent kinetic-energy density, enstrophy density, scalar concentration, or another physically relevant quantity.

For an unbalanced optimal transport formulation that admits a Riemannian geodesic structure, such as the Wasserstein–Fisher–Rao metric, the geometry of the decoded states can be pulled back through the decoder to define a state-dependent Riemannian metric on

the latent coordinates. Locally, this metric satisfies

$$d_{\text{WFR}}^2(\mu(z), \mu(z + dz)) = dz^\top G(z) dz + o(\|dz\|^2), \quad (5.3)$$

with components

$$G_{ij}(z) = \left\langle \frac{\partial \mu}{\partial z_i}, \frac{\partial \mu}{\partial z_j} \right\rangle_{\text{WFR}, \mu(z)}. \quad (5.4)$$

Consequently, the infinitesimal length of a latent displacement is

$$ds^2 = dz^\top G(z) dz. \quad (5.5)$$

Applications of conventional autoencoders typically represent and operate on latent coordinates using Euclidean geometry, despite the non-Euclidean geometry that may be induced by the nonlinear decoder. The proposed formulation would make this geometry explicit by pulling back an unbalanced optimal transport metric onto the latent coordinates. Equal coordinate displacements may therefore represent different amounts of coherent-structure motion or intensity variation in different flow regimes. Geodesics of  $G(z)$  would provide intrinsic paths constrained to the learned family of realizable states rather than straight-line interpolations through the latent space. This construction would extend the minimum-distortion embedding introduced in Chapter 3. Rather than requiring a Euclidean latent space to reproduce the global pairwise transport geometry, a future model could seek to preserve the local unbalanced optimal transport metric.

Continuous symmetries provide a particularly compelling application and could potentially be identified directly from flow data rather than specified a priori. Many fluid systems contain translational or rotational degrees of freedom associated with traveling vortices, periodic shedding, and rotating coherent structures. A reduced representation could separate a symmetry coordinate  $\theta$  from coordinates  $\eta$  describing the intrinsic structure remaining after

translation or rotation,

$$z = (\theta, \eta). \quad (5.6)$$

A data-driven model could seek transformations  $g_\theta$  that preserve the set of admissible flow states, satisfy the group composition law, and commute approximately with the learned dynamics. Optimal transport could identify the displacement associated with the resulting group orbit, while the remaining coordinates describe changes in strength, scale, spacing, and deformation. This admits a quotient-space interpretation in which symmetry phase and intrinsic structure are represented separately, while the off-diagonal components of  $G(z)$  retain coupling between them.

A further question is whether the discovered symmetries are associated with conserved or slowly varying physical quantities. For ideal fluid models possessing a variational formulation, continuous symmetries may be connected through Noether’s theorem to conservation laws involving energy, momentum, angular momentum, or circulation. In viscous, forced, or wall-bounded flows, these quantities would generally satisfy balance laws rather than remain exactly conserved. Comparing learned symmetry generators with known conservation or balance laws could therefore provide a means of assessing whether the latent representation has recovered physically meaningful structure. This suggests a broader research direction combining symmetry discovery, optimal transport geometry, and structure-preserving reduced-order dynamics.

The resulting geometry could support downstream state estimation and control. Forecast and observation errors could be evaluated using intrinsic geodesic distances rather than Euclidean coordinate differences, reducing the tendency to treat displaced but otherwise similar structures as unrelated states. Likewise, a controller could minimize the geodesic distance from the current flow to a desired set of attached or laminarized states. This would provide an objective based on the transport and modification of coherent structures rather than an arbitrary quadratic penalty on latent coordinates.

In these examples, unbalanced transport geometry should be interpreted as a metric for organizing flow states rather than as a replacement for the governing fluid dynamics. In particular, a Navier–Stokes trajectory need not follow a geodesic of the learned metric. The physical dynamics could instead be enforced separately through nonintrusive latent evolution equations, projection-based models, or physics-based residual constraints. Within this interpretation, unbalanced optimal transport on manifolds offers a possible framework for combining transport-aware model reduction, continuous symmetry reduction, manifold learning, state estimation, and flow control in a common physically interpretable latent representation.

### 5.2.6 Toward Unified Physics-Informed Latent Representations

Taken together, the developments presented in this thesis point toward a unified perspective on data-driven modeling in fluid mechanics, in which dimensionality reduction, representation learning, and downstream application are viewed as interconnected components of a single framework rather than isolated tasks. Across aerodynamic design optimization, flow control, and probabilistic state estimation, the central role of the latent representation emerges consistently as the determining factor that governs not only computational efficiency, but also physical interpretability and predictive reliability.

Future progress will likely depend on embedding increasingly rich mathematical structure into learned representations. Differential geometry provides tools for preserving the intrinsic manifold structure of nonlinear dynamical systems, while computational topology offers complementary information concerning their global organization and connectivity. Probabilistic formulations further enable uncertainty to be represented directly within the latent dynamics, allowing reduced-order models to remain robust under stochastic forcing and incomplete observations. Collectively, these developments suggest a pathway toward latent representations that are simultaneously compact, interpretable, physically consistent, and capable of generalizing across a broad range of aerodynamic phenomena.

The results of this thesis therefore suggest that future advances in data-driven fluid mechanics will likely depend less on increasing model complexity in the ambient high-dimensional state space, and more on developing principled latent representations in which the essential geometric, topological, and physical structure of the governing dynamics is made explicit.

## APPENDIX A

### Probability Theory Basics for Optimal Transport

We begin with the notion of a measurable space. A measurable space is a pair  $(\mathcal{X}, \mathcal{A})$ , where  $\mathcal{X}$  is a set and  $\mathcal{A} \subseteq 2^{\mathcal{X}}$  is a  $\sigma$ -algebra, with  $2^{\mathcal{X}}$  being the power set of  $\mathcal{X}$ . That is,  $\mathcal{A}$  is a collection of subsets of  $\mathcal{X}$  satisfying<sup>1</sup>

- $\emptyset \in \mathcal{A}$ ,
- if  $A \in \mathcal{A}$ , then  $A^c \in \mathcal{A}$ ,
- if  $\{A_i\}_{i=1}^{\infty} \subseteq \mathcal{A}$ , then  $\bigcup_{i=1}^{\infty} A_i \in \mathcal{A}$ .

A *measure* on  $(\mathcal{X}, \mathcal{A})$  is a then set function which assigns a numerical value to the “measurable” sets of  $\mathcal{X}$ .

$$\pi : \mathcal{A} \rightarrow \mathbb{R}_{\geq 0} \cup \{\infty\}$$

such that:

- $\pi(\emptyset) = 0$ ,
- (countable additivity) if  $\{A_i\}_{i=1}^{\infty} \subset \mathcal{A}$  are pairwise disjoint, then

$$\pi\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \pi(A_i).$$

---

<sup>1</sup>From these properties it follows that  $\mathcal{A}$  is also closed under countable intersections.

The triple  $(\mathcal{X}, \mathcal{A}, \pi)$  is called a *measure space*. When  $\pi(\mathcal{X}) = 1$ , the measure is called a *probability measure*, and  $(\mathcal{X}, \mathcal{A}, \pi)$  is a *probability space*. Intuitively, a measure assigns a notion of “size” or “mass” to measurable subsets of  $\mathcal{X}$ . In physical terms, if  $\mathcal{X}$  represents a spatial domain,  $\pi(A)$  can be interpreted as the amount of mass, energy, or probability contained in the region  $A \subseteq \mathcal{X}$ .

A *measurable function* (or random variable in probability theory) is a mapping  $f : \mathcal{X} \rightarrow \mathbb{R}$  such that the preimage  $f^{-1}(B) \in \mathcal{A}$  for every Borel set  $B \subseteq \mathbb{R}$ .<sup>2</sup> This ensures that integrals of the form  $\int_{\mathcal{X}} f d\pi$  are well-defined.

In the context of fluid mechanics, under certain assumptions, a flow field can be interpreted as a measurable function or, more generally, as a measure-valued distribution over a spatial domain. This provides a natural framework for comparing flow configurations using optimal transport distances between probability measures representing quantities such as mass density, energy density, or other physically relevant observables.

To further refine the notion of distributions, it is useful to introduce the concept of *signed measures*. A signed measure on a measurable space  $(\mathcal{X}, \mathcal{A})$  is a set function

$$\pi : \mathcal{A} \rightarrow \mathbb{R}$$

that is countably additive, but unlike standard measures, is allowed to take negative values. Signed measures naturally arise in fluid mechanics (e.g., velocity or vorticity).

A central result from measure theory used throughout this work is the Radon–Nikodym theorem, which provides a way of representing probability measures by densities with respect to a reference measure. Recall that a measure  $\pi$  is said to be *absolutely continuous* with

---

<sup>2</sup>A *Borel set* is any element of the Borel  $\sigma$ -algebra on  $\mathbb{R}$ , denoted  $\mathcal{B}(\mathbb{R})$ , which is the smallest  $\sigma$ -algebra containing all open subsets of  $\mathbb{R}$ . Equivalently, it is obtained from open sets through countable unions, countable intersections, and complements.

respect to a measure  $\pi_0$ , denoted  $\pi \ll \pi_0$ , if

$$\pi_0(A) = 0 \implies \pi(A) = 0, \quad \forall A \in \mathcal{A}.$$

**Theorem A.0.1** (Radon–Nikodym). *Let  $(\mathcal{X}, \mathcal{A})$  be a measurable space, and let  $\pi$  and  $\pi_0$  be  $\sigma$ -finite measures on  $(\mathcal{X}, \mathcal{A})$ . If  $\pi \ll \pi_0$ , then there exists a measurable function  $p : \mathcal{X} \rightarrow \mathbb{R}$  such that*

$$\pi(A) = \int_A p d\pi_0, \quad \forall A \in \mathcal{A}.$$

*Moreover, the function  $p$  is unique up to  $\pi_0$ -almost everywhere equality. It is called the Radon–Nikodym derivative of  $\pi$  with respect to  $\pi_0$  and is denoted by*

$$p = \frac{d\pi}{d\pi_0}.$$

The Radon–Nikodym derivative may be interpreted as the density of  $\pi$  relative to the reference measure  $\pi_0$ . In particular, when  $\pi_0$  is the Lebesgue measure on  $\mathbb{R}^n$  and  $\pi(\mathcal{X}) = 1$ ,  $\frac{d\pi}{d\pi_0}$  coincides with the usual probability density function.



## APPENDIX B

### OT Autoencoder Model Additional Details

In the OT based analysis in Chapter 3, we compare the learned latent spaces for two autoencoder architectures: a standard autoencoder that only uses reconstruction loss ( $\mathcal{L}_1$  loss only) and the OT-based autoencoder ( $\mathcal{L}_1$  and  $\mathcal{L}_2$  loss terms). We note that both models can qualitatively reconstruct the flow fields at similar levels of accuracy. We find that the inclusion of the OT-based embedding term does not have any noticeable impacts on the training process in practice, as depicted by the example loss curves in Figure B.1.

During the parameter study, we find that when increasing the latent space dimension the reduction in the loss slowed after  $n = 2$ , prompting the use of a two-dimensional latent space. Additionally, we find that for higher-dimensional latent spaces, while the regular autoencoder produced mostly arbitrary representations of the latent geometry, the OT-based autoencoder consistently produced the same two-dimensional structure as seen in Figure B.2 and B.3 for  $\alpha = 9^\circ$  and  $6^\circ$ , respectively.

To illustrate the difference between the OT-based distance and the  $L^2$  distance we plot pairwise UOT-based distances ( $\tilde{d}_{\text{field}}$ ) vs. pairwise  $L^2$  distances for discretized flow fields normalized by the maximum distance in the dataset in Figure B.4. Similar to the case with the previous example of the advecting and diffusing Gaussian in Figure 3.2, while the distances appear correlated, the  $L^2$  distance appears to saturate at around half the maximum value of  $\tilde{d}_{\text{field}}$  after the separation bubble has been suppressed. This means that the UOT distance is capable of more effectively distinguishing cases as the separation bubble is suppressed. This intuitively manifests as a change in curvature, in which most cases are

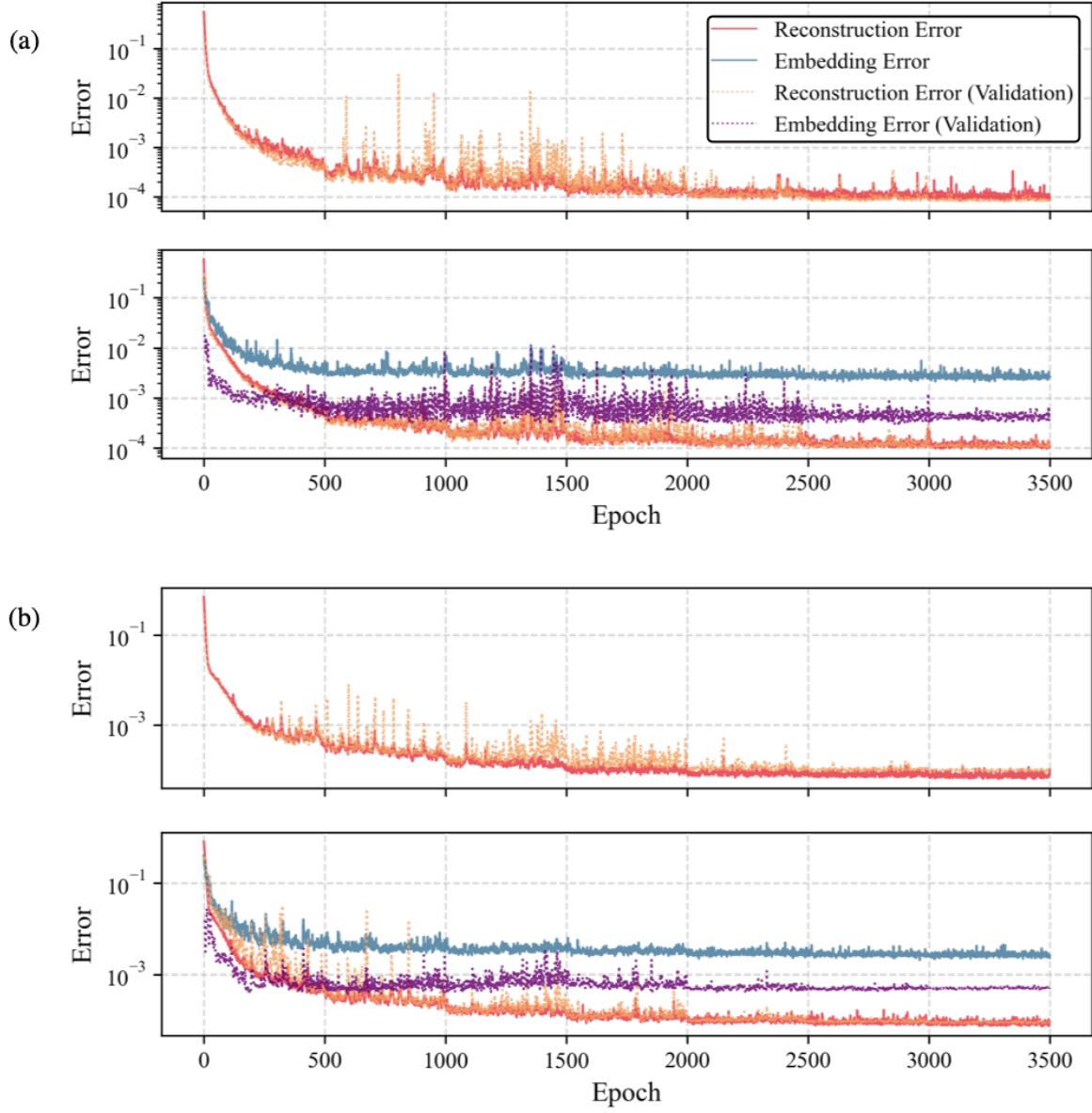


Figure B.1: Plot of loss curves for (a)  $\alpha = 9^\circ$  and (b)  $\alpha = 6^\circ$  cases of both regular (top) and OT-based autoencoders (bottom).

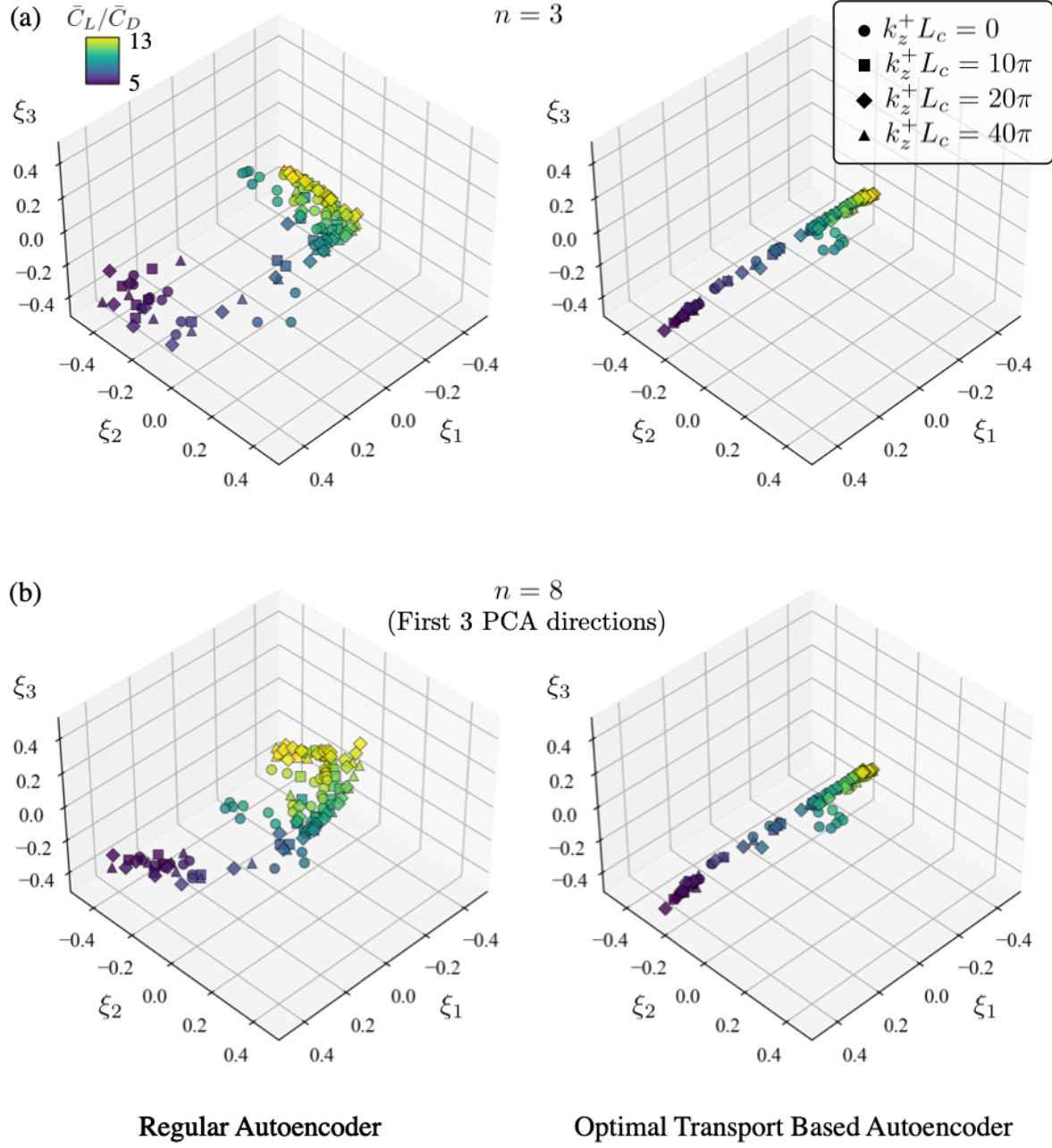


Figure B.2: Plot of  $\alpha = 9^\circ$  latent space for both regular and OT-based autoencoder with (a)  $n = 3$  and (b)  $n = 8$  (showing the first 3 principal axes).

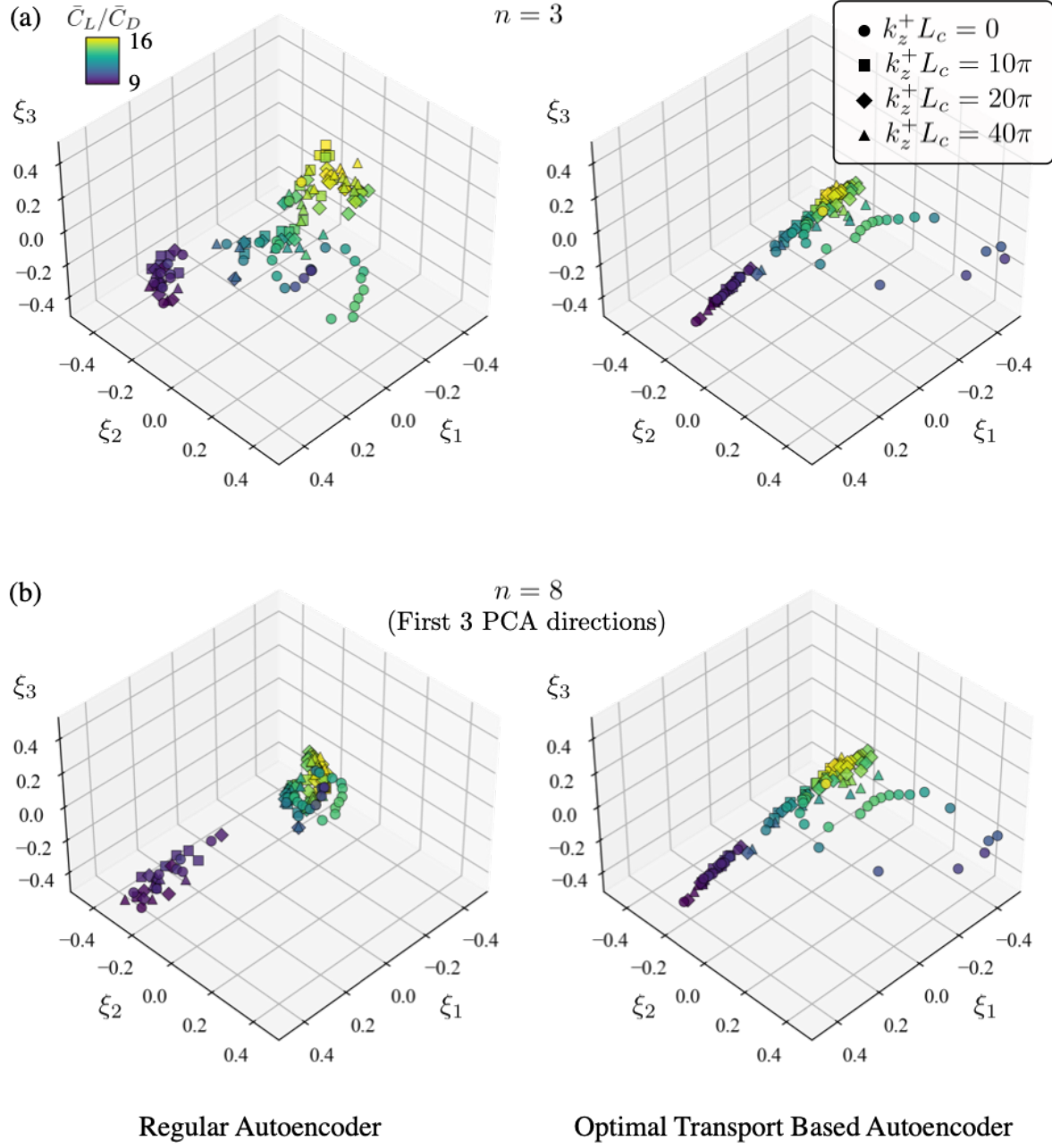


Figure B.3: Plot of  $\alpha = 6^\circ$  latent space for both regular and OT-based autoencoder with (a)  $n = 3$  and (b)  $n = 8$  (showing the first 3 principal axes).

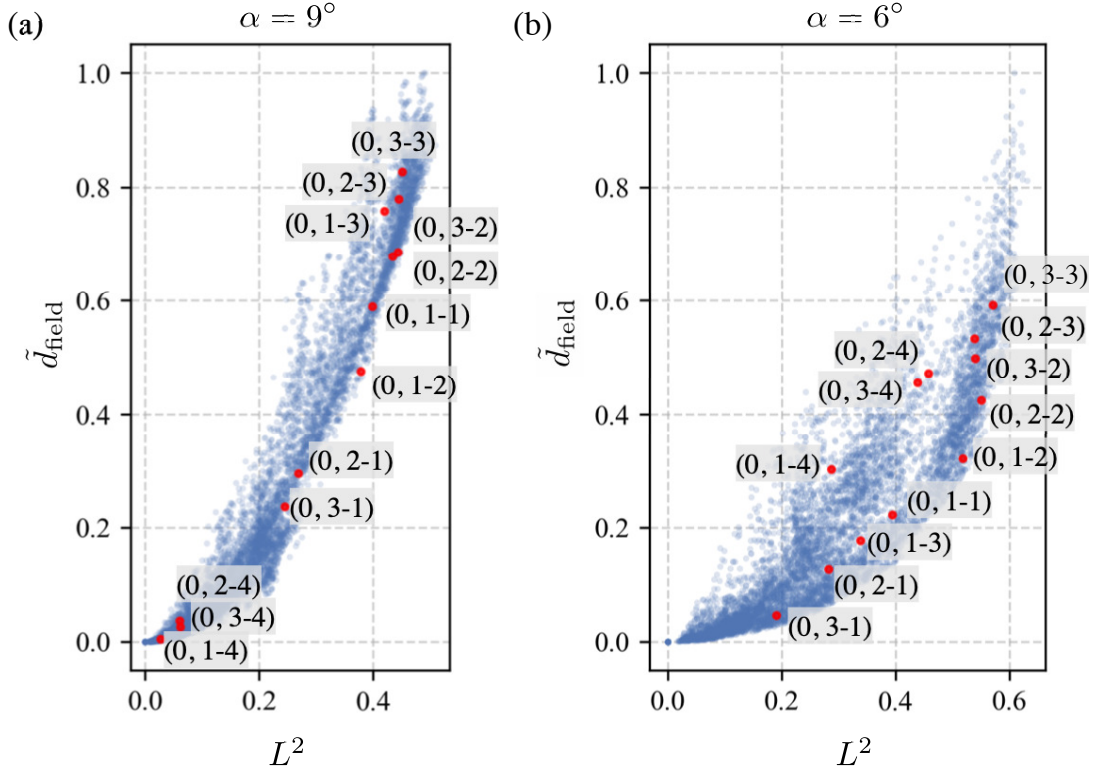


Figure B.4: Plot of pairwise UOT-based distances ( $\tilde{d}_{\text{field}}$ ) vs. pairwise  $L^2$  distances for discretized flow fields for (a)  $\alpha = 9^\circ$  and (b)  $\alpha = 6^\circ$ . Both axes have been normalized by the maximum pairwise distance. Red points denote example pairwise distances from the baseline flow corresponding to the cases in Figure 3.8 and Figure 3.10.

flattened along the  $\xi_1$  axis in the OT-AE latent space.

## APPENDIX C

### Positive Definiteness of the OT-Based Flow Field Dissimilarity

For completeness, this appendix verifies the theoretical properties of the optimal transport-based flow field dissimilarity introduced in Chapter 3. The proposed flow field dissimilarity is constructed by applying entropically regularised Sinkhorn divergences to the positive and negative components of signed flow fields. While the properties of the Sinkhorn divergence itself are well established in the literature, the resulting properties of the complete flow field dissimilarity, including the signed decomposition and multi-variable aggregation used here, are not explicitly derived for this particular construction. The following analysis therefore provides a direct verification that these standard properties are preserved in the proposed formulation, and that the optimal transport-based flow field dissimilarity serves as a valid semimetric over our flow fields.<sup>1</sup>

The analysis proceeds in three stages. First, we recall the relevant properties of the Sinkhorn divergence and identify the assumptions required on the transport cost and associated kernel. Second, we verify these assumptions for the Euclidean transport cost on a compact domain, being symmetry, Lipschitz continuity, and universality of the associated kernel. Finally, we show that separating signed fields into their positive and negative measures and combining the resulting componentwise discrepancies preserves definiteness of the proposed flow field dissimilarity.

---

<sup>1</sup>We use the term *semimetric* rather than metric because, in general, the Sinkhorn divergence does not satisfy the triangle inequality, although certain unregularized optimal transport formulations do.

Recall that, for positive measures  $m_1, m_2 \in \mathcal{M}_+(\mathcal{X})$ , the Sinkhorn divergence is defined as

$$S^\varepsilon(m_1, m_2) = d_{\text{UOT}}^\varepsilon(m_1, m_2) - \frac{1}{2} \left[ d_{\text{UOT}}^\varepsilon(m_1, m_1) + d_{\text{UOT}}^\varepsilon(m_2, m_2) \right] + \frac{\varepsilon}{2} \left( \int_{\mathcal{X}} dm_1 - \int_{\mathcal{X}} dm_2 \right)^2. \quad (\text{C.1})$$

The properties of the Sinkhorn divergence depend on the regularity of the transport cost and the universality of the associated kernel. The following result, established as Theorem 5 in [SFV19], provides the conditions under which the entropically regularised Sinkhorn divergence defines a valid discrepancy measure.

**Theorem C.0.1** (Positive definiteness and convexity of the Sinkhorn divergence [SFV19]). *Assume that the cost function  $C$  is symmetric and  $\gamma$ -Lipschitz continuous, and that the associated kernel*

$$k_\varepsilon(x, y) = \exp \left( -\frac{C(x, y)}{\varepsilon} \right)$$

*is a positive universal<sup>2</sup> kernel. Then, for any  $\varepsilon > 0$  and any entropy function  $\varphi$ , the Sinkhorn divergence  $S^\varepsilon(\alpha, \beta)$  is positive definite, symmetric, and convex in each argument (but not jointly convex).*

Consequently, the validity of the Sinkhorn divergence as a discrepancy measure reduces to verifying the assumptions of Theorem C.0.1 for the transport cost employed in this work. The remainder of this section establishes these properties for the Euclidean cost.

---

<sup>2</sup>A kernel  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{C}$  is *positive definite* if, for every  $n \in \mathbb{N}$ , every choice of points  $x_1, \dots, x_n \in \mathcal{X}$ , and every choice of coefficients  $c_1, \dots, c_n \in \mathbb{C}$ ,

$$\sum_{i,j=1}^n c_i \overline{c_j} k(x_i, x_j) \geq 0.$$

It is said to be *universal* if its associated reproducing kernel Hilbert space (RKHS), equivalently the closure of the linear span of the kernel sections  $k(\cdot, x)$ , is dense in the space of continuous functions on a compact domain with respect to the uniform norm.

## C.1 Verification of the assumptions of Theorem C.0.1

We verify that the Euclidean distance cost function

$$C(x, y) = \|x - y\|_2,$$

defined on a compact domain  $\mathcal{X} \subset \mathbb{R}^d$ , satisfies the assumptions required by Theorem C.0.1.

**Proposition C.1.1** (The Euclidean cost satisfies Theorem C.0.1). *Let  $\mathcal{X} \subset \mathbb{R}^d$  be a compact domain and let*

$$C(x, y) = \|x - y\|_2, \quad x, y \in \mathcal{X}.$$

*Then  $C$  is symmetric and  $\gamma$ -Lipschitz continuous on  $\mathcal{X} \times \mathcal{X}$  for  $\gamma = 2\sqrt{d}$ , and the associated kernel*

$$k_\varepsilon(x, y) = \exp\left(-\frac{\|x - y\|_2}{\varepsilon}\right), \quad \varepsilon > 0,$$

*is a positive universal kernel on  $\mathcal{X}$ . Consequently, by Theorem C.0.1, the Sinkhorn divergence  $S^\varepsilon(\alpha, \beta)$  associated with the cost  $C$  is positive definite, symmetric, and convex in each argument.*

*Proof.* We verify the two hypotheses of Theorem C.0.1 in turn: symmetry and Lipschitz continuity of  $C$  (Lemma C.1.1), and positivity and universality of the kernel  $k_\varepsilon$  (Lemmas C.1.2–C.1.4). Given these, the conclusion follows directly from Theorem C.0.1.  $\square$

### C.1.1 Symmetry and Lipschitz continuity

**Lemma C.1.1** (Symmetry and Lipschitz continuity of the Euclidean cost). *The cost  $C(x, y) = \|x - y\|_2$  is symmetric and  $\gamma$ -Lipschitz continuous on  $\mathcal{X} \times \mathcal{X}$  with respect to the  $\ell^\infty$  norm, with constant  $\gamma = 2\sqrt{d}$ .*

*Proof.* The cost is clearly symmetric since  $C(x, y) = \|x - y\|_2 = \|y - x\|_2 = C(y, x)$ .



To establish Lipschitz continuity with respect to the  $\ell^\infty$  norm on  $\mathcal{X} \times \mathcal{X}$ , let  $(x, y), (x', y') \in \mathcal{X} \times \mathcal{X}$ . Using the reverse triangle inequality,

$$\begin{aligned} |C(x, y) - C(x', y')| &= |\|x - y\|_2 - \|x' - y'\|_2| \\ &\leq \|(x - y) - (x' - y')\|_2 \\ &= \|(x - x') - (y - y')\|_2 \\ &\leq \|x - x'\|_2 + \|y - y'\|_2. \end{aligned}$$

Since all norms are equivalent on  $\mathbb{R}^d$ ,

$$\|z\|_2 \leq \sqrt{d} \|z\|_\infty,$$

and therefore

$$|C(x, y) - C(x', y')| \leq 2\sqrt{d} \max\{\|x - x'\|_\infty, \|y - y'\|_\infty\}.$$

Since the product space  $\mathcal{X} \times \mathcal{X}$  is equipped with the  $\ell^\infty$  norm, the previous inequality becomes

$$|C(x, y) - C(x', y')| \leq 2\sqrt{d} \|(x, y) - (x', y')\|_\infty.$$

Therefore, there exists a finite constant  $\gamma = 2\sqrt{d}$  such that

$$|C(x, y) - C(x', y')| \leq \gamma \|(x, y) - (x', y')\|_\infty$$

for all  $(x, y), (x', y') \in \mathcal{X} \times \mathcal{X}$ . Hence,  $C$  is  $\gamma$ -Lipschitz continuous on  $\mathcal{X} \times \mathcal{X}$ . □

### C.1.2 Properties of the associated kernel

The remaining condition of Theorem C.0.1 concerns the kernel induced by the transport cost. For the Euclidean distance cost, this kernel takes the form

$$k_\varepsilon(x, y) = \exp\left(-\frac{\|x - y\|_2}{\varepsilon}\right), \quad \varepsilon > 0.$$

Note that unlike the usual Laplacian kernel, the norm in the exponent is the  $\ell^2$  norm rather than the  $\ell^1$  norm. However, the procedure to demonstrate that it is a positive universal kernel follows standard arguments from the literature of reproducing kernel Hilbert spaces. To verify that this defines a positive universal kernel, we will use general results for translation-invariant kernels. We note that  $k_\varepsilon(x, y) = \exp(-\|x - y\|_2/\varepsilon) = \psi(x - y)$  with  $\psi(x) = e^{-\|x\|_2/\varepsilon}$  shows directly that  $k_\varepsilon$  is translation invariant.

To analyze the positive definiteness and universality of this kernel, we use Bochner's theorem.

**Theorem C.1.1** (Bochner's theorem [SFL11]). *A function  $\psi \in C_b(\mathbb{R}^d)$  is positive definite on  $\mathbb{R}^d$  if and only if it is the Fourier transform of a finite non-negative Borel measure  $\Lambda$  on  $\mathbb{R}^d$ , namely,*

$$\psi(x) = \int_{\mathbb{R}^d} e^{-ix^\top \omega} d\Lambda(\omega), \quad x \in \mathbb{R}^d.$$

*The measure  $\Lambda$  is called the spectral measure of  $\psi$ .*

Since  $\psi(x) = e^{-\|x\|_2/\varepsilon} \in C_b(\mathbb{R}^d)$ , we can apply Theorem C.1.1 to  $\psi$  to establish positive definiteness. To invoke it, we first need the Fourier transform of  $\psi$ , which we now provide one derivation of.

**Lemma C.1.2** (Fourier transform of the Euclidean cost kernel). *For  $\psi(x) = \exp(-\|x\|_2/\varepsilon)$ , the Fourier transform*

$$\widehat{\psi}(\omega) = \int_{\mathbb{R}^d} e^{-i\omega^\top x} \psi(x) dx$$

is given by<sup>3</sup>

$$\widehat{\psi}(\omega) = \frac{2^d \pi^{(d-1)/2} \Gamma\left(\frac{d+1}{2}\right) \varepsilon^d}{(1 + \varepsilon^2 \|\omega\|_2^2)^{\frac{d+1}{2}}}.$$

*Proof.* It suffices to treat the case  $\varepsilon = 1$ , since the general case follows by the scaling property of the Fourier transform: if  $f(x) = \varphi(x/\varepsilon)$  then  $\widehat{f}(\omega) = \varepsilon^d \widehat{\varphi}(\varepsilon\omega)$ , applied to  $\varphi(x) = e^{-\|x\|_2^2}$ .

Consider the following representation of  $e^{-a}$  for  $a \geq 0$  as a Gaussian mixture [GR14]:

$$e^{-a} = \frac{2}{\sqrt{\pi}} \int_0^\infty \exp\left(-t^2 - \frac{a^2}{4t^2}\right) dt. \quad (*)$$

Applying  $(*)$  with  $a = \|x\|_2$  gives

$$e^{-\|x\|_2} = \frac{2}{\sqrt{\pi}} \int_0^\infty \exp\left(-t^2 - \frac{\|x\|_2^2}{4t^2}\right) dt,$$

which expresses the Euclidean cost kernel (at  $\varepsilon = 1$ ) as a mixture, over  $t$ , of isotropic Gaussians in  $x$  of width  $2t$ .

By Fubini's theorem (justified since the integrand is absolutely integrable),

$$\int_{\mathbb{R}^d} e^{-i\omega^\top x} e^{-\|x\|_2} dx = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-t^2} \int_{\mathbb{R}^d} \exp\left(-i\omega^\top x - \frac{\|x\|_2^2}{4t^2}\right) dx dt.$$

Substituting  $x = 2ty$  (so  $dx = (2t)^d dy$ , and  $\|x\|_2^2/(4t^2) = \|y\|_2^2$ ) turns the  $t$ -dependent Gaussian into a fixed, unit-variance one:

$$\int_{\mathbb{R}^d} \exp\left(-i\omega^\top x - \frac{\|x\|_2^2}{4t^2}\right) dx = (2t)^d \int_{\mathbb{R}^d} \exp(-2it\omega^\top y - \|y\|_2^2) dy.$$

---

<sup>3</sup>The Gamma function, which generalizes the factorial to non-integer arguments, is defined for  $s > 0$  by

$$\Gamma(s) = \int_0^\infty t^{s-1} e^{-t} dt.$$

In particular, for positive integers  $n$ ,  $\Gamma(n) = (n-1)!$ .

We now evaluate the Gaussian integral. Note that for any  $b \in \mathbb{R}^d$ ,

$$\int_{\mathbb{R}^d} e^{-ib^\top y - \|y\|_2^2} dy = \pi^{d/2} e^{-\|b\|_2^2/4},$$

which follows by completing the square coordinate-wise,  $-y_k^2 - ib_k y_k = -(y_k + ib_k/2)^2 - b_k^2/4$ , and using  $\int_{-\infty}^{\infty} e^{-(y+ic)^2} dy = \sqrt{\pi}$  for any real constant  $c$  (by shifting the contour from  $\mathbb{R} + ic$  to  $\mathbb{R}$ , invoking Cauchy's integral theorem). Taking  $b = 2t\omega$  gives

$$\int_{\mathbb{R}^d} \exp(-2it\omega^\top y - \|y\|_2^2) dy = \pi^{d/2} e^{-\|\omega\|_2^2 t^2}.$$

We have that,

$$\int_{\mathbb{R}^d} e^{-i\omega^\top x} e^{-\|x\|_2^2} dx = \frac{2}{\sqrt{\pi}} \int_0^\infty e^{-t^2} (2t)^d \pi^{d/2} e^{-\|\omega\|_2^2 t^2} dt = \frac{2^{d+1} \pi^{d/2}}{\sqrt{\pi}} \int_0^\infty t^d e^{-t^2(1+\|\omega\|_2^2)} dt.$$

Substituting  $u = t^2(1 + \|\omega\|_2^2)$  and using  $\int_0^\infty t^d e^{-ct^2} dt = \frac{1}{2} \Gamma\left(\frac{d+1}{2}\right) c^{-(d+1)/2}$  for  $c > 0$ , with  $c = 1 + \|\omega\|_2^2$ , yields

$$\int_{\mathbb{R}^d} e^{-i\omega^\top x} e^{-\|x\|_2^2} dx = 2^d \pi^{(d-1)/2} \Gamma\left(\frac{d+1}{2}\right) (1 + \|\omega\|_2^2)^{-\frac{d+1}{2}}.$$

Applying the scaling property recalled above with  $\varphi(x) = e^{-\|x\|_2^2}$  and  $\psi(x) = \varphi(x/\varepsilon)$  gives

$$\widehat{\psi}(\omega) = \varepsilon^d \widehat{\varphi}(\varepsilon\omega) = 2^d \pi^{(d-1)/2} \Gamma\left(\frac{d+1}{2}\right) \varepsilon^d (1 + \varepsilon^2 \|\omega\|_2^2)^{-\frac{d+1}{2}},$$

as claimed. □

**Lemma C.1.3** (Positive definiteness of the Euclidean cost kernel). *For every  $\varepsilon > 0$ , the kernel*

$$k_\varepsilon(x, y) = \exp\left(-\frac{\|x - y\|_2}{\varepsilon}\right)$$

*is positive definite.*

*Proof.* By Lemma C.1.2,

$$\widehat{\psi}(\omega) = \frac{2^d \pi^{(d-1)/2} \Gamma((d+1)/2) \varepsilon^d}{(1 + \varepsilon^2 \|\omega\|_2^2)^{(d+1)/2}} \geq 0$$

for every  $\omega \in \mathbb{R}^d$ .

Since  $\widehat{\psi} \in L^1(\mathbb{R}^d)$ , the Fourier inversion theorem gives

$$\psi(x) = \int_{\mathbb{R}^d} e^{-ix^\top \omega} (2\pi)^{-d} \widehat{\psi}(\omega) \, d\omega.$$

Moreover,  $\widehat{\psi}$  is continuous (and thus Borel measurable) and non-negative, so

$$d\Lambda(\omega) = (2\pi)^{-d} \widehat{\psi}(\omega) \, d\omega$$

defines a finite non-negative Borel measure on  $\mathbb{R}^d$ .

Therefore, by Bochner's theorem,  $\psi$  is positive definite. Since

$$k_\varepsilon(x, y) = \psi(x - y),$$

the kernel  $k_\varepsilon$  is positive definite. □

**Lemma C.1.4** (Universality of the Euclidean Laplacian kernel on compact sets). *Let  $\mathcal{X} \subset \mathbb{R}^d$  be compact. Then, for every  $\varepsilon > 0$ , the kernel  $k_\varepsilon$  is universal on  $\mathcal{X}$ .*

*Proof.* By Lemma C.1.3,  $k_\varepsilon$  is positive definite and its spectral measure  $\Lambda$  has density

$$\frac{d\Lambda}{d\omega}(\omega) = (2\pi)^{-d} \widehat{\psi}(\omega).$$

Lemma C.1.2 shows that this density is strictly positive on  $\mathbb{R}^d$ . Hence, for every nonempty open set  $U \subset \mathbb{R}^d$ ,

$$\Lambda(U) = (2\pi)^{-d} \int_U \widehat{\psi}(\omega) \, d\omega > 0.$$

Therefore,

$$\text{supp}(\Lambda) = \mathbb{R}^d.$$

As observed above,  $k_\varepsilon$  is translation invariant. Moreover,

$$\psi(x) = e^{-\|x\|_2/\varepsilon} \in C_0(\mathbb{R}^d),$$

so  $k_\varepsilon$  is a  $C_0$ -kernel on  $\mathbb{R}^d$ . From Proposition 5.6(iii) of Carmeli et al. [CVT10], since  $\text{supp}(\Lambda) = \mathbb{R}^d$ , we have that the associated RKHS  $\mathcal{H}_{k_\varepsilon}$  is dense in  $L^2(\mathbb{R}^d, \mu)$  for every probability measure  $\mu$ .

We also have that, for a  $C_0$ -kernel, density of its RKHS in  $L^p(\mathbb{R}^d, \mu)$  for every probability measure  $\mu$ , for some  $1 \leq p < \infty$ , is equivalent to density of the RKHS in  $C_0(\mathbb{R}^d)$  with respect to the uniform norm.<sup>4</sup> Applying this characterization with  $p = 2$  therefore gives

$$\overline{\mathcal{H}_{k_\varepsilon}}^{\|\cdot\|_\infty} = C_0(\mathbb{R}^d).$$

Hence  $k_\varepsilon$  is  $c_0$ -universal on  $\mathbb{R}^d$ .

It remains to pass this to the compact domain  $\mathcal{X}$ . Every  $f \in C(\mathcal{X})$  admits an extension  $F \in C_0(\mathbb{R}^d)$ . By  $c_0$ -universality, there exists a sequence  $h_n \in \mathcal{H}_{k_\varepsilon}$  such that

$$\|h_n - F\|_{C_0(\mathbb{R}^d)} := \sup_{x \in \mathbb{R}^d} |h_n(x) - F(x)| \rightarrow 0.$$

The restrictions  $h_n|_{\mathcal{X}}$  belong to the RKHS associated with the restricted kernel  $k_\varepsilon|_{\mathcal{X} \times \mathcal{X}}$ . Since

---

<sup>4</sup>The norm used here is the uniform norm on  $C_0(\mathbb{R}^d)$ , defined using the ordinary supremum over all  $x \in \mathbb{R}^d$ . This differs conceptually from the previously defined  $L^\infty$ -norm, which is the essential supremum with respect to a specified measure and therefore disregards sets of measure zero. For continuous functions on  $\mathbb{R}^d$ , with respect to Lebesgue measure, the two norms have the same value; however, the uniform norm is the natural norm for the definition of  $c_0$ -universality and does not depend on a choice of measure.

$F|_{\mathcal{X}} = f$ , we have

$$\begin{aligned}
\|h_n|_{\mathcal{X}} - f\|_{C(\mathcal{X})} &:= \sup_{x \in \mathcal{X}} |h_n(x) - f(x)| \\
&= \sup_{x \in \mathcal{X}} |h_n(x) - F(x)| \\
&\leq \sup_{x \in \mathbb{R}^d} |h_n(x) - F(x)| \\
&= \|h_n - F\|_{C_0(\mathbb{R}^d)} \rightarrow 0.
\end{aligned}$$

Thus the RKHS of  $k_\varepsilon|_{\mathcal{X} \times \mathcal{X}}$  is dense in  $C(\mathcal{X})$  with respect to the uniform norm. Therefore, the restricted kernel is universal on  $\mathcal{X}$ .  $\square$

This completes the proof of Proposition C.1.1.

## C.2 Extension to signed and multi-component flow fields

The preceding analysis establishes that the Sinkhorn divergence with the Euclidean transport cost satisfies the required definiteness properties for positive measures. The flow fields considered in this work, however, are signed quantities. To extend the discrepancy to signed fields, the positive and negative components of the corresponding signed measures are compared independently. Likewise, multi-component flow fields are compared by aggregating the discrepancies of their individual components.

The definiteness results for these extensions follow directly from the corresponding properties of the Sinkhorn divergence and are therefore largely straightforward. They are included here for completeness, as they justify the signed- and multi-component discrepancies used throughout the remainder of this work.

The flow field discrepancy introduced in Chapter 3 is defined by comparing the positive and negative components of the signed measures obtained by the Jordan decomposition independently:

$$d_{\text{field}}(V_1, V_2) = S^\varepsilon(m_1^+, m_2^+) + S^\varepsilon(m_1^-, m_2^-). \quad (\text{C.2})$$

The following proposition establishes that this construction preserves non-negativity and definiteness.

**Proposition C.2.1** (Positivity of the signed-field discrepancy). *Let  $\mathcal{X} \subset \mathbb{R}^d$  be a compact measurable domain, and let  $V_1, V_2 \in L^1(\mathcal{X})$  be scalar flow fields with corresponding signed measures*

$$\mathrm{d}m_i(x) = V_i(x) \mathrm{d}x, \quad i = 1, 2,$$

*and Jordan decompositions*

$$m_i = m_i^+ - m_i^-.$$

*Assume that the conditions of Theorem C.0.1 hold. Then*

$$d_{\text{field}}(V_1, V_2) \geq 0,$$

*and*

$$d_{\text{field}}(V_1, V_2) = 0 \iff V_1 = V_2 \text{ almost everywhere on } \mathcal{X},$$

*where  $d_{\text{field}}(V_1, V_2)$  is the signed-field discrepancy defined by (C.2).*

*Proof.* By Theorem C.0.1, the Sinkhorn divergence satisfies  $S^\varepsilon(\alpha, \beta) \geq 0$  for all finite positive measures  $\alpha, \beta$ , with equality if and only if  $\alpha = \beta$ .

Since  $m_i^+$  and  $m_i^-$  are finite positive measures, it follows that

$$S^\varepsilon(m_1^+, m_2^+) \geq 0, \quad S^\varepsilon(m_1^-, m_2^-) \geq 0.$$

Therefore,

$$d_{\text{field}}(V_1, V_2) = S^\varepsilon(m_1^+, m_2^+) + S^\varepsilon(m_1^-, m_2^-) \geq 0.$$

Now suppose that  $d_{\text{field}}(V_1, V_2) = 0$ . Because both terms are non-negative, this implies

$$S^\varepsilon(m_1^+, m_2^+) = 0, \quad S^\varepsilon(m_1^-, m_2^-) = 0.$$



By definiteness of the Sinkhorn divergence,

$$m_1^+ = m_2^+, \quad m_1^- = m_2^-.$$

Hence their densities satisfy (from the uniqueness of the Radon–Nikodym derivative)

$$V_1^+ = V_2^+, \quad V_1^- = V_2^-$$

almost everywhere. Since

$$V = V^+ - V^-,$$

we obtain

$$V_1 = V_2$$

almost everywhere on  $\mathcal{X}$ .

Conversely, if  $V_1 = V_2$  almost everywhere, then their positive and negative parts coincide, implying

$$m_1^+ = m_2^+, \quad m_1^- = m_2^-,$$

and therefore

$$d_{\text{field}}(V_1, V_2) = 0.$$

□

**Proposition C.2.2** (Positivity of the multi-field discrepancy). *Let*

$$\mathbf{V}_1 = \{V_1^{(i)}\}_{i=1}^N, \quad \mathbf{V}_2 = \{V_2^{(i)}\}_{i=1}^N,$$

where  $V_j^{(i)} \in L^1(\mathcal{X})$  for  $j = 1, 2$  and  $i = 1, \dots, N$ . Suppose that  $d_{\text{field}}(V_1^{(i)}, V_2^{(i)})$  is the

signed-field discrepancy defined by (C.2). Define the multi-field discrepancy by

$$\tilde{d}_{\text{field}}(\mathbf{V}_1, \mathbf{V}_2) = \left( \sum_{i=1}^N d_{\text{field}}(V_1^{(i)}, V_2^{(i)})^p \right)^{1/p}, \quad 1 \leq p < \infty,$$

and, for  $p = \infty$ , by

$$\tilde{d}_{\text{field}}(\mathbf{V}_1, \mathbf{V}_2) = \max_{1 \leq i \leq N} d_{\text{field}}(V_1^{(i)}, V_2^{(i)}).$$

Then

$$\tilde{d}_{\text{field}}(\mathbf{V}_1, \mathbf{V}_2) \geq 0,$$

and

$$\tilde{d}_{\text{field}}(\mathbf{V}_1, \mathbf{V}_2) = 0$$

if and only if

$$V_1^{(i)} = V_2^{(i)} \quad \text{almost everywhere on } \mathcal{X}, \quad i = 1, \dots, N.$$

*Proof.* By Proposition C.2.1,  $d_{\text{field}}(V_1^{(i)}, V_2^{(i)}) \geq 0$  for every component  $i$ . Hence,

$$\tilde{d}_{\text{field}}(\mathbf{V}_1, \mathbf{V}_2) \geq 0.$$

Suppose that

$$\tilde{d}_{\text{field}}(\mathbf{V}_1, \mathbf{V}_2) = 0.$$

For  $1 \leq p < \infty$ , this implies

$$\sum_{i=1}^N d_{\text{field}}(V_1^{(i)}, V_2^{(i)})^p = 0.$$

Since every summand is non-negative, it follows that

$$d_{\text{field}}(V_1^{(i)}, V_2^{(i)}) = 0$$

for every  $i$ . For  $p = \infty$ , the same conclusion follows because the maximum of the non-

negative componentwise discrepancies is zero. Applying Proposition C.2.1 componentwise yields

$$V_1^{(i)} = V_2^{(i)}$$

almost everywhere on  $\mathcal{X}$  for all  $i = 1, \dots, N$ .

Conversely, if

$$V_1^{(i)} = V_2^{(i)}$$

almost everywhere for every  $i$ , then

$$d_{\text{field}}(V_1^{(i)}, V_2^{(i)}) = 0$$

for every component, and therefore

$$\tilde{d}_{\text{field}}(\mathbf{V}_1, \mathbf{V}_2) = 0.$$

□

Therefore, the proposed optimal transport-based flow field dissimilarity is a well-defined semimetric on the space of signed measurable flow fields in  $L^1(\mathcal{X})$ , where  $\mathcal{X}$  is a compact, measurable subset of  $\mathbb{R}^d$ . Moreover, its multi-field extension likewise defines a semimetric on finite collections of such flow fields.

## REFERENCES

- [ACB17] M. Arjovsky, S. Chintala, and L. Bottou. “Wasserstein Generative Adversarial Networks.” In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 214–223, 06–11 Aug 2017.
- [AG95] H. Alt and M. Godau. “Computing the Fréchet distance between two polygonal curves.” *International Journal of Computational Geometry & Applications*, **5**(01n02):75–91, 1995.
- [AG02] M. Amitay and A. Glezer. “Role of actuation frequency in controlled flow reattachment over a stalled airfoil.” *AIAA Journal*, **40**(2):209–216, 2002.
- [AHH18] G. Arvanitidis, L. K. Hansen, and S. Hauberg. “Latent Space Oddity: on the Curvature of Deep Generative Models.” In *Proceedings of International Conference on Learning Representations*, International Conference on Learning Representations, 2018.
- [AJS08] A. Apte, C. K. R. T. Jones, A. M. Stuart, and J. Voss. “Data assimilation: Mathematical and statistical perspectives.” *International Journal for Numerical Methods in Fluids*, **56**(8):1033–1046, 2008.
- [And82] B. D. O. Anderson. “Reverse-time diffusion equation models.” *Stochastic Processes and their Applications*, **12**(3):313–326, 1982.
- [ARF84] S. R. Ahmed, G. Ramm, and G. Faltin. “Some salient features of the time-averaged ground vehicle wake.” *Society of Automotive Engineers Transactions*, **93**:473–503, 1984.
- [ASR21] S. E. Ahmed, O. San, A. Rasheed, and T. Iliescu. “Nonlinear proper orthogonal decomposition for convection-dominated flows.” *Physics of Fluids*, **33**(12), 2021.
- [AW10] H. Abdi and L. J. Williams. “Principal component analysis.” *Wiley interdisciplinary reviews: computational statistics*, **2**(4):433–459, 2010.
- [BC94] D. J. Berndt and J. Clifford. “Using dynamic time warping to find patterns in time series.” In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, pp. 359–370, 1994.
- [BG05] I. Borg and P. J. F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York, NY, 2005.
- [BHL93] G. Berkooz, P. Holmes, and J. L. Lumley. “The proper orthogonal decomposition in the analysis of turbulent flows.” *Annual Review of Fluid Mechanics*, **25**(1):539–575, 1993.

- [BHN17] G. A. Brés, F. E. Ham, J. W. Nichols, and S. K. Lele. “Unstructured large-eddy simulations of supersonic jets.” *AIAA Journal*, **55**(4):1164–1184, 2017.
- [Bre12] P. Brémaud. *An introduction to probabilistic modeling*. Springer Science & Business Media, 2012.
- [CA21] F. Chen and K. Akasaka. “3D flow field estimation around a vehicle using convolutional neural networks.” In *Proceedings of The 32nd British Machine Vision Conference*, number 936. The British Machine Vision Association, 2021.
- [CEW10] T. C. Corke, C. L. Enloe, and S. P. Wilkinson. “Dielectric barrier discharge plasma actuators for flow control.” *Annual Review of Fluid Mechanics*, **42**(1):505–529, 2010.
- [CHS21] D. Chung, N. Hutchins, M. P. Schultz, and K. A. Flack. “Predicting the drag of rough surfaces.” *Annual Review of Fluid Mechanics*, **53**:439–471, 2021.
- [CPS18] L. Chizat, G. Peyré, B. Schmitzer, and F.-X. Vialard. “Scaling algorithms for unbalanced transport problems.” *Mathematics of Computation*, **87**(314):2563–2609, 2018.
- [CRB18] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. “Neural ordinary differential equations.” *Advances in neural information processing systems*, **31**, 2018.
- [Cut13] M. Cuturi. “Sinkhorn distances: Lightspeed computation of optimal transport.” In *Advances in Neural Information Processing Systems 26*, 2013.
- [CVT10] C. Carmeli, E. De Vito, A. Toigo, and V. Umanitá. “Vector valued reproducing kernel Hilbert spaces and universality.” *Analysis and Applications*, **8**(01):19–61, 2010.
- [DBW22] S. Druc, A. Balu, P. Wooldridge, A. Krishnamurthy, and S. Sarkar. “Concept activation vectors for generating user-defined 3D shapes.” In *Proceedings of the Institute of Electrical and Electronics Engineers Conference on Computer Vision and Pattern Recognition*, pp. 2993–3000, 2022.
- [DCF26] A. Dasgupta, A. M. da Cunha, A. Fardisi, M. Aminy, B. Binder, B. Shaddy, and A. A. Oberai. “Unifying and extending diffusion models through pdes for solving inverse problems.” *Computer Methods in Applied Mechanics and Engineering*, **448**:118431, 2026.
- [DG99] J. Dargay and D. Gately. “Income’s effect on car and vehicle ownership, worldwide: 1960–2015.” *Transportation Research Part A: Policy and Practice*, **33**(2):101–138, 1999.

- [DKM94] A. V. Dovgal, V. V. Kozlov, and A. Michalke. “Laminar boundary layer separation: Instability and associated phenomena.” *Progress in Aerospace Sciences*, **30**:61–94, 1994.
- [Don00] D. L. Donoho. “High-dimensional data analysis: The curses and blessings of dimensionality.” *AMS math challenges lecture*, **1**(2000):32, 2000.
- [DPF24] P. Du, M. H. Parikh, X. Fan, X.-Y. Liu, and J.-X. Wang. “Conditional neural field latent diffusion model for generating spatiotemporal turbulence.” *Nature Communications*, **15**(1):10416, 2024.
- [DS00] M. L. Davison and S. G. Sireci. “Multidimensional scaling.” In *Handbook of Applied Multivariate Statistics and Mathematical Modeling*, pp. 323–352. Elsevier, 2000.
- [DT91] A. Debussche and R. Temam. “Inertial manifolds and slow manifolds.” *Applied Mathematics Letters*, **4**(4):73–76, 1991.
- [EDA24] M. Elrefaie, A. Dai, and F. Ahmed. “DrivAerNet: A Parametric Car Dataset for Data-Driven Aerodynamic Design and Graph-Based Drag Prediction.” In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 88360, p. V03AT03A019. American Society of Mechanical Engineers, 2024.
- [eia23] “International energy outlook 2023.” *US Energy Information Administration*, 2023.
- [EPS14] O. Edenhofer, R. Pichs-Madruga, Y. Sokona, E. Farahani, S. Kadner, K. Seyboth, A. Adler, I. Baum, S. Brunner, P. Eickemeier, et al. *Climate Change 2014: Mitigation of Climate Change*. Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2014.
- [ES95] R. Everson and L. Sirovich. “Karhunen–Loeve procedure for gappy data.” *Journal of the Optical Society of America A*, **12**(8):1657–1664, 1995.
- [Eva12] L. C. Evans. *An introduction to stochastic differential equations*, volume 82. American Mathematical Soc., 2012.
- [Eva22] L. C. Evans. *Partial differential equations*, volume 19. American mathematical society, 2022.
- [FCG21] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, et al. “POT: Python Optimal Transport.” *Journal of Machine Learning Research*, **22**(78):1–8, 2021.

- [Fel91] W. Feller. *An introduction to probability theory and its applications, Volume 2*, volume 2. John Wiley & Sons, 1991.
- [FMN16] C. Fefferman, S. Mitter, and H. Narayanan. “Testing the manifold hypothesis.” *Journal of the American Mathematical Society*, **29**(4):983–1049, 2016.
- [FMR01] C. Foias, O. Manley, R. Rosa, and R. Temam. *Navier-Stokes equations and turbulence*, volume 83. Cambridge University Press, 2001.
- [FMT88] C. Foias, O. Manley, and R. Temam. “Modelling of the interaction of small and large eddies in two dimensional turbulent flows.” *European Series in Applied and Industrial Mathematics: Mathematical Modelling and Numerical Analysis*, **22**(1):93–118, 1988.
- [Fre97] J. B. Freund. “Proposed inflow/outflow boundary condition for direct computation of aerodynamic sound.” *AIAA Journal*, **35**(4):740–742, 1997.
- [FST88] C. Foias, G. R. Sell, and R. Temam. “Inertial manifolds for nonlinear evolutionary equations.” *Journal of Differential Equations*, **73**(2):309–353, 1988.
- [FT23] K. Fukami and K. Taira. “Grasping extreme aerodynamics on a low-dimensional manifold.” *Nature Communications*, **14**(1):6480, 2023.
- [FT24] K. Fukami and K. Taira. “Single-snapshot machine learning for super-resolution of turbulence.” *Journal of Fluid Mechanics*, **1001**:A32, 2024.
- [GAS09] E. Garnier, N. Adams, and P. Sagaut. *Large eddy simulation for compressible flows*. Springer Science & Business Media, 2009.
- [GCB19] A. Genevay, L. Chizat, F. F. Bach, M. Cuturi, and G. Peyré. “Sample Complexity of Sinkhorn divergences.” In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1574–1583, 2019.
- [Ger81] J. J. Gerbrands. “On the relationships between SVD, KLT and PCA.” *Pattern recognition*, **14**(1-6):375–381, 1981.
- [GPI23] R. Garcia-Fernandez, K. Portal-Porras, O. Irigaray, Z. Ansa, and U. Fernandez-Gamiz. “CNN-based flow field prediction for bus aerodynamics analysis.” *Sci. Rep.*, **13**(1):21213, 2023.
- [GR14] I. S. Gradshteyn and I. M. Ryzhik. *Table of integrals, series, and products*. Academic press, 2014.
- [GSE95] N. Gordon, D. Salmond, and C. Ewing. “Bayesian state estimation for tracking and guidance using the bootstrap filter.” *Journal of Guidance, Control, and Dynamics*, **18**(6):1434–1443, 1995.

- [GT18] A. N. Gorban and I. Y. Tyukin. “Blessing of dimensionality: mathematical foundations of the statistical physics of data.” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, **376**(2118):20170237, 2018.
- [GV25] L. Guastoni and R. Vinuesa. “A new perspective on the simulation of stochastic problems in fluid mechanics with diffusion models: Fluid dynamics.” *Nature Machine Intelligence*, **7**(6):816–817, 2025.
- [GW00] D. Greenblatt and I. J. Wygnanski. “The control of flow separation by periodic excitation.” *Progress in Aerospace Sciences*, **36**(7):487–545, 2000.
- [HBA00] C. P. Häggmark, A. A. Bakchinov, and P. H. Alfredsson. “Experiments on a two-dimensional laminar separation bubble.” *Philosophical Transactions of the Royal Society of London A.*, **358**(1777):3193–3205, 2000.
- [HGR04] C. Hinterberger, M. Garcia-Villalba, and W. Rodi. “Large eddy simulation of flow around the Ahmed body.” In *The Aerodynamics of Heavy Vehicles: Trucks, Buses, and Trains*, pp. 77–87. Springer, 2004.
- [HI04] F. Ham and G. Iaccarino. “Energy conservation in collocated discretization schemes on unstructured meshes.” *Annual Research Briefs*, **2004**(3-14):118, 2004.
- [HMI06] F. Ham, K. Mattsson, and G. Iaccarino. “Accurate and stable finite volume operators for unstructured flow solvers.” *Annual Research Briefs*, **243**, 2006.
- [HO93] P. C. Hansen and D. P. O’Leary. “The use of the L-curve in the regularization of discrete ill-posed problems.” *SIAM Journal on Scientific Computing*, **14**(6):1487–1503, 1993.
- [Hot33] H. Hotelling. “Analysis of a complex of statistical variables into principal components.” *Journal of educational psychology*, **24**(6):417, 1933.
- [HP71] J. K. Harvey and F. J. Perry. “Flowfield produced by trailing vortices in the vicinity of the ground.” *AIAA Journal*, **9**(8):1659–1660, 1971.
- [HS06] G. E. Hinton and R. R. Salakhutdinov. “Reducing the dimensionality of data with neural networks.” *Science*, **313**(5786):504–507, 2006.
- [HZR16] K. He, X. Zhang, S. Ren, and J. Sun. “Deep residual learning for image recognition.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [iea22] “World energy outlook 2022.” International Energy Agency, 2022.



- [iea23] “World energy outlook 2023.” International Energy Agency, 2023.
- [IKM19] R. Ishar, E. Kaiser, M. Morzyński, D. Fernex, R. Semaan, M. Albers, P. S. Meysonnat, W. Schröder, and B. R. Noack. “Metric for attractor overlap.” *Journal of Fluid Mechanics*, **874**:720–755, 2019.
- [IS15] S. Ioffe and C. Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift.” In *International Conference on Machine Learning*, pp. 448–456. pmlr, 2015.
- [JFV23] T. Jaroslowski, M. Forte, O. Vermeersch, J.-M. Moschetta, and E. R. Gowree. “Disturbance growth in a laminar separation bubble subjected to free-stream turbulence.” *Journal of Fluid Mechanics*, **956**:A33, 2023.
- [JG23] C. E. P. De Jesús and M. D. Graham. “Data-driven low-dimensional dynamic model of Kolmogorov flow.” *Physical Review Fluids*, **8**(4):044402, 2023.
- [JM09] R. D. Joslin and D. N. Miller. *Fundamentals and applications of modern flow control*. American Institute of Aeronautics and Astronautics, 2009.
- [JMO21] S. J. Jacob, M. Mrosek, C. Othmer, and H. Köstler. “Deep learning for real-time aerodynamic evaluations of arbitrary vehicle shapes.” *arXiv:2108.05798*, 2021.
- [Jol02] I. T. Jolliffe. *Principal component analysis*. Springer, 2002.
- [Kal18] G. Kalghatgi. “Is it really the end of internal combustion engines and petroleum in transport?” *Applied energy*, **225**:965–974, 2018.
- [Kan60] L. V. Kantorovich. “Mathematical methods of organizing and planning production.” *Management Science*, **6**(4):366–422, 1960.
- [Kar46] K. Karhunen. “Zur spektraltheorie stochastischer prozesse.” *Annales Academiae Scientiarum Fennicae Series A. I. Mathematica*, **34**, 1946.
- [Kar47] K. Karhunen. “Über lineare Methoden in der Wahrscheinlichkeitsrechnung.” *Annales Academiæ Scientiarum Fennicae Mathematica*, **37**:1, 1947.
- [KB15] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization.” In *International Conference on Learning Representations (ICLR)*, 2015.
- [KHN11] Y. Khalighi, F. Ham, J. Nichols, S. Lele, and P. Moin. “Unstructured Large Eddy Simulation for Prediction of Noise Issued from Turbulent Jets in Various Configurations.” *17th AIAA/CEAS Aeroacoustics Conference (32nd AIAA aeroacoustics conference)*. *AIAA Paper* 2011-2886, 2011.
- [KM24] D. Kelshaw and L. Magri. “Proper Latent Decomposition.” *arXiv:2412.00785*, 2024.

- [KMM10] R. B. Kotapati, R. Mittal, O. Marxen, F. Ham, D. You, and L. N. Cattafesta III. “Nonlinear dynamics and synthetic-jet-based control of a canonical separated flow.” *Journal of Fluid Mechanics*, **654**:65–97, 2010.
- [KP77] P. E. Kloeden and R. A. Pearson. “The numerical solution of stochastic differential equations.” *The ANZIAM Journal*, **20**(1):8–12, 1977.
- [Kru64] J. B. Kruskal. “Nonmetric multidimensional scaling: a numerical method.” *Psychometrika*, **29**(2):115–129, 1964.
- [KS23] M. D. Kvalheim and E. D. Sontag. “Why should autoencoders work?” *arXiv:2310.02250*, 2023.
- [KSM97] M. Kiya, M. Shimizu, and O. Mochizuki. “Sinusoidal forcing of a turbulent separation bubble.” *Journal of Fluid Mechanics*, **342**:119–139, 1997.
- [KSS25] C. Klewicki, A. Schenkman, and G. Spedding. “Aerodynamic Control of Laminar Separation on a Wall-Bounded Airfoil at Transitional Reynolds Numbers.” *arXiv:2504.15191*, 2025.
- [KT16] T. Kajishima and K. Taira. *Computational fluid dynamics: incompressible turbulent flows*. Springer, 2016.
- [Kur78] Y. Kuramoto. “Diffusion-induced chaos in reaction systems.” *Progress of Theoretical Physics Supplement*, **64**:346–367, 1978.
- [KV02] K. Kunisch and S. Volkwein. “Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics.” *SIAM Journal on Numerical Analysis*, **40**(2):492–515, 2002.
- [KW78] J. B. Kruskal and M. Wish. *Multidimensional scaling*. Number 11. Sage, Newbury Park, CA, 1978.
- [KW14] D. P. Kingma and M. Welling. “Auto-Encoding Variational Bayes.” In *International Conference on Learning Representations (ICLR)*, 2014.
- [Lac61] G. V. Lachmann. *Boundary layer and flow control: its principles and application*. Pergamon Press, 1961.
- [Lax14] P. D. Lax. *Functional analysis*. John Wiley & Sons, 2014.
- [LB03] H. Lienhart and S. Becker. “Flow and turbulence structure in the wake of a simplified car model.” *Society of Automotive Engineers transactions*, **112**:785–796, 2003.

- [LBB02] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition.” *Proceedings of the IEEE*, **86**(11):2278–2324, 2002.
- [LBT23] A. J. Linot, J. W. Burby, Q. Tang, P. Balaprakash, M. D. Graham, and R. Maulik. “Stabilized neural ordinary differential equations for long-time forecasting of dynamical systems.” *Journal of Computational Physics*, **474**:111838, 2023.
- [Led01] M. Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2001.
- [Lee03] J. M. Lee. *Introduction to smooth manifolds*. Springer, 2003.
- [LG22] A. J. Linot and M. D. Graham. “Data-driven reduced-order modeling of spatiotemporal chaos with neural ordinary differential equations.” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **32**(7), 2022.
- [Lis83] P. B. S. Lissaman. “Low-Reynolds-number airfoils.” *Annual Review of Fluid Mechanics*, **15**(1):223–239, 1983.
- [LMS18] M. Liero, A. Mielke, and G. Savaré. “Optimal entropy-transport problems and a new Hellinger–Kantorovich distance between positive measures.” *Inventiones Mathematicae*, **211**(3):969–1117, 2018.
- [Loe78] M. Loève. “Probability theory. II, volume 46 of.” *Graduate Texts in Mathematics*, 1978.
- [Lor63] E. N. Lorenz. “Deterministic nonperiodic flow.” *Journal of the Atmospheric Sciences*, **20**:130–141, 1963.
- [LYS22] Y. Lee, S. Yoon, M. Son, and F. C. Park. “Regularized autoencoders for isometric representation learning.” In *International Conference on Learning Representations*, 2022.
- [LYT67] J. L. Lumley, A. M. Yaglom, and V. I. Tatarski. “Atmospheric turbulence and radio wave propagation.” *Journal of Computational Chemistry*, **23**(13):1236–1243, 1967.
- [LZG23] A. J. Linot, K. Zeng, and M. D. Graham. “Turbulence control in plane Couette flow using low-dimensional neural ODE-based models and deep reinforcement learning.” *International Journal of Heat and Fluid Flow*, **101**:109139, 2023.
- [Mai12a] E. Mainini. “A description of transport cost for signed measures.” *Journal of Mathematical Sciences*, **181**:837–855, 2012.

- [Mai12b] E. Mainini. “Well-posedness for a mean field model of Ginzburg–Landau vortices with opposite degrees.” *Nonlinear Differential Equations and Applications*, **19**:133–158, 2012.
- [MD03] T. J. Mueller and J. D. DeLaurier. “Aerodynamics of small vehicles.” *Annual Review of Fluid Mechanics*, **35**(1):89–111, 2003.
- [MD22] L. Magri and N. A. K. Doan. “On interpretability and proper latent decomposition of autoencoders.” In *Center for Turbulence Research Proceedings of the Summer Program*, pp. 107–115, 2022.
- [ME25] H. Mousavi and J. D. Eldredge. “Low-order flow reconstruction and uncertainty quantification in disturbed aerodynamics using sparse pressure measurements.” *Journal of Fluid Mechanics*, **1013**:A41, 2025.
- [MFF20] T. Murata, K. Fukami, and K. Fukagata. “Nonlinear mode decomposition with convolutional neural networks for fluid dynamics.” *Journal of Fluid Mechanics*, **882**:A13, 2020.
- [MFZ22] M. Morimoto, K. Fukami, K. Zhang, and K. Fukagata. “Generalization techniques of neural networks for fluid flow estimation.” *Neural Computing and Applications*, **34**(5):3647–3669, 2022.
- [MJE25] H. Mousavi, A. Jones, and J. D. Eldredge. “Sequential estimation of disturbed aerodynamic flows from sparse measurements via a reduced latent space.” *arXiv preprint arXiv:2509.03795*, 2025.
- [MKK18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. “Spectral Normalization for Generative Adversarial Networks.” In *International Conference on Learning Representations*, 2018.
- [MKO15] K. Machida, M. Kaneko, and A. Ogawa. “Aerodynamic development of the new Honda Fit/Jazz.” Technical report, Society of Automotive Engineers Technical Paper, 2015.
- [MLR13] O. Marxen, M. Lang, and U. Rist. “Vortex formation and vortex breakup in a laminar separation bubble.” *Journal of Fluid Mechanics*, **728**:58–90, 2013.
- [MNA18] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, et al. “Mixed Precision Training.” In *International Conference on Learning Representations*, 2018.
- [Ngu25] K. Nguyen. “An Introduction to Sliced Optimal Transport: Foundations, Advances, Extensions, and Applications.” *Foundations and Trends® in Computer Graphics and Vision*, **17**(3-4):171–406, 2025.

- [NOT15] D. Nakamura, Y. Onishi, and Y. Takehara. “Flow field analysis in the development of the 2013 model year accord hybrid.” Technical report, Society of Automotive Engineers Technical Paper, 2015.
- [NYA24] H. Nagaoka, B. Yenerdag, K. Ambo, D. Philips, C. Ivey, G. Brès, and S. Bose. “Prediction of aerodynamic drag in SUVs with different specifications by using Large-Eddy simulations.” Technical report, Society of Automotive Engineers Technical Paper, 2024.
- [PC19] G. Peyré and M. Cuturi. “Computational optimal transport.” *Foundations and Trends in Machine Learning*, **11**(5–6):55–607, 2019.
- [Peh22] B. Peherstorfer. “Breaking the Kolmogorov barrier with nonlinear model reduction.” *Notices of the American Mathematical Society*, **69**(5):725–733, 2022.
- [Pra25] L. Prandtl. “Magnuseffekt und windkraftschiff.” *The Science of Nature*, **13**(6):93–108, 1925.
- [Pre98] L. Prechelt. “Automatic early stopping using cross validation: quantifying the criteria.” *Neural Networks*, **11**(4):761–767, 1998.
- [PW09] O. Pele and M. Werman. “Fast and robust earth mover’s distances.” In *2009 IEEE 12th International Conference on Computer Vision*, pp. 460–467. IEEE, 2009.
- [RDM23] A. Racca, N. A. K. Doan, and L. Magri. “Predicting turbulent dynamics with the convolutional autoencoder echo state network.” *Journal of Fluid Mechanics*, **975**:A2, 2023.
- [RHW86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. “Learning representations by back-propagation errors.” *Nature*, **322**:533–536, 1986.
- [Row02] C. W. Rowley. *Modeling, simulation, and control of cavity flow oscillations*. California Institute of Technology, 2002.
- [RSM19] T. Rios, B. Sendhoff, S. Menzel, T. Bäck, and B. van Stein. “On the efficiency of a point cloud autoencoder as a geometric representation for shape optimization.” In *2019 Institute of Electrical and Electronics Engineers Symposium Series on Computational Intelligence*, pp. 791–798. IEEE, 2019.
- [RSW21] T. Rios, B. van Stein, P. Wollstadt, T. Bäck, B. Sendhoff, and S. Menzel. “Exploiting local geometric features in vehicle design optimization with 3D point cloud autoencoders.” In *2021 Institute of Electrical and Electronics Engineers Congress on Evolutionary Computation*, pp. 514–521. IEEE, 2021.

- [RTG98] Y. Rubner, C. Tomasi, and L. J. Guibas. “A metric for distributions with applications to image databases.” In *Sixth International Conference on Computer Vision*, 1998.
- [SAK12] A. Sinha, H. Alkandry, M. Kearney-Fischer, M. Samimy, and T. Colonius. “The impulse response of a high-speed jet forced with localized arc filament plasma actuators.” *Physics of Fluids*, **24**(12), 2012.
- [SE20] Y. Song and S. Ermon. “Improved techniques for training score-based generative models.” *Advances in neural information processing systems*, **33**:12438–12448, 2020.
- [SFL11] B. K. Sriperumbudur, K. Fukumizu, and G. R. G. Lanckriet. “Universality, Characteristic Kernels and RKHS Embedding of Measures.” *Journal of Machine Learning Research*, **12**(7), 2011.
- [SFS24] L. Smith, K. Fukami, G. Sedky, A. Jones, and K. Taira. “A cyclic perspective on transient gust encounters through the lens of persistent homology.” *Journal of Fluid Mechanics*, **980**:A18, 2024.
- [SFV19] T. Séjourné, J. Feydy, F.-X. Vialard, A. Trounev, and G. Peyré. “Sinkhorn divergences for unbalanced optimal transport.” *arXiv:1910.12958*, 2019.
- [SHS11] W. J. Sweeting, A. R. Hutchinson, and S. D. Savage. “Factors affecting electric vehicle energy consumption.” *International Journal of Sustainable Engineering*, **4**(3):192–201, 2011.
- [Sil18] B. W. Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.
- [SMH24] S. Syrota, P. Moreno-Muñoz, and S. Hauberg. “Decoder ensembling for learned latent geometries.” In *Proceedings of the Geometry-grounded Representation Learning and Generative Modeling Workshop (GRaM)*, volume 251 of *Proceedings of Machine Learning Research*, pp. 277–285. PMLR, 29 Jul 2024.
- [SP99] A. Seifert and L. G. Pack. “Oscillatory control of separation at high Reynolds numbers.” *AIAA Journal*, **37**(9):1062–1071, 1999.
- [SPO16] M. Sieber, C. O. Paschereit, and K. Oberleithner. “Spectral proper orthogonal decomposition.” *Journal of Fluid Mechanics*, **792**:798–828, 2016.
- [SSK20] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. “Score-based generative modeling through stochastic differential equations.” *arXiv preprint arXiv:2011.13456*, 2020.

- [STG21] J. A. Sanguesa, V. Torres-Sanz, P. Garrido, F. J. Martinez, and J. M. Marquez-Barja. “A review on electric vehicles: Technologies and challenges.” *Smart Cities*, **4**(1):372–404, 2021.
- [SYP23] B. Song, C. Yuan, F. Permenter, N. Arechiga, and F. Ahmed. “Surrogate modeling of car drag coefficient with depth and normal renderings.” In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 87301, p. V03AT03A029. American Society of Mechanical Engineers, 2023.
- [Tan64] I. Tani. “Low-speed flows involving bubble separations.” *Progress in Aerospace Sciences*, **5**:70–103, 1964.
- [Tao11] T. Tao. *An introduction to measure theory*, volume 126. American Mathematical Society, 2011.
- [TBD17] K. Taira, S. L. Brunton, S. T. M. Dawson, C. W. Rowley, T. Colonius, B. J. McKeon, O. T. Schmidt, S. Gordeyev, V. Theofilis, and L. S. Ukeiley. “Modal analysis of fluid flows: An overview.” *AIAA Journal*, **55**(12):4013–4041, 2017.
- [Tem89] R. Temam. “Do inertial manifolds apply to turbulence?” *Physica D: Nonlinear Phenomena*, **37**(1-3):146–152, 1989.
- [Tem12] R. Temam. *Infinite-dimensional dynamical systems in mechanics and physics*. Springer Science & Business Media, 2012.
- [TFI24] J. Tran, K. Fukami, K. Inada, D. Umehara, Y. Ono, K. Ogawa, and K. Taira. “Aerodynamics-guided machine learning for design optimization of electric vehicles.” *Communications Engineering*, **3**(1):174, 2024.
- [TYT26] J. Tran, C.-A. Yeh, and K. Taira. “Using optimal transport aligned latent embeddings for separated flow analysis.” *Journal of Fluid Mechanics*, **1027**:A24, 2026.
- [UB18] N. Umetani and B. Bickel. “Learning three-dimensional flow for interactive aerodynamic design.” *Association for Computing Machinery Transactions on Graphics*, **37**(4):1–10, 2018.
- [Vas69] L. N. Vaserstein. “Markov processes over denumerable products of spaces, describing large systems of automata.” *Problemy Peredachi Informatsii*, **5**(3):64–72, 1969.
- [VB22] R. Vinuesa and S. L. Brunton. “Enhancing computational fluid dynamics with machine learning.” *Nature Computational Science*, **2**(6):358–366, 2022.

- [VEN25] P. Vatani, M. Elrefaie, F. Nazarpour, and F. Ahmed. “TripOptimizer: Generative three-dimensional shape optimization and drag prediction using triplane variational autoencoder networks.” *Physics of Fluids*, **37**(12), 2025.
- [Vil09] C. Villani. *Optimal transport: old and new*. Springer, Berlin, 2009.
- [Vre04] A. W. Vreman. “An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications.” *Physics of fluids*, **16**(10):3670–3681, 2004.
- [WH18] Y. Wu and K. He. “Group normalization.” In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.
- [Wil91] D. Williams. *Probability with martingales*. Cambridge university press, Cambridge, 1991.
- [Wil06] K. Willcox. “Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition.” *Computers & fluids*, **35**(2):208–226, 2006.
- [WLD98] J.-Z. Wu, X.-Y. Lu, A. G. Denny, M. Fan, and J.-M. Wu. “Post-stall flow control on an airfoil by local unsteady forcing.” *Journal of Fluid Mechanics*, **371**:21–58, 1998.
- [WRK23] A. Warey, V. Raul, S. Kaushik, T. Han, and R. Chakravarty. “Generative Inverse Design of Aerodynamic Shapes Using Conditional Invertible Neural Networks.” *Journal of Computing and Information Science in Engineering*, **23**(3):031006, 2023.
- [Wyg97] I. Wygnanski. “Boundary layer and flow control by periodic addition of momentum.” In *4th Shear Flow Control Conference*, p. 2117, 1997.
- [YB21] Z. Yu and Z. Bingfu. “Recent advances in wake dynamics and active drag reduction of simple automotive bodies.” *Applied Mechanics Reviews*, **73**(6):060801, 2021.
- [YMT17a] C.-A. Yeh, P. Munday, and K. Taira. “Use of local periodic heating for separation control on a NACA 0012 airfoil.” *55th AIAA Aerospace Sciences Meeting, AIAA Paper* 2017-1451, 2017.
- [YMT17b] C.-A. Yeh, P. M. Munday, and K. Taira. “Laminar free shear layer modification using localized periodic heating.” *Journal of Fluid Mechanics*, **822**:561–589, 2017.
- [YSK09] S. Yarusevych, P. E. Sullivan, and J. G. Kawall. “On vortex shedding from an airfoil in low-Reynolds-number flows.” *Journal of Fluid Mechanics*, **632**:245–271, 2009.



- [YT19] C.-A. Yeh and K. Taira. “Resolvent-analysis-based design of airfoil separation control.” *Journal of Fluid Mechanics*, **867**:572–610, 2019.
- [ZFC16] N.-G. Zacharof, G. Fontaras, B. Ciuffo, S. Tsiakmakis, K. Anagnostopoulos, A. Marotta, and J. Pavlovic. *Review of in use factors affecting the fuel consumption and CO2 emissions of passenger cars*. Publications Office of the European Union Luxembourg, 2016.
- [ZLG22] K. Zeng, A. J. Linot, and M. D. Graham. “Data-driven control of spatiotemporal chaos with reduced-order neural ODE-based models and reinforcement learning.” *Proceedings of the Royal Society A*, **478**(2267):20220297, 2022.
- [ZMM23] A. Zahoor, F. Mehr, G. Mao, Y. Yu, and A. Sápi. “The carbon neutrality feasibility of worldwide and in China’s transportation sector by E-car and renewable energy sources before 2060.” *Journal of Energy Storage*, **61**:106696, 2023.
- [ZMR89] K. B. M. Q. Zaman, D. J. McKinzie, and C. L. Rumsey. “A natural low-frequency oscillation of the flow over an airfoil near stalling conditions.” *Journal of Fluid Mechanics*, **202**:403–442, 1989.