# Autoencoders for Lagrangian Computational Fluid Dynamics

Joseph O'Connor
EPCC, University of Edinburgh
Edinburgh, UK
j.oconnor@epcc.ed.ac.uk

## I. Introduction

Smoothed particle hydrodynamics (SPH) is the current state of the art for high-fidelity computational fluid dynamics (CFD) simulations of violent free-surface flows. However, SPH is computationally expensive, requiring a larger support stencil than traditional mesh-based CFD methods (e.g. finite difference/volume). This makes large-scale SPH simulations of real-world problems extremely challenging and has also limited its use in common engineering workflows (e.g. optimisation, uncertainty quantification). Recent advances in data-driven methods are having a transformative effect on accelerating mesh-based CFD, with speedups in the range of two to three orders of magnitude reported [1], [2]. However, the application of data-driven methods to Lagrangian CFD is still extremely limited. This is mainly due to the unique characteristics of its formulation (e.g. unstructured and dynamic particle distributions), which are not compatible with many popular data-driven methods.

One area where data-driven methods have had a significant impact on mesh-based CFD is the development of reduced-order models (ROMs). The main idea behind ROMs is that many natural dynamical systems can be described by a small number of dominant features whose dynamics evolve on a low-dimensional manifold [3]. By neglecting small-scale details and only modelling the dominant dynamics, these models are much more efficient than their high-fidelity counterparts. The first step in constructing a ROM is to identify a reduced set of coordinates along which the high-dimensional dynamics can be approximated. Traditional approaches to dimensionality reduction in mesh-based CFD have typically relied on proper orthogonal decomposition (POD) – otherwise known as principal component analysis or singular value decomposition – to find a reduced set of orthogonal coordinates (modes) that form an optimal basis on which to represent the flow [4]. Once a reduced set of coordinates have been identified, various approaches exist for constructing a dynamical model that approximates the high-dimensional dynamics in the reduced coordinates (e.g. Galerkin projection, symbolic regression, recurrent neural networks).

One of the main drawbacks with POD is, as a linear method, it is restricted to learning a linear subspace on which to project the high-dimensional dynamics. For highly nonlinear problems, such as the Navier-Stokes equations, this can result in deteriorated performance, requiring a large number of modes to capture the essential dynamics. However, modern data-driven techniques based on the autoencoder architecture can learn a reduced basis on a nonlinear manifold, offering greater compression in the reduced (latent) space [5]. The key idea behind an autoencoder is to compress the input data into a lower-dimensional latent representation (encoding) and then reconstruct the data back to its original form (decoding). This is extremely useful for tasks such as image classification or, in the case of CFD, constructing ROMs, and several recent works have demonstrated improved performance over POD [5], [6].

The integration of data-driven techniques with Lagrangian CFD methods, such as SPH, is almost non-existent. None of the methods described above – even POD, which has been applied to mesh-based CFD methods since the 1960s [7] – have ever been applied to SPH. This is mainly due to the unique characteristics of the Lagrangian SPH formulation, which is unstructured with dynamic particle positions and edge connectivity. This makes it impossible to apply popular data-driven techniques (e.g. convolutional neural networks) that have been widely applied to mesh-based CFD methods. As a result, there is currently no method for dimensionality reduction with SPH. Considering this, the purpose of this work is to develop a novel autoencoder architecture that is suitable for Lagrangian CFD methods. By enabling dimensionality reduction for SPH, this will unlock a number of novel downstream tasks, such as the construction of ROMs and efficient analysis of high-dimensional data.

## II. Materials & Methods

An autoencoder (illustrated in Figure 1) consists of two main components: an encoder, which compresses the input data into a compact latent space (bottleneck); and a decoder, which reconstructs the original data from this compressed representation. By minimising the reconstruction error, the autoencoder learns to extract the dominant features of the input data, making it ideal for dimensionality reduction. Autoencoders have found widespread success in image processing, as well as mesh-based CFD [5], [6], where they are typically composed of convolutional layers (designed to handle structured pixel/node data) combined with pooling operations to extract hierarchical features [8]. However, such techniques cannot be applied to Lagrangian CFD, owing to the unstructured data layout. Drawing inspiration from computer vision, point clouds (usually

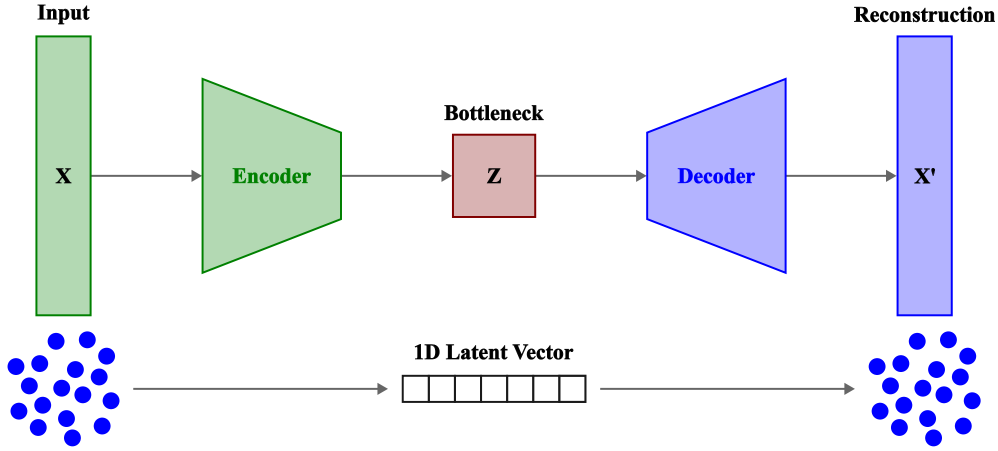


Fig. 1. Schematic of an autoencoder. The input (particle distribution) is passed through the encoder, which compresses the data down to a 1D latent vector of reduced dimension (bottleneck). Following this, the decoder reconstructs the original particle distribution from the latent vector.

obtained via laser scanning) are a collection of data points used to represent the surface of an object in 3D space. Moreover, point cloud classification is a common procedure for extracting a reduced set of features from point cloud data (e.g. for object identification). This is essentially equivalent to the goal of this work – identifying low-dimensional representations of SPH data – motivating the idea that techniques from computer vision can be adapted for dimensionality reduction with SPH.

The input data $\mathbf{X}$ is an array of size $N \times d$, where $N$ is the number of particles and $d$ is the number of particle features. For SPH, these features may include the particle's position, velocity, density, and any other descriptors (e.g. type). However, for simplicity, the present work only considers the positions. For the encoder, this work employs the popular PointNet architecture [9], originally designed for point cloud classification/segmentation. This consists of a shared multi-layer perceptron (MLP) applied to each individual particle to project its feature vector to a higher dimension. This results in a new array of size $N \times l$, where $l$ is sized according to the intended size of the latent space. Following this, a global pooling operation (e.g. max pooling) is performed to aggregate the features into a 1D latent space vector $\mathbf{Z}$ of size $l$. This latent vector provides a highly compressed global representation of the particle distribution which can be used for a variety of downstream tasks (e.g. constructing ROMs).

While the encoder provides the ability to map the high-dimensional state to the low-dimensional latent space, the decoder is required to map the latent vector back to the original high-dimensional space. In this work, a standard MLP is adopted to project the latent space vector to a new 1D vector of size $Md$, where $M$ is the number of particles that are to be generated. Following this, the 1D vector is reshaped into an array $\mathbf{X}'$ of size $M \times d$, which is the final output of the autoencoder and represents the reconstructed particle distribution, such that $\mathbf{X}' \approx \mathbf{X}$. Note that one of the limitations of using an MLP for the decoder is that the size of the output array is built into the model and therefore must be specified *apriori*. Future work will investigate more advanced generative methods that offer the flexibility of generating an arbitrary number of particles.

After reconstructing the particle distribution, the loss function must be defined for training the model. Traditional loss functions, such as mean squared error (MSE), are commonly used in image processing and mesh-based CFD. However, MSE is unsuitable for this work for three main reasons. Firstly, it requires the input and output arrays to be of equal size, which is not generally true for the model described above (i.e. $N \neq M$). Secondly, it is not permutation invariant, meaning that the same two distributions could give different loss values depending on how the particles are ordered within the array. Finally, it is sensitive to local differences in the distribution, meaning that it will adversely penalise two distributions of particles that are globally very similar but have local differences. Considering this, the present work adopts the earth mover's distance (EMD), which is based on optimal transport theory, and describes the cost of moving one distribution onto the other.

## III. Results & Discussion

To evaluate the performance of the proposed autoencoder, the model is trained on the *WaterDrop* dataset from [10]. This dataset consists of several 2D simulation trajectories generated using the material point method. Each trajectory begins with a fluid volume initially suspended inside an enclosed box, which is then released and allowed to evolve in time. The training set contains 1000 individual trajectories, while the validation and test sets each consist of 30 trajectories. Figure 2 provides a snapshot from one of these trajectories. For each trajectory, the size of the fluid volume (i.e. the number of particles) and its initial location are randomly sampled, ensuring a diverse range of scenarios. Across all trajectories, the maximum number of particles is 1000, and simulations span 1000 time steps.

Figure 2 shows preliminary results demonstrating the first autoencoder developed for Lagrangian CFD. The original particle distribution consists of 532 particles in 2D, corresponding to

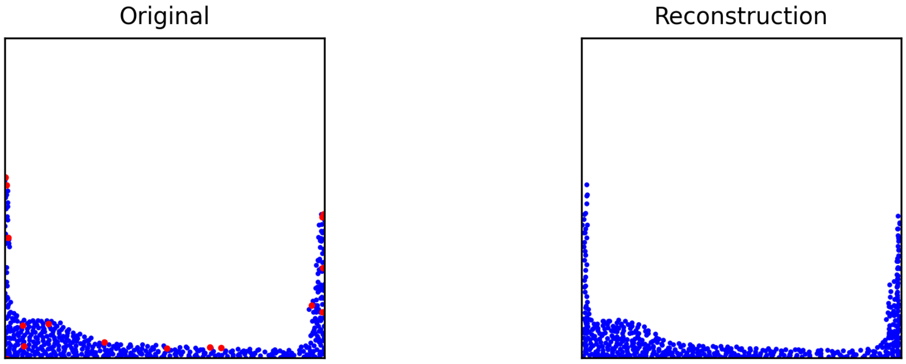


Fig. 2. Preliminary results demonstrating an autoencoder for Lagrangian CFD. The data consists of 532 particles in 2D, which has a dimension size of 532 x 2 = 1064. This high-dimensional representation is compressed to a latent space size of 16 before reconstructing the original distribution. Red particles indicate the particles which are retained in the latent space during the max pooling operation.

an input array $\mathbf{X}$ of size $532 \times 2 = 1064$. This is compressed to a latent space vector $\mathbf{Z}$ of size 16, achieving a compression ratio of 66.5. While local differences are evident at the scale of individual particles, the reconstructed distribution successfully captures the main features of the original, including the run-up along both side walls and the initial wave formation in the bottom left corner. Remarkably, the reconstructed distribution is generated from only 16 floating-point numbers, highlighting the effectiveness of the latent space in capturing the global properties of the flow. These results demonstrate significant potential for dimensionality reduction with SPH, which could be used to enable ROMs for SPH, as well as a variety of other tasks, such as post-processing complex datasets, characterising dynamics, and designing control strategies.

While the initial results are promising, there are certain limitations with the present approach. Firstly, the PointNet encoder operates on each particle separately without considering its local neighbourhood, neglecting local structure. To enrich the latent space, more advanced hierarchical encoders should be investigated (e.g. PointNet++ [11], graph neural networks). Secondly, the use of an MLP for the decoder fixes the number of particles that can be generated. For practical applications, a more flexible approach will be necessary. Therefore, more advanced generative methods should be investigated (e.g. denoising diffusion probabilistic models [12]). Finally, for simplicity, the present work only considers the particle positions in the feature vector. For real applications, it will be necessary to consider the hydrodynamic properties of each particle (e.g. velocity, density). Nevertheless, the present work is a significant first step towards dimensionality reduction for SPH.

## IV. Conclusions

This work presents the first application of autoencoders to Lagrangian CFD, introducing a novel dimensionality reduction technique for SPH. The proposed model demonstrates the ability to compress high-dimensional particle distributions into a compact latent space, while preserving the essential features of the flow. By leveraging the PointNet architecture, the autoencoder effectively captures global flow properties despite the unstructured nature of SPH data. Preliminary results highlight significant potential for future advancements, including the development of ROMs for SPH. However, several limitations remain, such as the need to capture local particle structure and flexible particle generation. Future efforts will focus on exploring hierarchical encoders and integrating advanced generative models to overcome these challenges. Overall, this work establishes a foundational approach to dimensionality reduction for SPH, paving the way for broader applications.

## References

[1] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer, "Machine learning–accelerated computational fluid dynamics," *Proc. Natl. Acad. Sci.*, vol. 118, no. 21, p. e2101784118, 2021.

[2] M. Lino, S. Fotiadis, A. A. Bharath, and C. D. Cantwell, "Multi-scale rotation-equivariant graph neural networks for unsteady Eulerian fluid dynamics," *Phys. Fluids*, vol. 34, no. 8, p. 087110, 2022.

[3] R. Vinuesa and S. L. Brunton, "Enhancing computational fluid dynamics with machine learning," *Nat Comput Sci*, vol. 2, no. 6, pp. 358–366, 2022.

[4] 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 J.*, vol. 55, no. 12, pp. 4013–4041, 2017.

[5] H. Csala, S. T. M. Dawson, and A. Arzani, "Comparing different nonlinear dimensionality reduction techniques for data-driven unsteady fluid flow modeling," *Physics of Fluids*, vol. 34, no. 11, p. 117119, 2022.

[6] T. Murata, K. Fukami, and K. Fukagata, "Nonlinear mode decomposition with convolutional neural networks for fluid dynamics," *J. Fluid Mech.*, vol. 882, p. A13, 2020.

[7] J. Lumley, "The Structure of Inhomogeneous Turbulent Flows," in *Atmospheric Turbul. Radio Wave Propag.*, 1967, pp. 166–177.

[8] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016.

[9] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," 2017.

[10] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, "Learning to Simulate Complex Physics with Graph Networks," 2020.

[11] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," 2017.

[12] J. Ho, A. Jain, and P. Abbeel, "Denoising Diffusion Probabilistic Models," 2020.