Autoencoders for reconstructing interfacial multiphase flows

M Cutforth[1] and S Mirjalili[1,2]

[1] Department of Mechanical Engineering, Stanford University, Stanford, USA
[2] Department of Engineering Mechanics, KTH Royal Institute of Technology, Stockholm, Sweden

Keywords: autoencoders, 3D reconstruction, interfaces, multiphase flows

Abstract. In this work, we perform a comprehensive investigation of autoencoders for reduced-order modeling of three-dimensional multiphase flows. Focusing on the accuracy of reconstructing multiphase flow volume/mass fractions with a standard convolutional architecture, we examine the advantages and disadvantages of different interface representation choices (diffuse, sharp, level set). We use a combination of synthetic data with non-trivial interface topologies and high-resolution simulation data of multiphase homogeneous isotropic turbulence for training and validation. This study clarifies the best practices for reducing the dimensionality of multiphase flows via autoencoders. Consequently, this paves the path for uncoupling the training of autoencoders for accurate reconstruction and the training of temporal or input/output models such as neural operators (e.g., FNOs, DeepONets) and neural ODEs on the lower-dimensional latent space given by the autoencoders. As such, the implications of this study are significant and of interest to the multiphase flow community and beyond.

1. Introduction

Fluid flows are considered high-dimensional dynamical systems, where real-time prediction, control, and optimization are prohibitively expensive without some form of reduced-order modelling. A low-dimensional representation of the state of a fluid system is required for a reduced-order model (ROM). While there has been substantial progress in developing ROMs for single-phase flows [1-4], interfacial multiphase flows present a unique challenge. Multiphase flows consist of two or more immiscible fluid phases separated by a deformable interface. In this case, the state of the dynamical system strongly depends on the instantaneous location of the different phases, due to the discontinuity in density, viscosity, pressure, and other fields across the deformable phase interface. As such, it is crucial for the ROM to represent the shape of three-dimensional interfaces in multiphase flows accurately. In this work, we study the performance of convolutional autoencoders (AE) for this task. We reveal the impact of the interface representation type (diffuse, sharp, level-set function) and shape complexity on the performance of the AE. This work constitutes a robust building block for developing ROMs to predict temporal dynamics in multiphase flows. Finally, considering that low-dimensional representations of complex three-dimensional shapes are highly sought in many applications, including computer graphics, computer vision, and robotics, our findings have implications beyond multiphase flows [5-7].

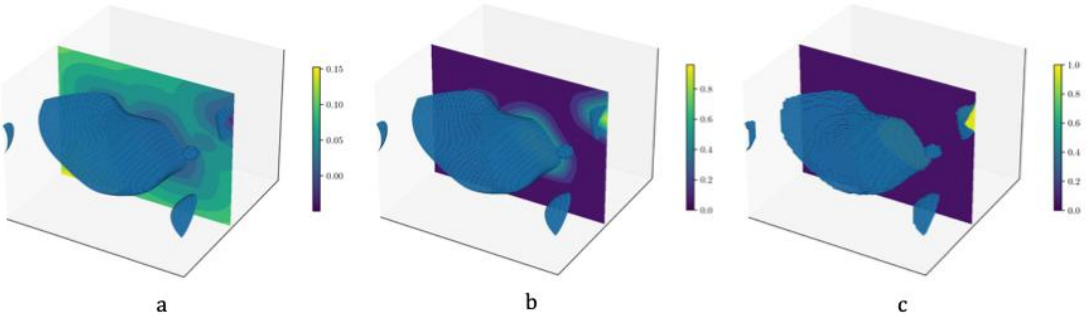

Figure 1: Illustration of the three interface representations compared in this study: (a) signed-distance (level-set) functions, (b) a diffuse (tanh) interface profile, and (c) a sharp-interface indicator function, showing the interface in one snapshot of the simulation of a droplet in homogeneous isotropic turbulence.

While substantial progress has been made in simulating (full-order modelling) multiphase flows using various approaches in the past few decades [8-9], these simulations are too expensive to allow for real-time predictions, control, or optimization. As such, ROMs for such multiphase flows are sought after.

Such models are scarce in the literature [10-14], especially when one seeks ROMs that aim to capture the interfaces and their evolution in such flows.

Here, we focus on the low-dimensional representation of the interfaces in multiphase flows via convolutional autoencoders (AEs). An AE is a neural network consisting of an encoder that maps the input onto a latent representation and a decoder that reconstructs the input from this representation. The encoder performs a nonlinear transformation, often reducing dimensionality, while the decoder attempts to approximate the original input as closely as possible. AEs have been successful in low-dimensional representation of single-phase flows [15-18], but their application to the reconstruction of multiphase flow states remains limited [10,13]. There are multiple ways to implicitly represent interfaces in multiphase flows via field variables. These include diffuse interface representations like phase field variables, sharp interface representations like volume fractions in Volume-of-Fluid models, and level-set functions, which are inherently available in level-set-based multiphase flow models. Since one can transform these representations into one another, we explore how the interface representation choice influences the performance of the AE. For training, we use a series of synthetic data sets with increasing interface complexity, as well as a data set consisting of snapshots from high-fidelity simulations of drops in decaying homogeneous isotropic turbulence (HIT) [14]. This also allows us to study the effect of the complexity of the interfacial structures on the reconstruction accuracy of the AEs. All in all, accurately representing flow states in a low-dimensional space is a prerequisite for most ROMs. This work elucidates the best practices and limitations of applying AEs for this purpose in multiphase flows.

2. Problem formulation

In this work, we are concerned with training convolutional AEs for the reconstruction of interfacial structures in two-phase systems. Interfacial structures can be captured with various field variables. Figure 1 shows three different choices for full-order representation of the same interfacial structure, where panel (a) displays the signed distance function (SDF), also known as the level-set function, typically denoted with $\boldsymbol{s}$. Panel (b) displays a diffuse interface representation using a so-called phase field variable given by $\boldsymbol{\phi} = (\mathbf{1} + \mathbf{tanh}(\boldsymbol{s}/\mathbf{2}\boldsymbol{\epsilon}))/\mathbf{2}$, where $\boldsymbol{\epsilon}$ is the interface thickness. Panel (c) gives a sharp interface representation, with an indicator function defined as $\boldsymbol{H} = (\mathbf{1} - \boldsymbol{sgn}(\boldsymbol{s}))/\mathbf{2}$. While these different fields represent the same interface (or phase) distributions, it is clear from figure 1 that they have vastly different properties. If we consider these fields $(\boldsymbol{s}, \boldsymbol{H}, \boldsymbol{\phi})$ as three-dimensional input fields denoted by $\boldsymbol{x}$, then the autoencoder, consisting of the encoding and decoding functions, $\boldsymbol{f}^{\boldsymbol{e}}_{\boldsymbol{\theta}_e}$ and $\boldsymbol{f}^{\boldsymbol{d}}_{\boldsymbol{\theta}_d}$, is trained to reconstruct $\boldsymbol{x}$ as accurately as possible. In other words, training the autoencoder is equivalent to finding the parameters $\boldsymbol{\theta_e}$and $\boldsymbol{\theta_d}$ that are the solution to the optimization problem given by $\mathbf{min}_{\boldsymbol{\theta_e},\boldsymbol{\theta_d}} |\boldsymbol{f}^{\boldsymbol{d}}_{\boldsymbol{\theta}_d}\left(\boldsymbol{f}^{\boldsymbol{e}}_{\boldsymbol{\theta}_e}(\boldsymbol{x})\right) - \boldsymbol{x}|^{\mathbf{2}}$. Note that $\boldsymbol{f}^{\boldsymbol{e}}_{\boldsymbol{\theta}_e}(\boldsymbol{x}) = \boldsymbol{z}$ is the latent variable, and the compression ratio is the ratio of the dimension of the input (output) field to the dimension of the latent variable.

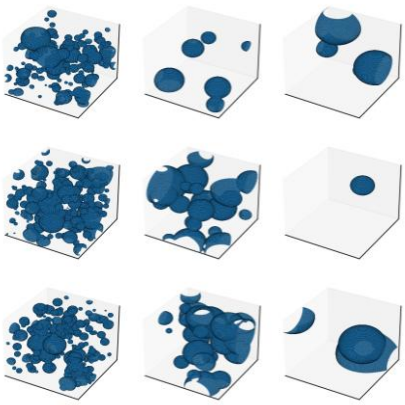

Figure 2: Interfacial contour visualisation in the synthetic dataset. Left column: $\boldsymbol{\mu}$=1, middle column: $\boldsymbol{\mu}$=2, right column: $\boldsymbol{\mu}$=2.5.

3. Data

To study the effect of interface complexity, we construct a series of three synthetic datasets, which are illustrated in figure 2. We parameterize the synthetic data using a single variable, denoted by $\boldsymbol{\mu}$, which controls the droplet size distribution and number of droplets per sample. In this way, as $\boldsymbol{\mu}$ decreases, we obtain datasets with progressively increasing topological and

Figure 3: Interfacial contour visualization in HIT dataset.

geometrical complexity, and finer spatial scales. Each sample is constructed by taking the union of $\boldsymbol{N_D}$ spherical droplets, where the droplet radii $\boldsymbol{r}$ are drawn independently from $\boldsymbol{r} \sim \mathbf{lognormal}(\boldsymbol{\mu}, {}^{\mathbf{1}}/_{\mathbf{2}})$. $\boldsymbol{N_D}$ is then drawn from a uniform distribution such that the total volume fraction is uniformly distributed in $[\mathbf{0}, \mathbf{1}]$. All samples are $\mathbf{64^3}$.

To incorporate a more realistic flow, we use snapshots from simulations of drops in decaying HIT that we previously used for training an ML model for drop breakup prediction [14]. The simulations were performed on a $\mathbf{256^3}$grid using a phase field (diffuse interface) approach [25,26]. Patches of size $\mathbf{64^3}$ were extracted from these snapshots, by randomly choosing 64 patch locations per volume and discarding empty patches (see figure 3). Knowing the interface thickness ($\boldsymbol{\epsilon_{sim}}$) used in the simulations, we compute the SDF ($\boldsymbol{s}$ in section 2), from which we construct input fields with various diffuse interface thicknesses, given by $\boldsymbol{\phi} = (\mathbf{1} + \mathbf{tanh}(\boldsymbol{s}/\mathbf{2}\boldsymbol{\epsilon}))/\mathbf{2}$, with the limit of $\boldsymbol{\epsilon} \to \mathbf{0}$ giving us the sharp interface representation ($\boldsymbol{H}$ in section 2).

4. Neural network architecture

Throughout this work we adopt a standard convolutional architecture; our objective is to compare interface representations rather than architecture optimization. The choice of a fully convolutional architecture [19] is motivated by our application to (potentially large) 3D volumes, as it enables a patch-based approach due to the translation equivariance of convolutional layers. Our network architecture follows the standard encoder-decoder structure described in Equation 1 and is based on the residual block illustrated on the right. Each layer in the encoder (decoder) consists of two such blocks in series, followed by a downsampling (upsampling) operation (figure 4). The choice of a ResNet-type architecture [20] along with weight standardised convolution layers is advantageous in the small batch size regime [21], which is necessarily the case in 3D due to the large feature maps which must be stored. Given an input of shape $\mathbf{1} \times \boldsymbol{H} \times \boldsymbol{W} \times \boldsymbol{D}$, we obtain a latent representation of shape $\boldsymbol{Z} \times \frac{\boldsymbol{H}}{\mathbf{2}^{\boldsymbol{N}}} \times \frac{\boldsymbol{W}}{\mathbf{2}^{\boldsymbol{N}}} \times \frac{\boldsymbol{D}}{\mathbf{2}^{\boldsymbol{N}}}$, where $\boldsymbol{N}$ is the number of downsample layers in the encoder. Unless otherwise specified, we use $\boldsymbol{N} = \mathbf{4}$ and $\boldsymbol{Z} = \mathbf{4}$, resulting in a compression ratio of 1024.

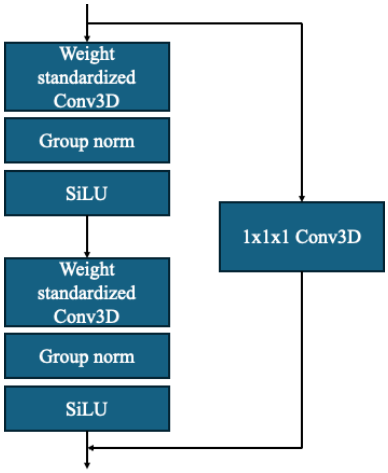

Figure 4: Residual block used in network architecture

5. Training

All models are trained using the Adam optimisation algorithm [22] and mean absolute error loss, with a batch size of 4 and learning rate of $\mathbf{10^{-4}}$. A learning rate range test is performed to confirm the suitability of this learning rate. The number of training epochs is held constant at 100 for the synthetic datasets and 15 for the HIT simulation dataset. This is confirmed to provide sufficient convergence. Each training run utilises 4 Nvidia V100 (16GB) GPUs in a data-parallel approach, running for less than 12 hours.

6. Testing and evaluation

Each case is evaluated using a held-out test set. Rather than standard $\boldsymbol{L^p}$ error norms, we adopt two metrics from the field of image segmentation. The evaluation of a (semantic) image segmentation is highly similar to the evaluation of an interfacial geometry. In both cases, the prediction consists of an arbitrary region represented by a binary mask. We transform all predictions to a sharp interface representation before computing metrics. The following two metrics offer a concise and understandable summary of the performance of this prediction. First, the Dice coefficient [23] provides a value in $[\mathbf{0}, \mathbf{1}]$ describing the overall volumetric agreement: $\boldsymbol{DSC}(\boldsymbol{X}, \boldsymbol{Y}) = \mathbf{2}\,|\boldsymbol{X} \cap \boldsymbol{Y}|/(|\boldsymbol{X}| + |\boldsymbol{Y}|)$. Second, the

Hausdorff distance [24] is a worst-case measure, analogous to the $\boldsymbol{L^{\infty}}$ norm. It is the maximum distance between the predicted and the true interface: $\boldsymbol{HD(\Gamma_X, \Gamma_Y) = \max\left(\sup_{x\in\Gamma_X}\inf_{y\in\Gamma_Y}|x-y|, \sup_{y\in\Gamma_Y}\inf_{x\in\Gamma_X}|x-y|\right)}$, where $\boldsymbol{\Gamma_X}$ represents an interface. We normalise this distance relative to the length scale of the domain, so a value of 1 corresponds to a distance of 64 cell widths (for a $\boldsymbol{64^3}$ grid).

7. Results

In the following section, we denote the level set / signed distance representation by "SDF", and the diffuse hyperbolic tangent with interface thickness $\boldsymbol{\epsilon}$ by "Tanh $\boldsymbol{\epsilon}$". Each volume represents a physical domain of $\boldsymbol{1\times1\times1}$, discretised to a $\boldsymbol{64^3}$ grid, and so $\boldsymbol{\epsilon = 1/64}$ corresponds to an interface thickness equal to one grid cell. Figures 5 and 6 show the reconstruction performance for the HIT dataset using various interface representations. Based on these results, the AEs perform well in reconstructing the HIT data for all interface representations, with the SDF and sharp representations being marginally the worst and best, respectively. Next, figure 7 shows performance metrics for the AEs trained and evaluated on the synthetic dataset. As expected, because of the increase in geometric complexity (see figure 2), the reconstruction accuracy degrades as $\boldsymbol{\mu}$ is decreased. Additionally, we see that similar to the HIT dataset, the Dice coefficient is worst for the SDF inputs and steadily improves as the interface is sharpened. Interestingly, while the SDF is again the worst option across the board, sometimes even failing to train successfully, the Hausdorff distance and volume error plots reveal a new trend. That is, for the more complex fields with the smallest drops, the sharpest representation is no longer the best option. Rather, a diffuse interface representation with $\boldsymbol{\epsilon \approx 1/32}$ seems to give the best results. This is confirmed in figure 8, where the drop size distributions in the input (ground truth) and reconstructed output fields are compared, showing how for $\boldsymbol{\mu = 1.0}$ the Tanh 1/64 representation gives the best results. To delve into this further, figure 9 compares the interfacial profiles for the predicted output fields from AEs trained using sharp and diffuse interface representations for a specific case in the test set. The AE trained with the sharp representation achieves a higher Dice coefficient by more accurately reconstructing the large features but ignoring the small features. Given that many multiphase flows are multiscale and involve poorly resolved features, this trade-off must be considered when choosing the interface

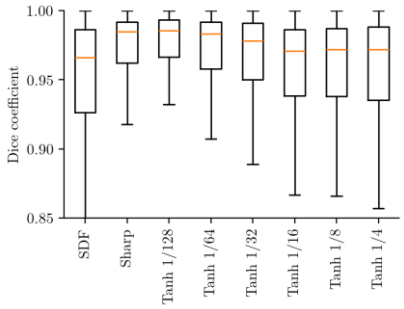

Figure 5: Dice coefficient for HIT data using various interface representations

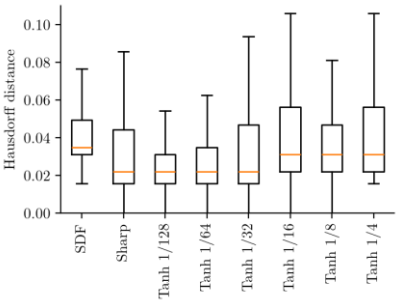

Figure 6: Hausdorff distance for HIT data using various interface representations

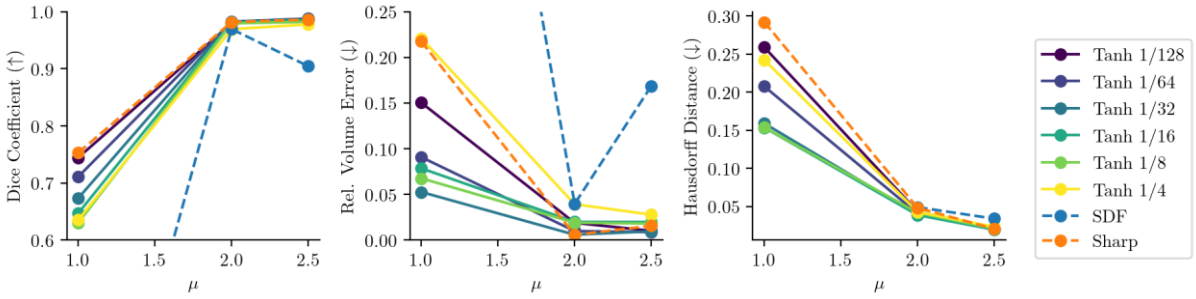

Figure 7: Comparison between interface representations for three performance metrics (mean Dice, mean relative volume error, mean Hausdorff distance) as a function of the interface complexity parameter $\boldsymbol{\mu}$ in the synthetic dataset. Smaller $\boldsymbol{\mu}$ corresponds to greater topological and geometric interface complexity.

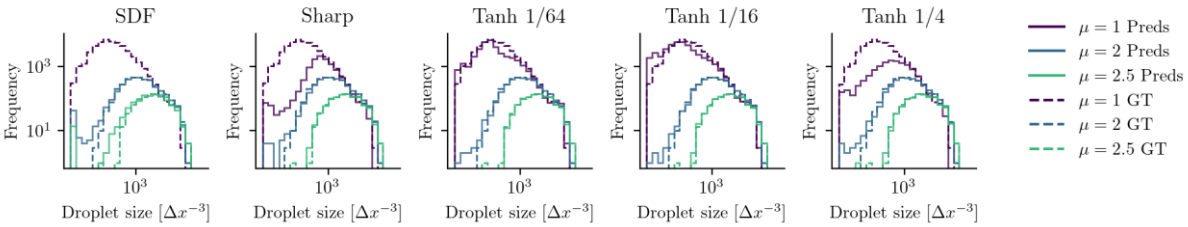

Figure 8: Comparison of predicted vs. true droplet size distributions for different interface representations. On the most complex dataset ($\mu = 1$), only the tanh representation with intermediate interface thicknesses provides predictions which preserve the droplet size distribution.

representation for training AEs. Finally, we investigated the generalisation ability of each interface representation by evaluating each combination of train/test dataset. Figure 10 shows that sharper interface representations transfer better to out-of-domain test datasets. Furthermore, sharp interface models trained only on synthetic data transfer surprisingly effectively to the HIT dataset, suggesting that synthetic data augmentation is an effective tool in this domain.

Overall, the under-performance of SDFs was unexpected. We repeated these experiments at different compression ratios between 100 and 8000 (not shown here), and found the same trends to hold. A limitation of our work is the use of a single model architecture in an empirical study, but we have chosen a standard architecture (with only minor enhancements for the small batch size inherent with 3D volumetric data) and leave an investigation of the effect of other model hyper-parameters for future work. Furthermore, due to the computational expense of training on 3D volumes, it was not possible to evaluate the variance in the training procedure using multiple random seeds. However, the presence of consistent trends with ϵ (diffuse interface width) suggests that this source of variance is smaller than the effect of interface representation.

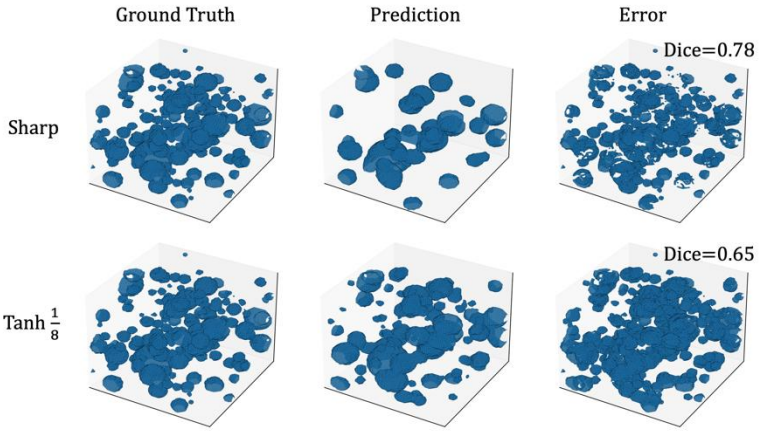

Figure 9: Comparison of reconstructed interface contour for sharp and diffuse representations. This illustrates how sharp representations faithfully reconstruct large-scale features while neglecting small-scale features.

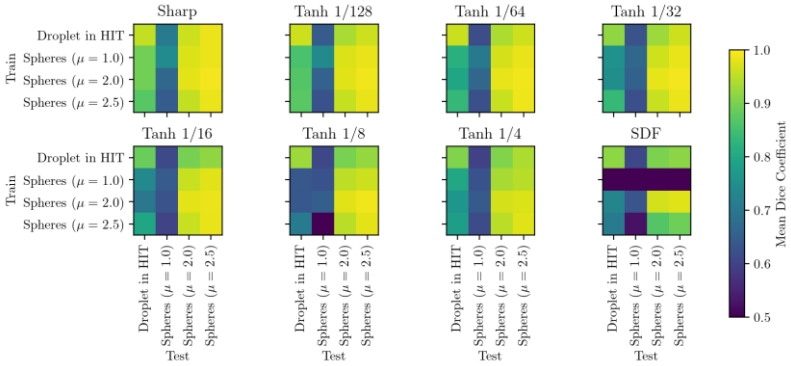

Figure 10: Comparison of generalization error when training and evaluating on different combinations of datasets

8. Conclusions

We have evaluated the performance of a fully-convolutional 3D autoencoder in reconstructing interfacial geometries. We have studied the influence of the choice of the interface representation using both simulated and synthetic datasets with differing complexity. We found that a signed distance function representation results in consistently worse interface reconstruction. On the other hand, a trade-off exists with diffuse representations, where a sharper representation results in a better reconstruction of large features at the expense of small-scale features, and a more diffuse interface representation (such as an interface width

of 2 – 4 times the grid spacing) results in small-scale features (e.g. small drops) being preserved better, and a more faithful reproduction of the droplet size distribution. Future work includes adding conservation constraints on the network output, investigation of variational AEs, reconstruction of other state variables, and predictions of temporal (dynamic).

Acknowledgments

We acknowledge financial support from the US Department of Energy's National Nuclear Security Administration via the Stanford PSAAP-III Center (DE-NA0003968). S. M. also acknowledges start-up funding from KTH Royal Institute of Technology and the Swedish e-science Research Center.

References

[1] Rowley, C. W., & Dawson, S. T. (2017). Model reduction for flow analysis and control. *Annual Review of Fluid Mechanics*, *49*(1), 387-417.
[2] Brunton, S. L., Noack, B. R., & Koumoutsakos, P. (2020). Machine learning for fluid mechanics. *Annual review of fluid mechanics*, *52*(1), 477-508.
[3] Vinuesa, R., & Brunton, S. L. (2022). Enhancing computational fluid dynamics with machine learning. *Nature Computational Science*, *2*(6), 358-366.
[4] Rozza, G., Stabile, G., & Ballarin, F. (Eds.). (2022). *Advanced reduced order methods and applications in computational fluid dynamics*. Society for Industrial and Applied Mathematics.
[5] Bagautdinov, T., Wu, C., Saragih, J., Fua, P., & Sheikh, Y. (2018). Modeling facial geometry using compositional VAEs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3877-3886).
[6] Dai, A., Ruizhongtai Qi, C., & Nießner, M. (2017). Shape completion using 3d-encoder-predictor CNNs and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5868-5877).
[7] Groueix, T., Fisher, M., Kim, V. G., Russell, B. C., & Aubry, M. (2018). A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 216-224).
[8] Tryggvason, G., Scardovelli, R., & Zaleski, S. (2011). *Direct numerical simulations of gas–liquid multiphase flows*. Cambridge university press.
[9] Mirjalili, S., Jain, S. S., & Dodd, M. (2017). Interface-capturing methods for two-phase flows: An overview and recent developments. *Center for Turbulence Research Annual Research Briefs*, *2017*(117-135), 13.
[10] Wiewel, S., Becher, M., & Thuerey, N. (2019, May). Latent space physics: Towards learning the temporal evolution of fluid flow. In *Computer graphics forum* (Vol. 38, No. 2, pp. 71-82).
[11] Kani, J. N., & Elsheikh, A. H. (2019). Reduced-order modeling of subsurface multi-phase flow models using deep residual recurrent neural networks. *Transport in Porous Media*, *126*, 713-741.
[12] Haas, T., Schubert, C., Eickhoff, M., & Pfeifer, H. (2020). BubCNN: Bubble detection using Faster RCNN and shape regression network. *Chemical Engineering Science*, *216*, 115467.
[13] Zhu, L. T., Chen, X. Z., Ouyang, B., Yan, W. C., Lei, H., Chen, Z., & Luo, Z. H. (2022). Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors. *Industrial & Engineering Chemistry Research*, *61*(28), 9901-9949.
[14] Cundy*, C., Mirjalili*, S., Laurent, C., Ermon, S., Iaccarino, G., & Mani, A. (2024). A physics-informed machine learning model for the prediction of drop breakup in two-phase flows. *International Journal of Multiphase Flow*, *180*, 104934.
[15] Solera-Rico, A., Sanmiguel Vila, C., Gómez-López, M., Wang, Y., Almashjary, A., Dawson, S. T., & Vinuesa, R. (2024). β-variational autoencoders and transformers for reduced-order modelling of fluid flows. *Nature Communications*, *15*(1), 1361.
[16] Brunton, S. L., & Kutz, J. N. (2022). *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press.
[17] Kneer, S., Sayadi, T., Sipp, D., Schmid, P., & Rigas, G. (2021). Symmetry-aware autoencoders: s-PCA and s-nlPCA. *arXiv preprint arXiv:2111.02893*.
[18] Csala, H., Dawson, S., & Arzani, A. (2022). Comparing different nonlinear dimensionality reduction techniques for data-driven unsteady fluid flow modeling. *Physics of Fluids*, *34*(11).
[19] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. (2015). "Fully convolutional networks for semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*
[20] He, Kaiming, et al. (2016). "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition.*
[21] Qiao, Siyuan, et al. (2019). "Micro-batch training with batch-channel normalization and weight standardization." *arXiv preprint arXiv:1903.10520*
[22] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
[23] Dice, Lee R. (1945). "Measures of the amount of ecologic association between species." *Ecology* 26.3 297-302.
[24] Huttenlocher, Daniel P., Gregory A. Klanderman, and William J. Rucklidge. (1993). "Comparing images using the Hausdorff distance." *IEEE Transactions on pattern analysis and machine intelligence* 15.9: 850-863.
[25] Mirjalili, S., Ivey, C. B., & Mani, A. (2020). A conservative diffuse interface method for two-phase flows with provable boundedness properties. *Journal of Computational Physics*, *401*, 109006.
[26] Mirjalili, S., & Mani, A. (2021). Consistent, energy-conserving momentum transport for simulations of two-phase flows using the phase field equations. *Journal of Computational Physics*, *426*, 109918.