RESEARCH

Model order reduction for the 1D Boltzmann-BGK equation: identifying intrinsic variables using neural networks

Julian Koellermeier[1] · Philipp Krah[2] · Julius Reiss[3] · Zachary Schellin[3]

Abstract

Kinetic equations are crucial for modeling non-equilibrium phenomena, but their computational complexity is a challenge. This paper presents a data-driven approach using reduced order models (ROM) to efficiently model non-equilibrium flows in kinetic equations by comparing two ROM approaches: proper orthogonal decomposition (POD) and autoencoder neural networks (AE). While AE initially demonstrate higher accuracy, POD's precision improves as more modes are considered. Notably, our work recognizes that the classical POD model order reduction approach, although capable of accurately representing the non-linear solution manifold of the kinetic equation, may not provide a parsimonious model of the data due to the inherently non-linear nature of the data manifold. We demonstrate how AEs are used in finding the intrinsic dimension of a system and to allow correlating the intrinsic quantities with macroscopic quantities that have a physical interpretation.

Keywords Model order reduction · Data-driven methods · Kinetic equations · Neural autoencoder networks · Proper orthogonal decomposition · Sod shock tube · Boltzmann-BGK

1 Introduction

Kinetic equations are widely used in science and engineering (Koellermeier and Torrilhon 2018; Maes et al. 2023; McClarren and Hauck 2010; Struchtrup and Torrilhon 2008). They allow the modeling of deviations from an equilibrium model which is given by an underlying macroscopic equation like the Euler equations, providing detailed insight into fundamental physical processes (Torrilhon 2016). However, kinetic equations are often characterized by a large dimensional phase space consisting of physical space and the velocity space of gas particles, making them computationally expensive to solve and sometimes even unfeasible for realistic applications (Torrilhon 2016).

✉ Julian Koellermeier
  j.koellermeier@rug.nl

  Philipp Krah
  philipp.krah@univ-amu.fr

1 Bernoulli Institute, University of Groningen, Groningen, The Netherlands

2 Institut de Mathématiques de Marseille, Aix-Marseille Université, Marseille, France

3 Institute of Fluid Mechanics and Engineering Acoustics, Technical University of Berlin, Berlin, Germany

Solving kinetic equations is, therefore, more costly than the computation using simpler equilibrium models like the Euler equations. Investing in solving kinetic equations is thus only beneficial if large deviations from equilibrium are present that cannot be predicted by the equilibrium models (Torrilhon 2016). Striking a balance between a fast but inaccurate equilibrium solver and a slow but accurate non-equilibrium solver remains an open challenge.

In the field of non-equilibrium gas flows, several standard methods to efficiently discretize the high dimensional phase space exist. While particle-based Monte Carlo methods have recently been used even in the transition and early slip regime, they are best suited in the free flight regime and typically become more computationally costly in denser regions of only moderate non-equilibrium unless special techniques are used (Debrabant et al. 2017; Degond et al. 2011; Garcia et al. 1999). The straightforward discrete velocity method (DVM) uses a pointwise discretization of the velocity space, potentially leading to a large number of equations (Mieussens 2000; Mieussens et al. 2012; Brull and Prigent 2020). Specially tailored moment models are based on the expansion of the particle distribution function and lead to a set of extended fluid dynamical equations (Torrilhon 2016). However, it is by no means clear a-priori how many equations are sufficient and which variables

are optimal (Torrilhon 2015). Our work aims to address this challenge by providing a proof-of-concept for a data-driven solution model of flows in different non-equilibrium regimes.

To tackle the computational complexity of kinetic equations, recently, other data-driven reduced order models (ROM) have been introduced, enabling reductions in computational complexity by orders of magnitudes (Bernard et al. 2018; Einkemmer 2019; Einkemmer et al. 2021a, b). Two different approaches have been followed in the literature. The classical offline-online decomposition as used by Bernard et al. (2018) involves a two-stage procedure. In the offline stage, the full order model (FOM) is assessed to create a database, which is then utilized to generate a data-dependent basis through proper orthogonal decomposition (POD). This basis allows an efficient description of the FOM on a low-dimensional linear subspace during the online phase. On the other hand, the online adaptive basis method called dynamic low-rank approximation (Koch and Lubich 2007) constructs the low dimensional linear basis during the online phase itself, eliminating the need to evaluate the expensive FOM. It has been successfully applied to kinetic equations in Einkemmer (2019); Einkemmer et al. (2021a, b). However, the additional complexity of updating the basis during the evolution makes it less online efficient than the classical offline-online approach shown in Koellermeier et al. (2023) for a shallow water moment model.

In this work, we adopt the same offline strategies as in Bernard et al. (2018). Specifically, we sample data for a classical test case called Sod shock tube using a discrete velocity model as our FOM and compare the compression of the linear reduced subspace created by POD with a non-linear description provided by neural autoencoder networks. Neural networks, based on the universal approximation theorem (Pinkus 1999), allow for the approximation of a wide range of function classes and appear promising in identifying the intrinsic dimension of a system. However, the non-linear relation between macroscopic model equations and the discrete velocity model hinders the determination of these dimensions using linear reduction methods like the POD.

This paper aims to utilize these data-driven model reduction techniques to reduce the number of describing variables and equations and determine how many and which variables are useful in specific test cases. For non-vanishing Knudsen number, we expect to need more non-equilibrium variables with corresponding balance laws, while in the limit of vanishing Knudsen number, we expect to recover the Euler equations, given by conservation laws for mass, momentum, and energy. To the knowledge of the authors, this is the first paper aiming to bridge the gap between equilibrium and non-equilibrium flows using neural networks in this way.

The long-term objective of this line of work is to enable dynamically adapting the model by varying the number of variables during the online phase, paving the way for more efficient and accurate model adaptive simulations of kinetic equations.

The organization of the paper is as follows: in Sect. 2, we introduce the 1D model problem and the reference data used for model reduction. Section 3 describes the two model reduction techniques used in this study: Proper Orthogonal Decomposition (POD) and Autoencoder Networks. The results are presented in Sect. 4, and the paper concludes with a summary in Sect. 5.

2 The Boltzmann-BGK model and data

This paper considers a proof-of-concept of using reduced models for the solution approximation of the 1D Boltzmann-BGK equation (Bhatnagar et al. 1954) for monoatomic, ideal gasses

$$\partial_t f + c\partial_x f = \frac{1}{\tau}(f_M - f), \tag{1}$$

which is a potentially high-dimensional equation for the unknown probability density function $f(t, x, c)$, where $t \in \mathbb{R}^+$ is the time, $x \in \mathbb{R}$ is the spatial variable, and $c \in \mathbb{R}$ the microscopic particle velocity. For simplicity we consider the one-dimensional case in this paper, but the results can be extended to the multi-dimensional case.

Computing solutions and generating data of the Boltzmann-BGK model is essential for industrial and scientific applications, but often so computationally prohibitive that a large number of test cases is not feasible. To reduce time and cost during the data generating process, experiments or numerical simulations can be replaced by reduced-order models (ROMs).

For standard continuum flows the widely-used Euler equations can be applied, but more rarefied regimes require different extended fluid dynamical models. Rarefaction levels are distinguished with aid of the Knudsen number Kn defined by the ratio of the mean free path length of the particles λ over a reference length l as

$$\text{Kn} = \frac{\lambda}{l}. \tag{2}$$

The right-hand side of the BGK collision operator Eq. (1) models the relaxation with relaxation time $\tau \in \mathbb{R}^+$ toward the equilibrium Maxwellian distribution $f_M(t, x, c)$ given by

$$f_M(t, x, c) = \frac{\rho(t, x)}{(2\pi RT(t, x))^{\frac{3}{2}}} \exp\left(-\frac{(c - u(t, x))^2}{2RT(t, x)}\right), \tag{3}$$

where $\rho(t, x)$, $v(t, x)$ and $T(t, x)$ are density, bulk velocity, and temperature of the flow, respectively, and R is the universal gas constant. In this work, we consider the relaxation time τ a parameter and set it equal to the Knudsen number, $\tau = \text{Kn}$,

however, the relaxation time can also be changed, e.g., to depend on the gas density and temperature in addition.

For practical computations, we consider macroscopic moments of the distribution function, which are given by multiplying the distribution function with the co-called collision invariants $(1, c, \frac{1}{2}c^2)$ and integrating in velocity space

$$\rho(t,x) = \int f(t,x,c)\,dc, \tag{4}$$

$$\rho(t,x)u(t,x) = \int cf(t,x,c)\,dc, \tag{5}$$

$$E(t,x) = \int \frac{1}{2}c^2 f(t,x,c)\,dc, \tag{6}$$

where E denotes the total energy. The temperature $T(t,x)$ and the pressure $p(t,x)$ can be obtained by

$$T(t,x) = \frac{2E(t,x)}{3\rho(t,x)} - \frac{u(t,x)^2}{3} \quad \text{and} \quad p(t,x) = \rho(t,x)T(t,x). \tag{7}$$

Figure 1 illustrates the relation between the macroscopic moments and the distribution function $f(t, x, c)$ at a certain position in time and space. The density $\rho(t,x)$ is the integral of the distribution function, which is centered around the macroscopic velocity $u(t, x)$, and the mean deviation is related to the temperature $T(t, x)$.

The Boltzmann-BGK equation Eq. (1) is in equilibrium when $f = f_M$. Multiplying the equilibrium solution with the collision invariants and integrating in velocity space, one finds the Euler equations of classical gas dynamics
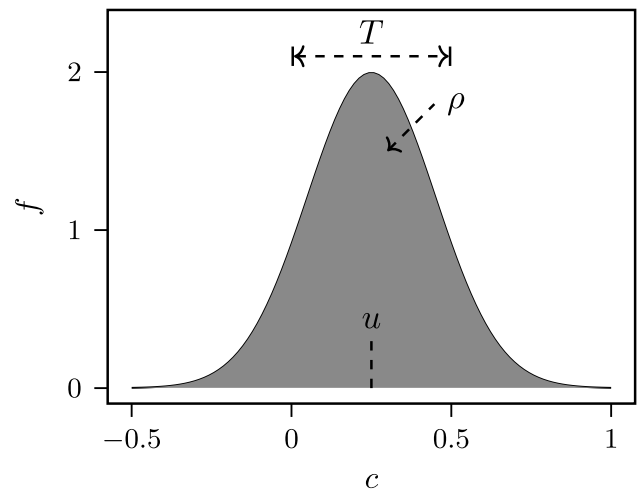
$$\partial_t \rho + \partial_x(\rho u) = 0, \tag{8}$$

$$\partial_t(\rho u) + \partial_x(\rho u^2 + p) = 0, \tag{9}$$

$$\partial_t E + \partial_x(u(E + p)) = 0, \tag{10}$$

which are conservation laws for mass, momentum, and energy, respectively.

For distribution functions out of equilibrium, for example due to a larger relaxation time τ and a significantly large Knudsen number Kn, the Euler equations do not give accurate results. In this case, additional equations can be used, which are derived by the so-called method of moments (Torrilhon 2016). This effectively leads to an extended set of equations, called moment model. It is possible to preserve important properties like hyperbolicity with moment models (Fan et al. 2016). The additional equations (for example for the heat flux and higher-order moments) add complexity, but allow for more accurate solutions (Koellermeier and Torrilhon 2017; Torrilhon 2015). However, it is often unclear a-priori, how many equations are needed for an efficiently accurate and computationally feasible solution. In this work, we aim to give a proof-of-concept for a data-based identification of the necessary number of variables, called the intrinsic physical dimension.

2.1 Sod shock tube test case and reference data

Sod's shock tube is a well-established test case in the field of rarefied gasses (Koellermeier and Torrilhon 2017). It uses discontinuous initial conditions based on equilibrium values

$$\begin{cases} (\rho_L, u_L, p_L) = (1, 0, 1) & \text{if } x < 0.5, \\ (\rho_R, u_R, p_R) = (0.125, 0, 0.1) & \text{if } x > 0.5, \end{cases} \tag{11}$$

corresponding to a jump in density and pressure at $x = 0.5$ due to a diaphragm at that position, which is removed at time $t = 0$.

The problem setup at $t = 0$ is shown in Fig. 2, which is split into two regions left and right of the diaphragm.

While Sod's shock tube has a seemingly simple setup, it is nonetheless challenging due to the discontinuous initial profile and the emerging non-equilibrium conditions. In most other test cases, the nature of the non-equilibrium flow conditions are similar. However, in some test cases without discontinuous initial data or shocks like the 2D driven cavity test case, some intrinsic variables of Sod's shock tube might not be relevant while others are more important. This could be investigated in further work.

For the generation of reference data we employ a discrete velocity method (DVM) (Mieussens 2000), which uses a pointwise microscopic velocity space discretization

Fig. 1 Illustration of the macroscopic moments corresponding to an example distribution function

Fig. 2 Problem setup for the 1D Sod shock tube. A diaphragm at the center is initially separating the domain in two regions, where initial conditions for density ρ, macroscopic velocity u, and pressure p are indicated
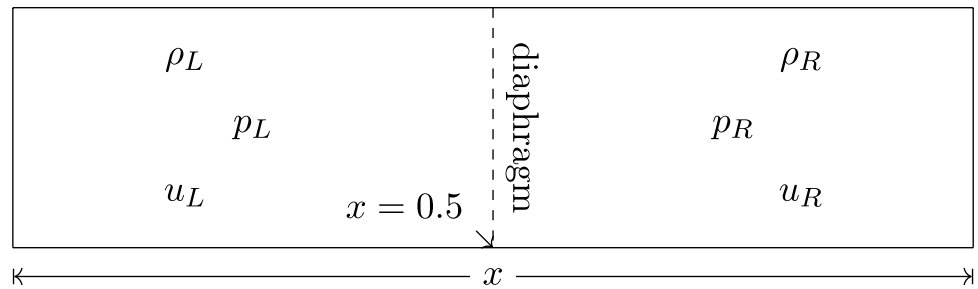
$$\partial_t f_k(t,x) = -(c_k)\partial_x f_k(t,x) + \frac{1}{\tau}\Big(M_{f_k}(t,x) - f_k(t,x)\Big), \qquad (12)$$

where a uniform grid in velocity space is considered with $c_k = k\Delta c$ to discretize the distribution function $f_k(t,x) = f(t,x,c_k)$, for some $k \in \mathbb{Z}$. After a subsequent discretization in space, the DVM Eq. (12) leads to a coupled ODE system in time than can be solved with standard methods.

For the numerical reference data, we use $N_x = 200$ spatial points $x_k \in [0,1]$, $N_c = 40$ discrete velocities $c_j \in [-10, 10]$ and $N_t = 25$ time steps $t^n \in [0, 0.12]$ summarized in Table 1. It is possible to choose another range for the discrete velocity points, but in typical applications the range of the bulk velocity is not known, such that one has to include a safety margin. We therefore chose the domain $[-10, 10]$. The goal of the model order reduction is now to reduce the complexity of the computation using lower dimensional models. For that matter, it is not relevant what the actual error of the numerical reference data is or if is fully converged. It is fair to say that a full reference solution might easily take into account more spatial points, time steps, and discrete velocities, which makes it even more necessary to reduce the complexity.

For the model order reduction later, we consider two different Knudsen numbers for Sod's shock-tube test case: Kn = 0.00001 for a small Knudsen number in the hydrodynamic regime and Kn = 0.01 for a relatively large Knudsen number in the rarefied regime.

To understand the behavior of the reference solutions for non-vanishing Knudsen numbers, we first describe the solution for vanishing Knudsen number in equilibrium, i.e., Kn = 0, which can be obtained using the method of characteristics and the Rankine Hugoniot jump conditions connecting the states before and after the shocks (LeVeque 2002).

Starting from the initial condition in Fig. 3a, the solution evolves for $t > 0$ and five regions are formed that are depicted in Fig. 3b (Sod 1978). A rarefaction wave is moving to the left between x_1 and x_2. The contact discontinuity is located at x_3, where the macroscopic velocity u and the pressure p are continuous in contrast to the density ρ and the energy E. x_4 is the position of the shock wave.

In non-equilibrium, i.e., for solutions evolving with Knudsen numbers Kn > 0, the solution does not have discontinuities due to the finite relaxation time τ. Figure 3c shows the reference solutions $f(t, x, c)$ at $t_0 = 0$, $t_1 = 0.06$ and $t_3 = 0.12$ for the two levels of rarefaction considered in this paper: Kn = 0.00001 and Kn = 0.01. Increasing the Knudsen number leads to a smoother transition from region 1 to region 5 with a less pronounced shock front.

3 Methods

In this section, we present two common methods used for reducing the dimensionality of high dimensional data: (1) the proper orthogonal decomposition (POD) and (2) neural autoencoder networks (AE). The methods will be used to parameterize the high dimensional data stemming from the DVM simulation, using a linear mapping in case of the POD and a non-linear mapping in case of AE. Although the classical POD model order reduction approach shows that linear mappings are sufficient to describe the non-linear solution manifold of the BGK equation to a good accuracy (Bernard et al. 2018), it is in general not sufficient to determine a parsimonious model of the full model data, since the data manifold can be non-linear. Here, neural autoencoder networks can be used as they are capable to find the intrinsic dimension of a system.

3.1 Proper orthogonal decomposition

POD (Sirovich 1987) approximates the data with the help of dyadic pairs:

$$f(x, t, c_i) \approx \sum_{k=1}^{r} \hat{f}_k(x, t)\psi_k(c_i) \qquad \text{for } r \ll N_c. \qquad (13)$$

The pairs $\{(\hat{f}_k(x,t), \psi_k(c_i))\}_{k=1,\dots,r}$ are the structures in the data that contain the most energy and they are chosen to minimize the gap between the data and the reconstruction Eq. (13). In the following, $\psi_k(c_i)$ are termed POD-modes and $\hat{f}_k(x,t)$ the corresponding reduced variables.
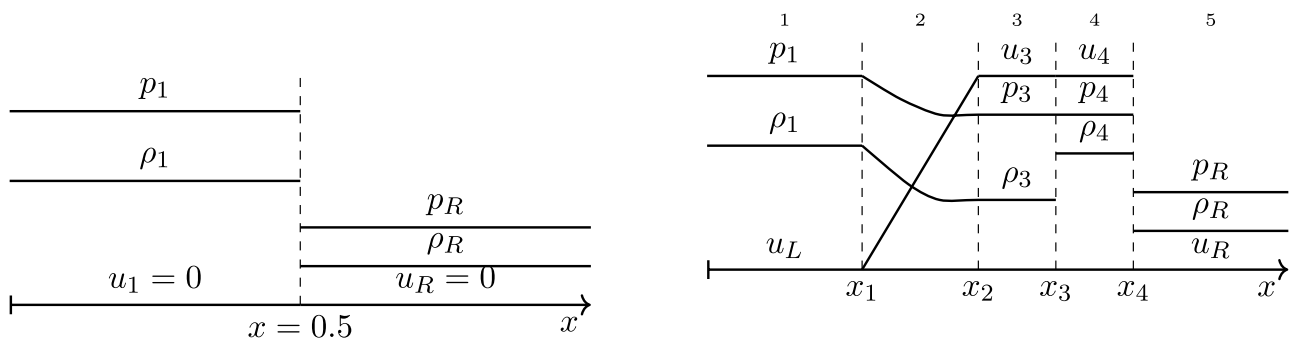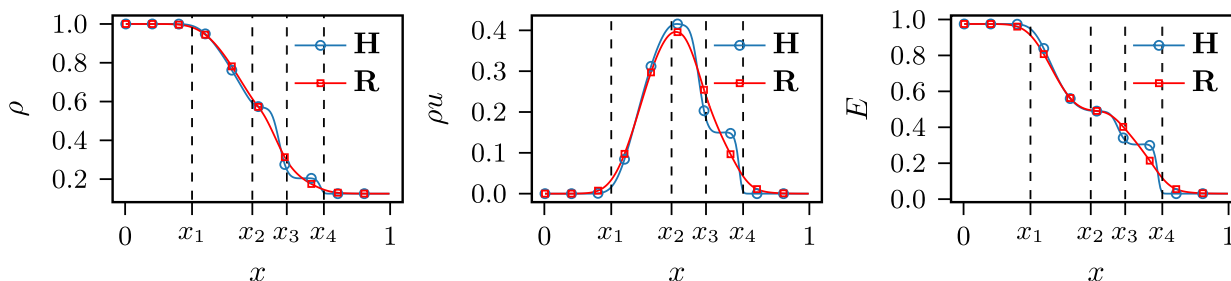
(a) Sod's shock tube ($t = 0$) containing two regions with corresponding initial conditions.

(b) Sod's shock tube ($t > 0$) containing the rarefaction wave, the contact discontinuity and the shock wave.
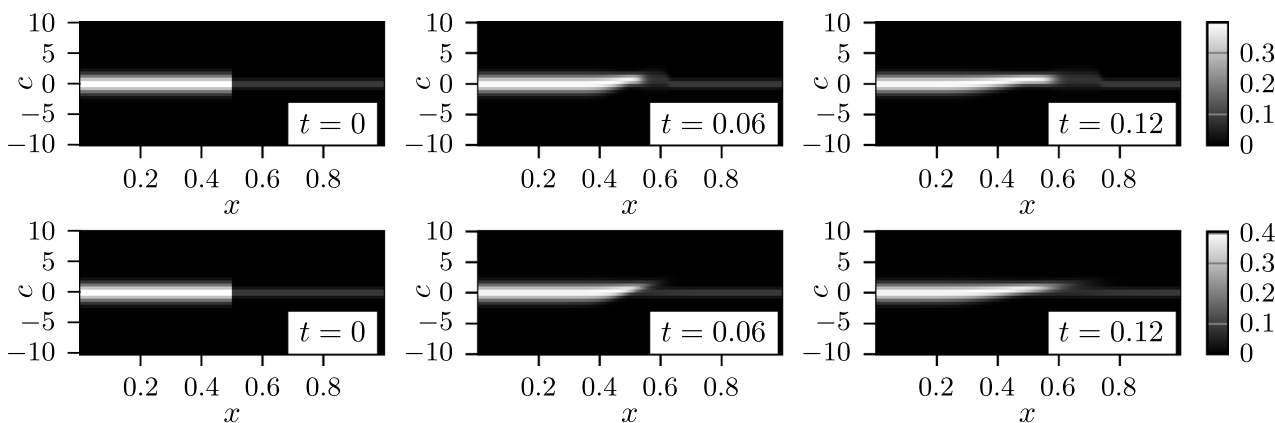
(c) Discrete reference solutions $f(t, x, c)$ for Kn = 0.00001 (top row) and Kn = 0.01 (bottom row), for times $t_0 = 0$ (left), $t_1 = 0.06$ (center) and $t_2 = 0.12$ (right).

(d) Macroscopic quantities ρ, ρu, E ($t = 0.12$) for Kn = 0.00001 (**H**) and Kn = 0.01 (**R**).

Fig. 3 Sod shock tube and reference data. Initial conditions (**a**); equilibrium solution (**b**); reference solutions in rarefied and hydrodynamic regime (**c**); macroscopic quantities at $t = 0.12$ s (**d**)

Variable	Number of nodes i	Domain extension	Step size (uniform)
x	200	[0, 1]	0.005
c	40	[− 10, 10]	≈ 0.51282051
t	25	[0, 0.12]	0.005

For notation, we define $f^{(i)}(x,t) = f(x,t,c_i)$ and the vectors $f(x,t) = (f^{(1)}(x,t),\dots,f^{(N_c)}(x,t))$ and $\psi_k = (\psi_k(c_1),\dots,\psi_k(c_{N_c}))$. The proper orthogonal decomposition computes the solution of the minimization problem:

$$\min_{\psi_k} \left\| f(x,t) - \sum_{k=1}^{r} \langle f(x,t), \psi_k \rangle \psi_k \right\|_2^2 \quad \text{such that} \quad \langle \psi_k, \psi_l \rangle = \delta_{kl}. \tag{14}$$

Technically one can solve this optimization problem using a singular value decomposition (SVD) of the so-called snapshot matrix (Kunisch and Volkwein 1999):

$$\mathbf{F} = \begin{bmatrix} f^{(1)}(x_1,t_1) & \cdots & f^{(N_c)}(x_1,t_1) \\ \vdots & \ddots & \vdots \\ f^{(1)}(x_{N_x},t_1) & \cdots & f^{(N_c)}(x_{N_x},t_1) \\ f^{(1)}(x_1,t_2) & \cdots & f^{(N_c)}(x_1,t_2) \\ \vdots & \ddots & \vdots \\ f^{(1)}(x_{N_x},t_{N_t}) & \cdots & f^{(N_c)}(x_m,t_{N_t}) \end{bmatrix} \in \mathbb{R}^{(N_x N_t) \times N_c}. \tag{15}$$

The snapshot matrix collects the time and space discrete distribution function in its columns. Each column holds the time-spatial values for a different discrete velocity. Performing an SVD factorizes $\mathbf{F}$ as

$$\mathbf{F} = \mathbf{\Phi \Sigma \Psi^T}, \tag{16}$$

with diagonal matrix $\mathbf{\Sigma} = \text{diag}(\sigma_1,\dots,\sigma_m)$, $m = \min(N_x N_t, N_c)$, containing the singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$ and $\mathbf{\Phi} \in \mathbb{R}^{N_x N_t \times m}$, $\mathbf{\Psi} \in \mathbb{R}^{N_c \times m}$ are orthogonal matrices containing the left and right singular vectors, respectively.

The first r columns of the truncated $\mathbf{\Psi}_r = [\psi_1,\dots,\psi_r] \in \mathbb{R}^{N_c \times r}$ contain the POD modes in Eq. (13). Together with $\mathbf{\Sigma}_r = \text{diag}(\sigma_1,\dots,\sigma_r)$ and the r leading left singular vectors $\mathbf{\Phi}_r \in \mathbb{R}^{(N_x N_t) \times r}$ they yield the rank r-term approximation $\mathbf{F}_r$ of the snapshot matrix given by

$$\mathbf{F}_r := \mathbf{\Phi}_r \mathbf{\Sigma}_r \mathbf{\Psi}_r^T. \tag{17}$$

According to the Eckart-Young-Mirsky theorem (Eckart and Young 1936; Mirsky 1960) $\mathbf{F}_r$ is the best rank r approximation and the resulting error in the Frobenius norm is rigorously computed from the trailing singular values

$$\|\mathbf{F} - \mathbf{F}_r\|_F^2 = \sum_{k=r+1}^{m} \sigma_k^2. \tag{18}$$

A common choice for r is to truncate after a certain energy percentage is reached in the reduced system compared to the full system:

$$E_{\text{cum}} = \frac{\|\mathbf{F}_r\|_F}{\|\mathbf{F}\|_F} = \frac{\sum_{k=1}^{r} \sigma_k^2}{\sum_{k=1}^{m} \sigma_k^2}. \tag{19}$$

In this paper, POD is used to compare with the autoencoder from the next section.

3.2 Autoencoders

Neural networks, particularly autoencoder networks, have become widely used tools for dimension reduction (Van Der Maaten et al. 2009), image segmentation (Minaee et al. 2021) and denoising (Tian et al. 2020), time series prediction (Han et al. 2019). A comprehensive introduction to autoencoder networks can be found in Goodfellow et al. (2016). Here we only summarize briefly the common idea of autoencoder networks and give the specific details of our implementation.

An autoencoder aims to reproduce the input data while compressing it through an information bottleneck. It consists of two main components:

- The encoder, denoted as g_{enc}, maps the input data f to points $\hat{f}$ in a lower-dimensional latent space: $g_{\text{enc}} : \mathbb{R}^M \to \mathbb{R}^r, f \mapsto \hat{f} = g_{\text{enc}}(f), r \ll M$.
- The decoder, denoted as g_{dec}, reconstructs the input space from the latent representation: $g_{\text{dec}} : \mathbb{R}^r \to \mathbb{R}^M, \hat{f} \mapsto g_{\text{dec}}(\hat{f}) = \tilde{f}, r \ll M$.

Note that the dimension of the latent space is denoted by r, to match the rank of the POD approximation.

The autoencoder is defined as the composition of both parts: $\tilde{f} = g_{\text{dec}}(g_{\text{enc}}(f))$. For our purpose, we identify the discrete velocity space as the input dimension $M = N_c$. Thus, the autoencoder maps each time-spatial value of the distribution function $f(x,t) \in \mathbb{R}^{N_c}$ onto a smaller latent space $\hat{f}(x,t) \in \mathbb{R}^r$, which parameterizes the necessary physical information of the system.

The goal of the optimization procedure is to determine g_{dec} and g_{enc} such that the reconstruction error over a set of training/testing data contained in $\mathbf{F}$ is minimized. The reconstruction error is defined as:

$$\mathcal{L} = \frac{1}{N_c} \| f(x,t) - g_{\text{dec}} \circ g_{\text{enc}} \circ f(x,t) \|_2^2. \tag{20}$$

The reconstruction error is the sum of the two-norm of the discrete velocities vector of the difference between the input data f and the reconstructed data $\tilde{f}$ that has been squeezed through the informational bottleneck. The assumption is, that if the original data can be represented well while the information went through a smaller latent space, there exists a physical law in the latent space that describes the system sufficiently. The intrinsic latent dimension $r = p^*$ which is sufficient to describe the data is then called the *intrinsic physical dimension* similar to the intrinsic dimension defined in Lee and Carlberg (2020). Such a reduced model is then termed parsimonious because it explains the data with a minimum number of variables.

In the training procedure, the functions g_{enc} and g_{dec} are determined by trainable parameters of the network, referred to as weights and biases. The networks are constructed using a composition of layers $g_{\text{enc}} = L_1 \circ L_2 \circ \ldots \circ L_N$. Typically, each layer $L_n : \mathbb{R}^i \to \mathbb{R}^o$ in the network consists of an affine linear mapping $x \mapsto h_n(\mathbf{W}_n x + b_n)$, where $\mathbf{W}_n \in \mathbb{R}^{o,i}$ represent the weights, $b_n \in \mathbb{R}^o$ denote the biases, and h_n are predefined non-linear functions. The configuration of the input and output dimensions i and o for each layer, the choice of activation function, and the number of layers collectively determine the architecture of the network. The choice of these so-called hyper-parameters is often difficult and a matter of trial and error.

Architecture In our studies we have exploited *fully connected neural autoencoder networks* (FCNN) and convolutional autoencoder networks (CNN). However, in this manuscript, we restrict ourselves to the results of the fully connected network, since it gave structurally the best results. We note that the poor performance of the CNN is most probably attributed to the small input size ($M = N_c = 40$), which proves insufficient for the effective functioning of a convolutional autoencoder. It is important to emphasize that CNNs are typically employed in scenarios involving larger input sizes than N_c, especially in the context of image processing. Therefore, the results may change if the velocity space is finer resolved. We have studied a variety

of different activation functions, hidden layers, batch sizes and depths of the network. The best results concerning the validation error and acceptable training time were obtained by the network defined in Table 2. A comprehensive study of the parameter optimization is attached to the manuscript in Sect. 1.

Training Before the training, we initialize the weights of the network using the standard initialization implemented in `pytorch`. Thus the weights are randomly uniformly distributed between $m^{-1/2}$ and $m^{1/2}$ with m being the number of input nodes in the layer. Our network is trained by splitting the data consisting of $N_x \times N_t$ samples in a testing and training set with an 80/20 split over 3000 epochs using a batch size of 4. In each epoch, the network is updated using the Adam optimizer with a learning rate of 10^{-5}. More information about hyperparameters and training of the network can be found in the "Appendix" section.

4 Results

We reconstruct the full order model (FOM) solution with the help of POD and an autoencoder, the FCNN, for which the selection of hyperparameters and the training are described in the previous section. Note that we apply both model reduction techniques to reconstruct both the rarefied reference data and the hydrodynamic reference data. The goal is to later determine the intrinsic dimension of the data for both cases. We therefore compare the two dimension reduction techniques through different measures. The intrinsic variables obtained from POD and the FCNN will be referred to as **h** and **r**, where the former describes the intrinsic variables when reducing the hydrodynamic data and the latter when reducing the rarefied data.

To compare the results we define the L_2-error

$$\mathcal{E}_{\text{rel}} = \frac{\|\mathbf{F} - \tilde{\mathbf{F}}\|_2}{\|\mathbf{F}\|_2}, \tag{21}$$

where $\mathbf{F}$ the reference data is given in Eq. (15) and the reconstructed data $\tilde{\mathbf{F}}$ is either $\mathbf{F}_r$ in case of the POD or the FCNN predictions with r latent variables for every (x, t, c) in the data set.

As computation time is highly dependent on the specific implementation and hardware configuration, we refrain from providing precise runtime measurements here. Most of the computation time is needed for the offline computation and training of the autoencoder, as detailed in the "Appendix". The online phase has virtually negligible computational cost in comparison to the offline phase as usual in an online-offline MOR approach, see also Koellermeier et al. (2023).

Table 2 Hyper-parameters of fully connected autoencoder network (FCNN) for the hydrodynamic and rarefied regime

Parameter	Hydrodynamic	Rarefied
Layer sizes	$[N_c, 30, r, 30, N_c]$	$[N_c, 40, r, 40, N_c]$
Activation function	ELU	ReLU
Loss function	MSE Eq. (20)	MSE Eq. (20)
Optimizer	Adam	Adam
Learning rate	10^{-5}	10^{-5}
Epochs	3000	3000
Batch size	4	4

(a) Hydrodynamic regime.   (b) Rarefied regime.
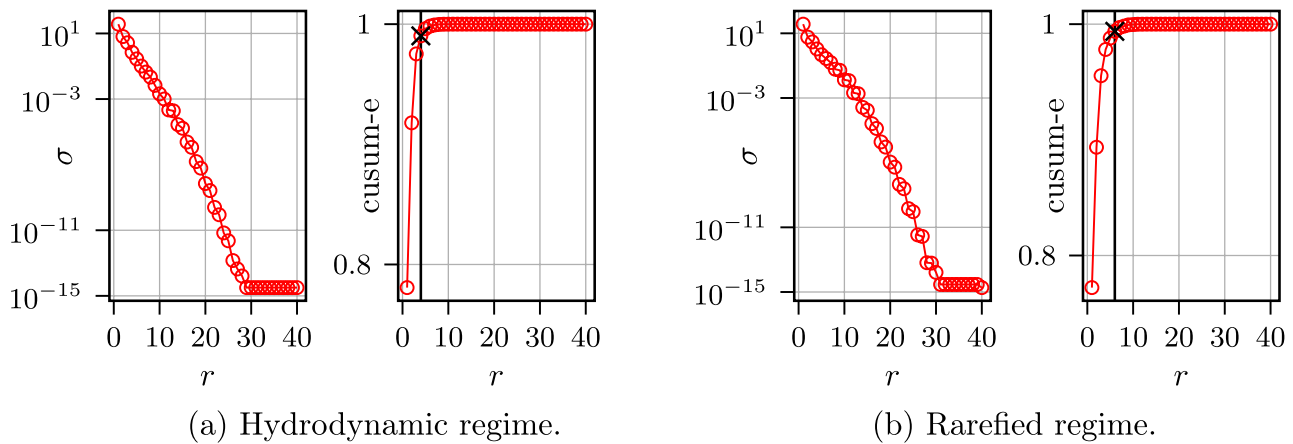
Fig. 4 Singular value decay σ and cumulative energy increase for the number of singular variables k in the hydrodynamic regime (**a**) and in the rarefied regime (**b**). A black cross marker corresponds to over 99% cumulative energy

4.1 Singular value decay of reference data

As a first step, we perform a POD with the hydrodynamic data and with the rarefied data. The obtained singular values σ, as well as the cumulative energy (cusum-e) defined in Eq. (19), are shown in Fig. 4. As expected, more modes are necessary in the rarefied regime compared to the hydrodynamic regime. With a total of $p^{\mathrm{POD}} = 4$ intrinsic variables, a cumulative energy of over 99% can be achieved for the hydrodynamic regime. The cumulative energy of the singular values of the rarefied regime only reaches above 99% with $p^{\mathrm{POD}} = 6$ singular values.

For the POD we define the *intrinsic dimension* p^{POD} as the smallest truncation rank r of the reduced system, at which the cumulative energy E_{cum} defined in Eq. (19) reaches 99%. Although this choice is arbitrary it is a common practice in classical model order reduction to truncate Eq. (13), whenever 99% of the cumulative energy is reached.

In Fig. 4, we further see that the rate at which the singular values drop is approximately exponential in both regimes, which has been also observed by Bernard et al. (2018). Consequently, a rapid decay of the Kolmogorov N-width is indicated. Note that the singular value decay is similar for both domains but not exactly the same, thus leading to an expected increase in the number of intrinsic variables in the rarefied regime necessary to achieve similar L_2-errors.

It is important to note that the parameter p^{POD} is not expected to precisely match the actual intrinsic dimension p^* of the solution manifold. The intrinsic dimension represents the minimum number of variables required to accurately describe the system's exact solution manifold. This discrepancy arises because the solution manifold is fundamentally nonlinear, making it challenging to adequately capture with a parsimonious linear approximation.

For the FCNN, the intrinsic dimension is defined as the smallest number of intrinsic variables that minimizes the error. In well-trained models, the FCNN's intrinsic dimension should ideally align with p^*.

From a fluid mechanics perspective, the hydrodynamic regime theoretically requires only $p^* = 3$ intrinsic variables. This is because near-equilibrium flows in this regime can be effectively characterized by three macroscopic quantities: density ρ, macroscopic velocity u, and total energy E, as outlined in Eqs. (8)–(10), see also Bernard et al. (2018) and Koellermeier and Torrilhon (2017).

Conversely, the rarefied regime demands a larger intrinsic dimension, denoted as p^*. This is due to the need for more than only the equilibrium Maxwellian distribution function to describe the microscopic velocities adequately. Therefore, we initially set $p^* = 3$ intrinsic variables (h) for the FCNN in the hydrodynamic case and choose $p^* = 5$ intrinsic variables (**r**) for the rarefied regime. This choice aligns with extended fluid dynamic models as described in Koellermeier and Torrilhon (2017) and Torrilhon (2016).

We note that each FCNN with a different latent space dimension r needs to be trained separately. This is different from the POD, where the decomposition is only performed once. Thereafter the approximation quality is given by the truncation rank r.

4.2 Variation of the number of intrinsic variables

The variation of the number of intrinsic variables r in Fig. 5 sheds light on the performance of the autoencoder with different bottleneck layer sizes. In the case of the POD r is the truncation rank of the decomposition Eq. (13) and the latent space dimension in case of the FCNN. To this end, r is varied for both the POD and the FCNN over
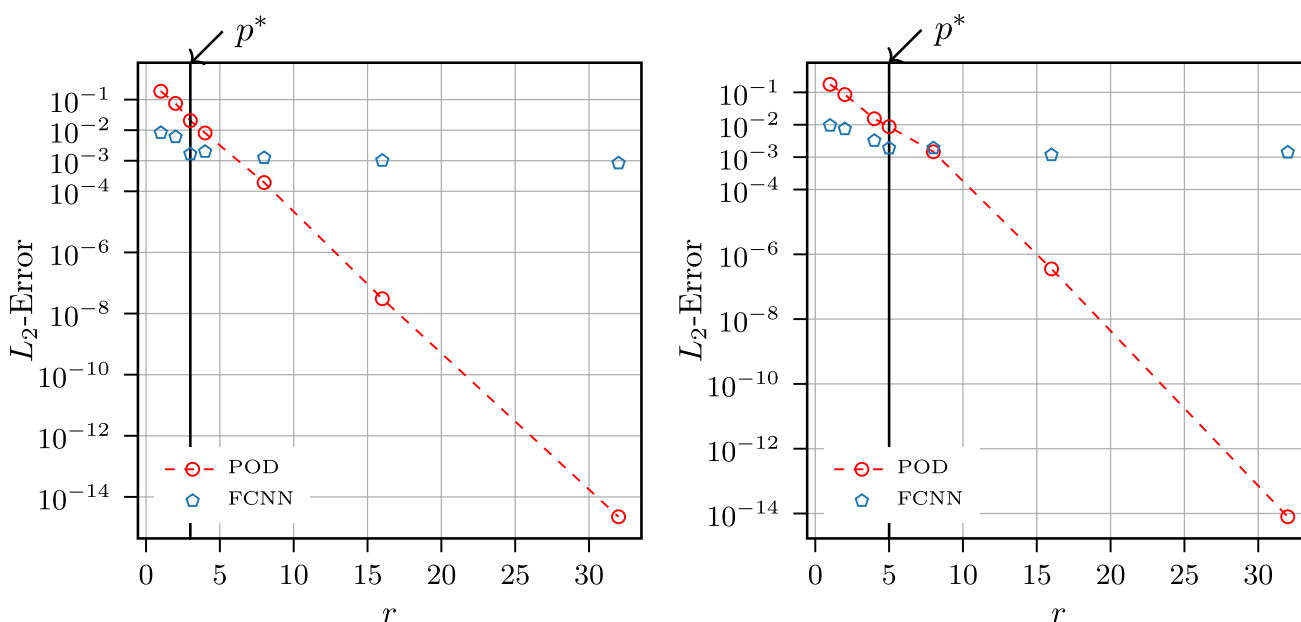
Fig. 5 The L_2-error over the variation of the latent space dimension/truncation rank r using FCNN/POD for the hydrodynamic regime (left) and the rarefied regime (right)

$r \in \{1, 2, 3, 4, 8, 16, 32\}$ for the hydrodynamic case and over $r \in \{1, 2, 4, 5, 8, 16, 32\}$ for the rarefied case. We note that the loss of information when applying POD goes exponentially to zero with increasing r, which is not surprising when consulting the *Eckard-Young Theorem* (Eckart and Young 1936). Note that the FCNN is retrained for each different r. By changing r, i.e. widening the bottleneck layer, a gain or loss of capacity occurs that can be connected to stability during training.

Both for the hydrodynamic and the rarefied regime, POD initially yields a larger error than the FCNN for a small number of intrinsic variables r. Not surprisingly, the POD accuracy increases with the number of singular values taken into account until the error reaches machine precision. The FCNN error decreases as well and then reaches a plateau, with a typical remaining error due to the network architecture and training. For the previously identified values $p^* = 3$ in the hydrodynamic case and $p^* = 5$ in the rarefied case, the FCNN results in a more accurate approximation than the POD.

We note that when testing the FCNN against POD and fixing r the FCNN is limited by the estimation error of the training and performs under its abilities. However, POD uses five to six times more parameters than the FCNN while the deterministic character enables POD to achieve any possible accuracy, which was not observed with the neural network.

In the following, we consider the intrinsic variables with constant values $p^* = 3$ in the hydrodynamic case and $p^* = 5$ in the rarefied case.

This leads to the number of trainable parameters of the POD and the FCNN shown in Table 3. We can see that the FCNN achieves a relatively small error with a small number of parameters in comparison with the POD for this choice of the number of intrinsic variables.

Next, a qualitative analysis with the actual reconstructions is presented. From computations of the L_2-error over time t, which are not shown due to conciseness, it became clear that the time step that contributes most to the error is the last time step in the case of POD, while the FCNN distributes the error more evenly over all time steps.

Table 3 Amount of parameters used to reconstruct f, the number of intrinsic variables p and the corresponding L_2-error for POD and FCNN, both for hydrodynamic (**H**) and rarefied (**R**) regimes

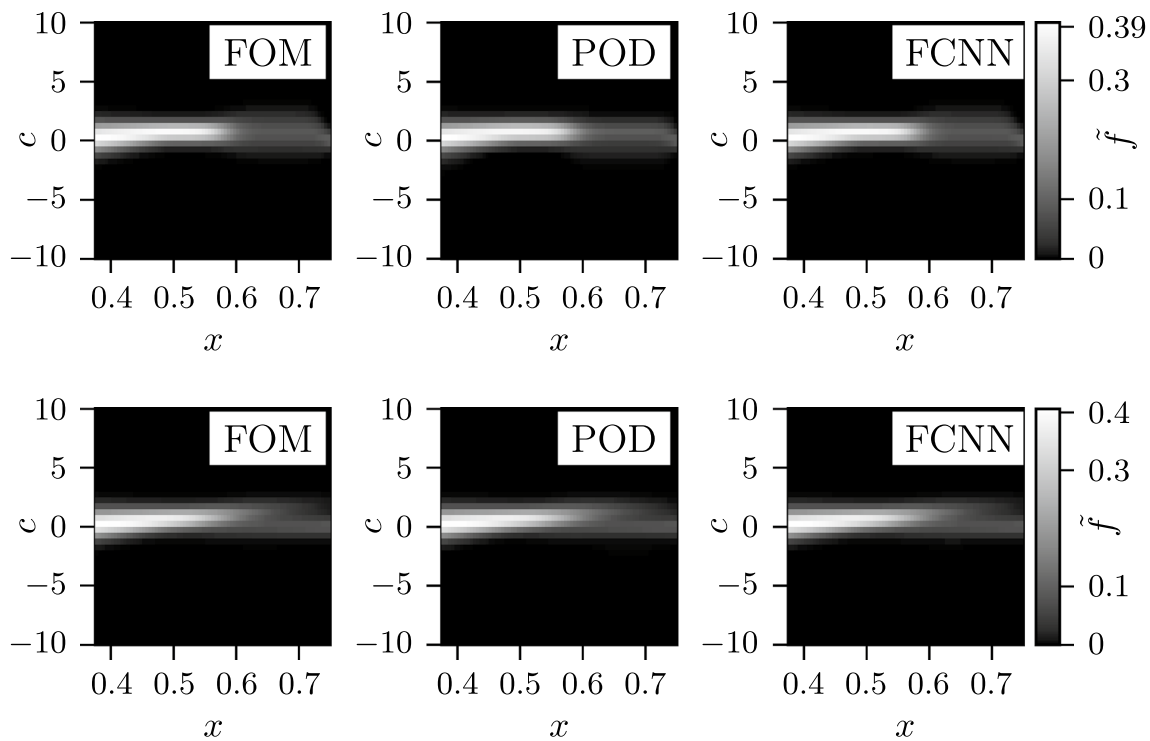| Algorithm	Parameters θ		Int. variables p		L_2-error	
	H	**R**	**H**	**R**	**H**	**R**
POD	15,129	25,225	3	5	0.0205	0.0087
FCNN	2683	3725	3	5	0.0008	0.0009

Fig. 6 Comparison of the FOM solutions f (left column) with reconstructions $\tilde{f}$ obtained from POD (middle column) and FCNN (right column) at end time $t = 0.12s$ for $x \in [0.375, 0.75]$ in the hydrodynamic regime (top row) and the rarefied regime (bottom row)

4.3 Reconstruction quality

The reconstructed solutions compared to the full order model (FOM) at the final time step $t = 0.12s$ are given in Fig. 6. Because of the small overall errors indicated in Table 3 both the POD and the FCNN reproduce the FOM solution without any visible differences at first sight.

As shown in Fig. 7, the more subtle information loss from the model reduction can unfold in actual differences in the macroscopic quantities ρ, ρu and E. Overall, Fig. 7 shows that the errors are larger for the hydrodynamic regime (top row), most notably for the momentum and energy of the POD model close to the contact discontinuity. However, the position of the shock is well approximated. In contrast, the FCNN model yields a very good agreement in the hydrodynamic case. For the rarefied regime, both models approximate the FOM solution very well. The lack of sharp shock structures in the full model and the increased intrinsic dimension $p^* = 5$ combined seem to notably influence the accuracy.

4.4 Conservation properties

The physical consistency of the reduced $\tilde{f}$, in terms of conservation of mass, momentum, and energy, is a critical criterion for its validity. Hence, conservation properties

are analyzed in the following. We note that conservation of mass, momentum, and energy is not directly built in using a specifically tailored loss function. Even though we can expect to recover some conservation properties as they are implicitly built into the numerical reference data. The investigation of different loss functions to improve upon this is left for future work.

We investigate the conservation properties by means of the derivative of the cell-averaged conserved quantities mass, momentum, and total energy, defined exemplary as

$$\frac{\mathrm{d}}{\mathrm{d}t} \int \rho(x,t)\,\mathrm{d}x\Delta t = \overline{\rho}, \tag{22}$$

for the mass. Note that a derivative $\overline{\rho} = 0$ denotes conservation of mass, for example. We expect the conservation of mass and total energy, while the momentum increases at a constant rate, due to the boundary conditions of the test case, featuring a larger pressure on the left-hand side of the domain.

Figure 8 shows the evolution of the derivatives of mass, momentum, and total energy as a function of time for the hydrodynamic regime (top row) and the rarefied regime (bottom row).

Indeed, Fig. 8 indicates that conservation of mass and energy is achieved with reasonable accuracy for the FCNN,
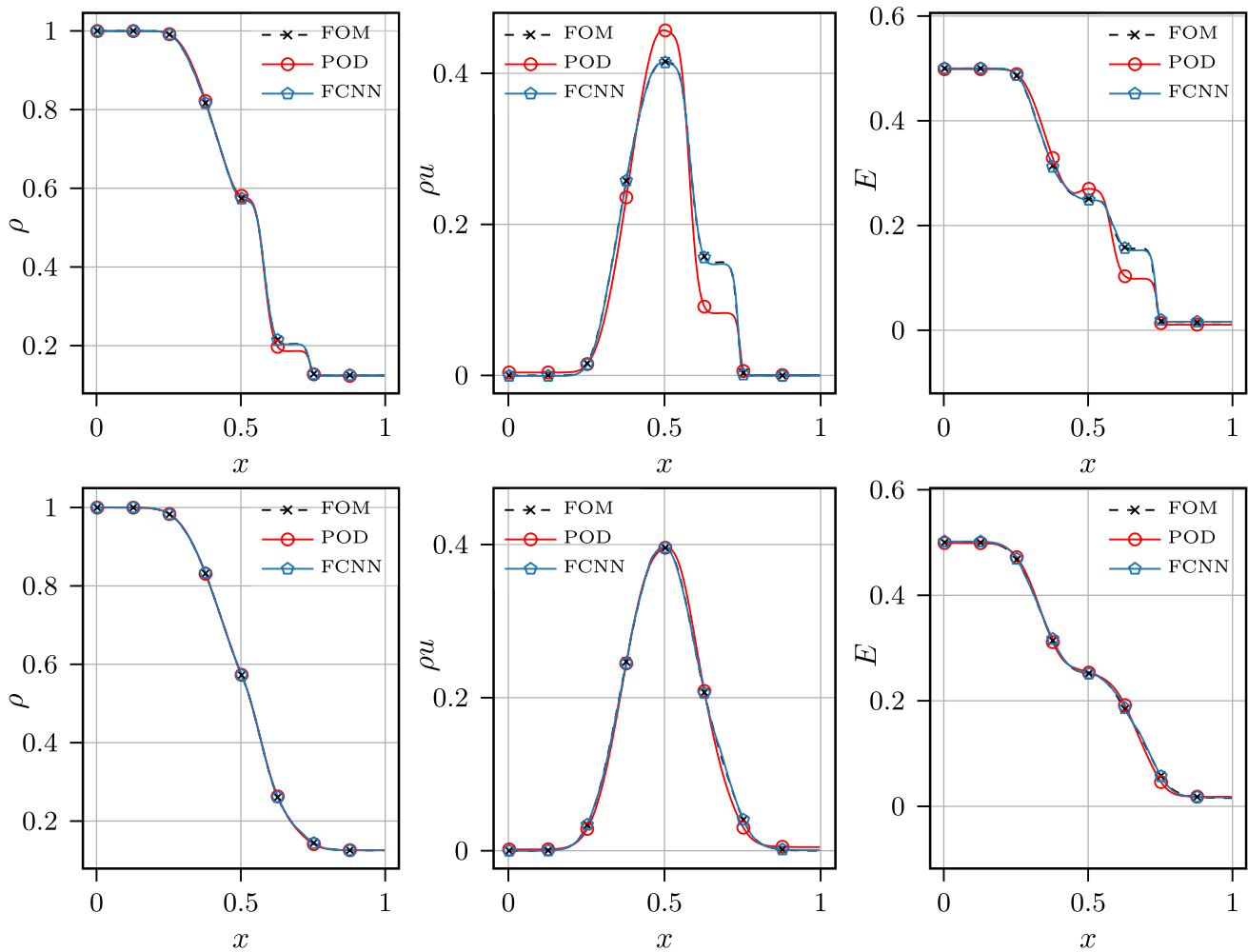
Fig. 7 Comparison of FOM macroscopic quantities ρ, ρu and E with POD and the FCNN for the hydrodynamic regime (top row) and the rarefied regime (bottom row) at time $t = 0.12s$

while the error is larger for the POD reconstruction for both regimes. Also, the increase in momentum of the full order model (FOM) is accurately described by the FCNN with a larger error for the POD method for both regimes. Overall, the errors are slightly smaller for the rarefied case compared to the hydrodynamic case, which might be due to the higher capacity of the neural network and more modes for the POD using five intrinsic variables in the rarefied regime.

4.5 Physical interpretability

An important question in the context of model reduction with neural networks is the interpretability of the results, because of the usual black-box nature of neural networks (Fan et al. 2020). Especially when benchmarking neural networks for model order reduction with POD, evaluating the interpretability of the intrinsic variables is important, since POD by construction achieves a so-called physically

interpretable decomposition of the input data (Brunton and Kutz 2019) as outlined previously.

Following the assumption that the hydrodynamic case can be fully described in terms of three macroscopic quantities and that the rarefied case is reasonably describable in a similar way with an extended set of five variables, we test the intrinsic variables **h** and **r** for similarities and investigate if they match any macroscopic quantity. Two related macroscopic quantities, namely the temperature T and macroscopic velocity u, are added to the three macroscopic variables ρ, ρu, and E. In Figs. 9 and 12, these are plotted first over the whole domain of x and t and for the end time $t = 0.12s$ for both regimes. Similarly, we plot both the FCNN's and the POD's first 3 intrinsic variables **h** of the hydrodynamic case and 5 intrinsic variables **r** of the rarefied test case

$$\mathbf{h} = [h_0(x,t), h_1(x,t), h_2(x,t)],$$

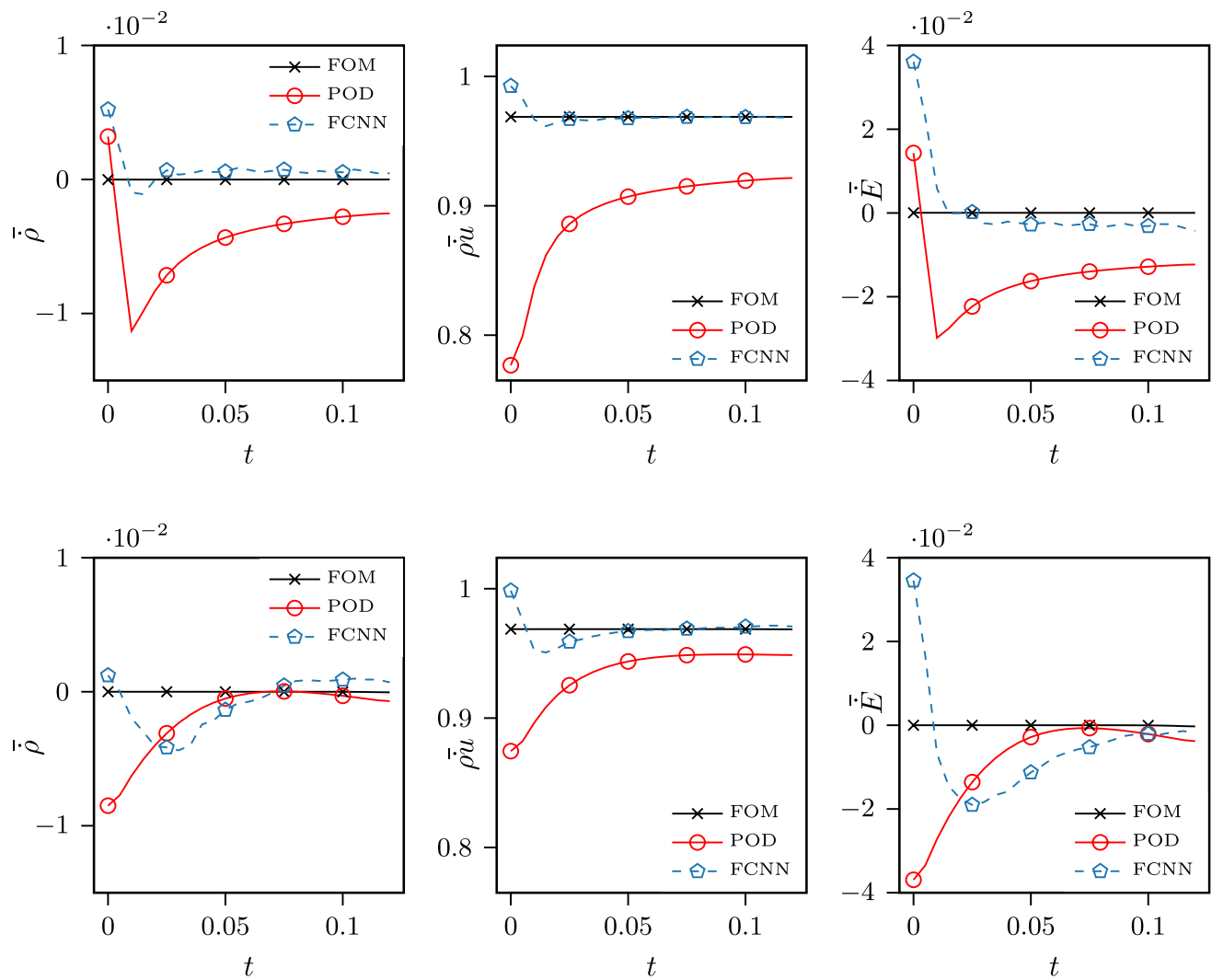$$\mathbf{r} = [r_0(x,t), r_1(x,t), r_2(x,t), r_3(x,t), r_4(x,t)],$$

Fig. 8 Comparison of the conservative properties of reconstructions obtained from POD and the FCNNs with the conservative properties of the FOM solution using the mass, momentum, and energy depicted in Figs. 10 and 13 respectively.

derivative. A value of 0 indicates conservation. FCNN approximately conserves mass and energy, while the momentum increases with the correct rate of the test case

Strikingly, most intrinsic variables of the FCNN in Figs. 10 and 13 and the POD in Figs. 11 and 14 appear to be a combination of the five intrinsic variables shown in Figs. 9 and 12, respectively. In particular consider FCNN in the hydrodynamic case by comparing Figs. 9 and 10. The rarefaction wave, shock wave, and contact discontinuity, which can be identified in h_0 reflect a combination of those found in the density ρ and the total energy E. Furthermore, h_1 seems to model the negative momentum ρu, with different boundary values. The temperature T appears linked to h_2, where the same fluctuation appears. Similar results hold for the POD variables in Fig. 11, where especially the first two variables closely resemble the density ρ and the momentum ρu. Interestingly, this does not relate to very good conservation properties of the POD in comparison with FCNN as

shown in the previous section and Fig. 8. Considering the FCNN in the rarefied case, we compare Figs. 12 and 13. Here r_3 clearly reflects the shape of the density ρ. Moreover, the peak of the velocity u can be observed in r_0. For the other intrinsic variables of r_1, r_2 and r_4 a clear discernability of macroscopic quantities is difficult to observe and might require linear or nonlinear combinations. Additionally, those intrinsic variables may resemble non-equilibrium variables not present among the macroscopic variables. Considering the POD results in Fig. 14, we again see a very good agreement of the first intrinsic variables with density and momentum.

For more physical insight into the relation between macroscopic variables and intrinsic variables, the Pearson correlation between all variable combinations is computed in Goodfellow et al. (2016) for the FCNN. Note that we expect
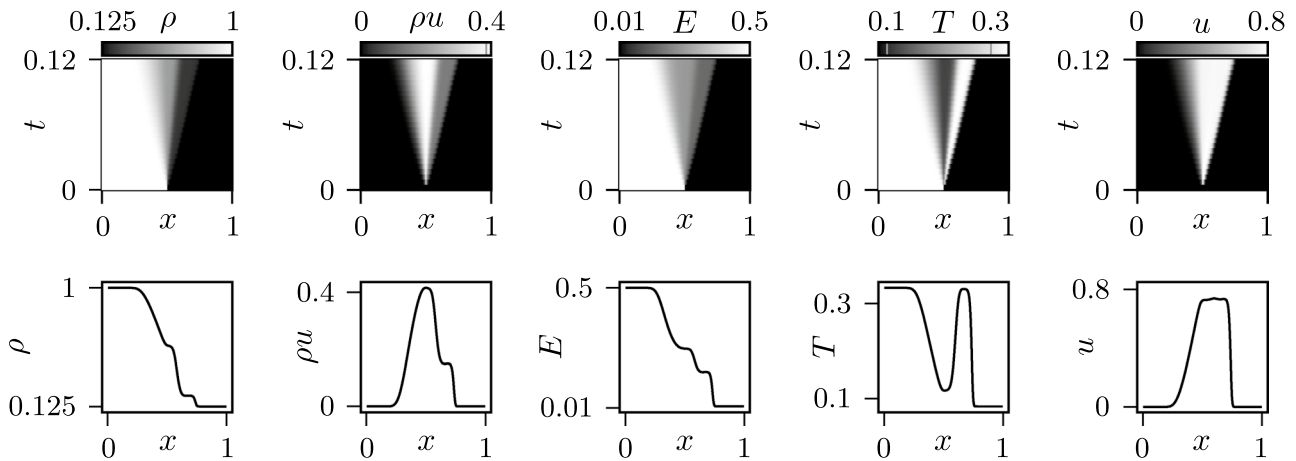
Fig. 9 Macroscopic quantities of the hydrodynamic case obtained by the FOM. Density ρ, momentum ρu, total energy E, temperature T, and velocity u over time t and space x in the top row and at $t = 0.12$ in the bottom row
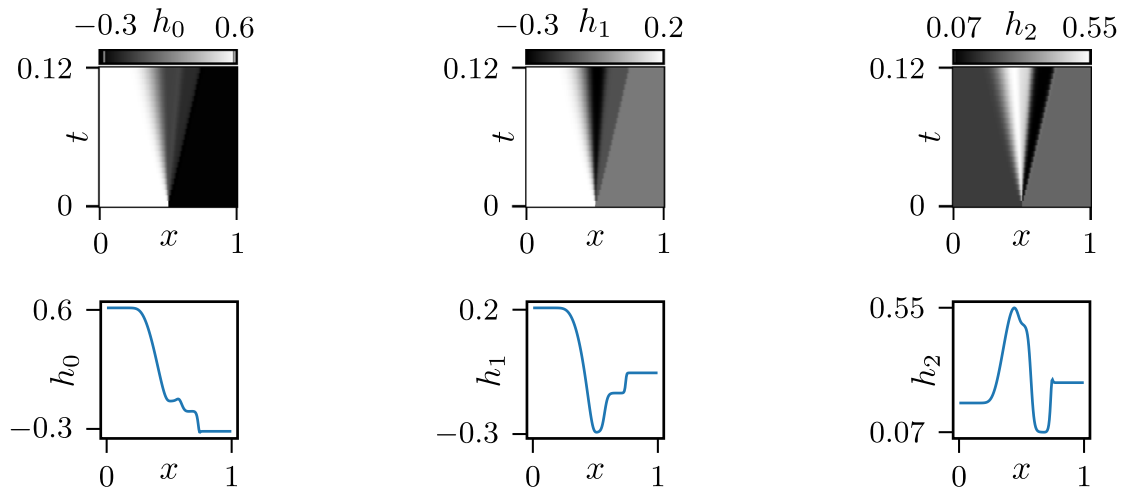

Fig. 10 Intrinsic variables $h_0(x, t)$, $h_1(x, t)$ and $h_2(x, t)$ of hydrodynamic case obtained by the FCNN. Top row depicts the whole (x, t) domain, bottom row is for $t = 0.12$

similar results for the POD based on the previous results, but do not present them here for conciseness. The Pearson correlation coefficient $r_{X,Y} = r_{Y,X}$ is a measure of linear correlation between two sets of data, here represented by a macroscopic variable $X \in \{\rho, \rho u, E, T, u\}$ and an intrinsic variable $Y \in \{r_0, r_1, r_2, \}$ for the hydrodynamic case and $Y \in \{r_0, r_1, r_2, r_3, r_4\}$ for the rarefied case. It is commonly defined as

$$r_{X,Y} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum \sqrt{(x_i - \bar{x})} \sqrt{(y_i - \bar{y})}}. \tag{23}$$

Note that $r_{X,Y} \in [-1, 1]$, with $r_{X,Y} = 0$ meaning that there is no correlation between both data sets, $r_{X,Y} = 1$ indicating a perfect correlation, and $r_{X,Y} = -1$ indicating a perfect anti-correlation.

The Pearson correlation coefficients for the hydrodynamic test case are presented in Fig. 15. As predicted by the previous analysis, there appears to be an almost perfect correlation of the first intrinsic variable r_0 with the density ρ. This means that the FCNN succeeds at identifying the density ρ precisely as an internal variable. Note that r_0 also correlates almost perfectly with the energy E, as the energy depends linearly on ρ. Additionally, there is a relatively strong linear correlation between r_2 and both the momentum
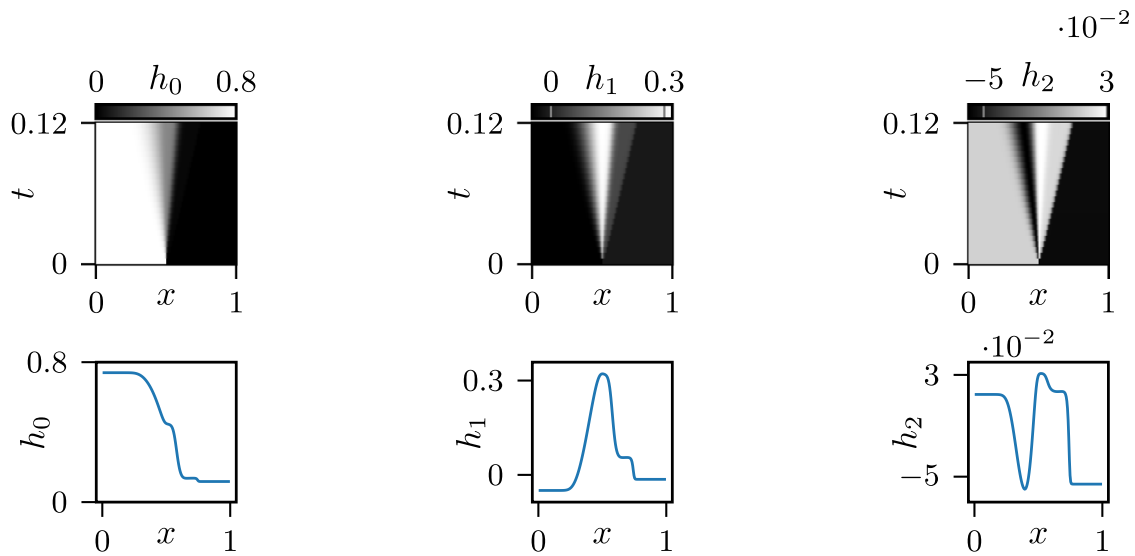
Fig. 11 First three intrinsic variables $h_0(x,t)$, $h_1(x,t)$ and $h_2(x,t)$ of hydrodynamic case obtained by the POD. Top row depicts the whole (x,t) domain, bottom row is for $t = 0.12$
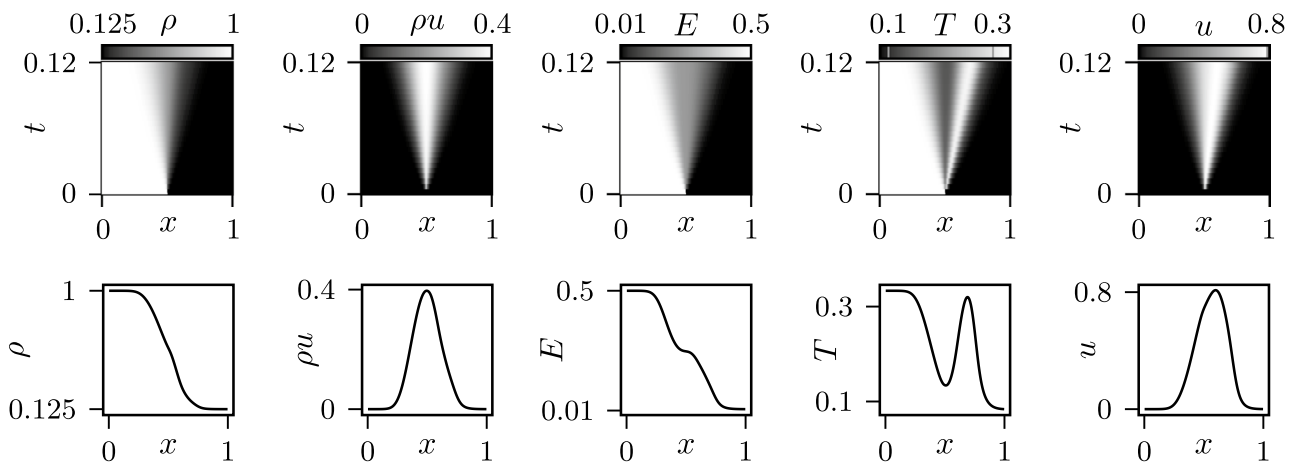

Fig. 12 Macroscopic quantities of the rarefied case obtained by the FOM. Density ρ, momentum ρu, total energy E, temperature T, and velocity u over time t and space x in the top row and at $t = 0.12$ in bottom row

ρu as well as the temperature T. The intrinsic variable r_2 on the other hand seems to be correlated with all variables.

The Pearson correlation coefficients for the rarefied test case are presented in Fig. 16. In agreement with the previous results, there is no clear correlation of most of the intrinsic variables. An exception is r_0, which is anti-correlated with u, and r_3, which correlates almost perfectly with the energy E. Note that only linear correlations are tested here. For further analysis of the disentanglement of the intrinsic variables, it might be more suitable to consider nonlinear correlations beyond the Pearson correlation coefficients. One option that could be explored in future work would be to train the functional relation between the intrinsic variables and

macroscopic variables, for example by means of symbolic regression (Cranmer 2023).

We do expect the correlations to change for different flow cases provided those are characterized by different physical phenomena. Changing the architecture of the neural network is not expected to change the correlation of the intrinsic and physical variables provided that the expressivity of the neural network is sufficient and it is well-trained. In this sense, the amount of data for the training might need to be adjusted, e.g., for larger neural network architectures.
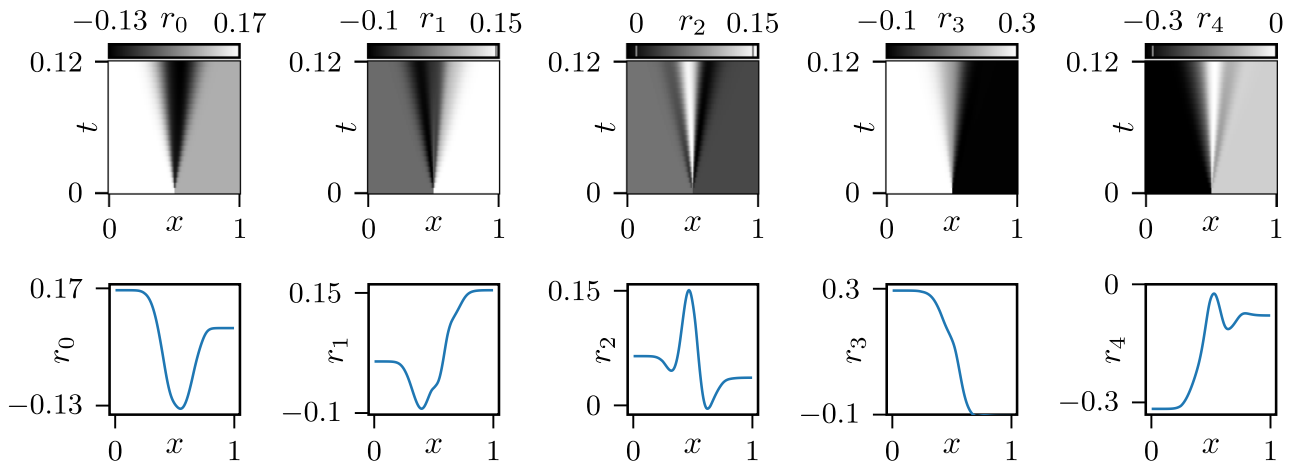
Fig. 13 Intrinsic variables $r_0(x, t)$, $r_1(x, t)$, $r_2(x, t)$, $r_3(x, t)$ and $r_4(x, t)$ of rarefied case obtained by the FCNN. Top row depicts the whole (x, t) domain, bottom row is for $t = 0.12$
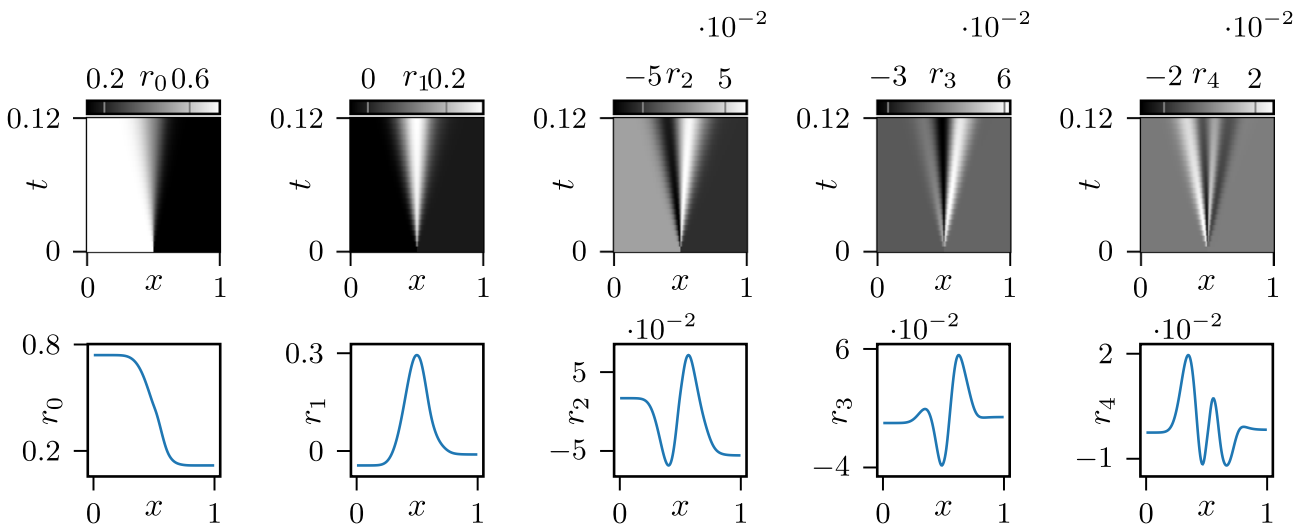
Fig. 14 First five intrinsic variables $r_0(x, t)$, $r_1(x, t)$, $r_2(x, t)$, $r_3(x, t)$ and $r_4(x, t)$ of rarefied case obtained by the POD. Top row depicts the whole (x, t) domain, bottom row is for $t = 0.12$

5 Conclusion

This paper marks the first comparison of velocity space model reduction techniques for rarefied flows using proper orthogonal decomposition (POD) and a fully connected neural network (FCNN).

As physically expected, the rarefied regime needs more modes than the hydrodynamic regime. Choosing three and five intrinsic variables for the hydrodynamic and rarefied case, respectively, leads to less than one percent error. The FCNN is initially more accurate but has a remaining error even for a larger number of intrinsic variables, while POD achieves subsequently higher accuracy with increasing the number of parameters. The resulting errors of the macroscopic variables are small, especially in the smoother rarefied case.

Even though not strictly enforced, the FCNN approximately exhibits the correct conservation of mass, momentum, and energy, while POD has a slightly larger error.

The correlation of intrinsic variables and macroscopic variables is investigated by means of the evolution of the reconstructed values and the pairwise correlation for the FCNN. The density is directly included in the latent space while the relation with other macroscopic variables seems to be more complex.

In addition to optimizing the neural network's performance, our research endeavors to enhance its predictive
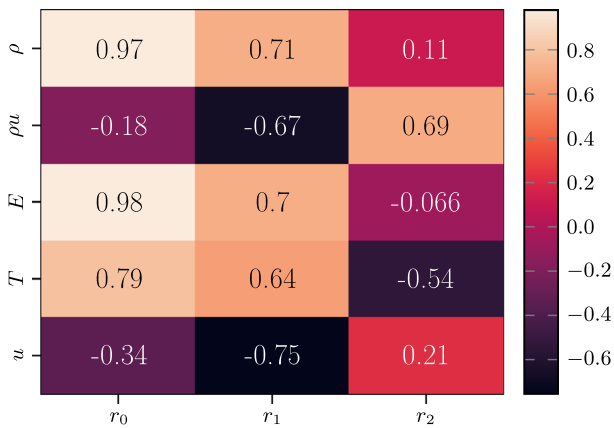
Fig. 15 Pearson correlation between macroscopic quantities and intrinsic variables for the hydrodynamic case
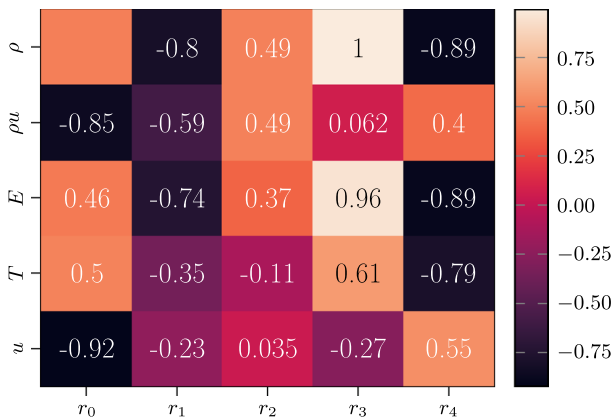

Fig. 16 Pearson correlation between macroscopic quantities and intrinsic variables for the rarefied case

capabilities by incorporating fundamental physical properties such as conservation and interpretability. This objective includes the exploration of different loss functions and regularization techniques. Furthermore, future work can utilize the model reduction approach in real-world scenarios, including simulations and parameter predictions. Lastly, the presented concepts could be applied in related fields, such as shallow flows (Kowalski and Torrilhon 2019) or fusion plasmas (Krah et al. 2023).

Appendix: Hyperparameters for the fully connected autoencoder

In this section, we describe the tests that have been conducted to optimize our hyperparameters. The hyperparameters include the number of layers, i.e., depth; the number of nodes per hidden layer, i.e., width, batch size, and non-linear activation functions; the number of epochs for

Table 4 The initial selection of batch size, bottleneck size, number of epochs, learning rate, and applied activation functions

Mini-batch size	Intr. dim	Epochs	Learn. rate	Activ. hidden/code
16	3/5	2000	0.0001	LeakyReLU/Tanh

training; and the learning rate. Experiments are evaluated through: the validation error which estimates the model's ability to generalize; the training error which estimates the optimization of training data; and the $L2$-error, which gives an estimate of how well the model performs on the whole dataset and hence is applied as a comparative metric against POD.

To start with a working model, an estimate over the initial hyperparameters is done, which are summarized in Table 4. These include a mini-batch size of 16, the width of the bottleneck layer is 3 and 5 for the hydrodynamic case **H** and the rarefied case **R**, respectively, and a learning rate of 0.0001. LeakyReLU is applied as an activation function for the output, input, and any hidden layer besides the output of the bottleneck layer, and is referred to as activation hidden. The hyperbolic tangent is applied as an activation function for the output of the last hidden layer in the encoder which outputs the code, referred to as the activation code. Moreover, 2000 initial number of epochs are used. This might appear exaggerated but is justified by the small amount of input data and the small size of the network which yields fast training.

Five designs for finding an optimal number of layers, i.e., depth, are explored. These are as follows:

1. 10 layers with layer widths: 40, 40, 20, 10, 5, 3/5, 5, 10, 20, 40, 40.
2. 8 layers with layer widths: 40, 40, 20, 10, 3/5, 10, 20, 40, 40.
3. 6 layers with layer widths: 40, 40, 20, 3/5, 20, 40, 40.
4. 4 layers with layer widths: 40, 40, 3/5, 40, 40.
5. 2 layers with layer widths: 40, 3/5, 40.

The model's depth is determined in a primary step because it determines the model's representational capacity and therefore can initiate over- and underfitting at an early stage in the hyperparameter search. The results of the experimentation are shown in Fig. 17 and Table 5 for both rarefaction levels.

For the hydrodynamic case **H**, the lowest validation error of 7.74×10^{-8} and an $L2$ error of 0.0031 is reached with 4 layers and constitutes the best-performing design. Additionally, as seen in Fig. 17(left), a design that exceeds 4 layers results in slight overfitting from the 500th epoch. Less than 4 layers do not reach the validation error and $L2$ error of the other designs, yielding the conclusion, that the capacity

Table 5 Results for the variation of the depth

Depth	Minimum training error		Minimum validation error		L2 error	
	H	**R**	**H**	**R**	**H**	**R**
10	1.53×10^{-7}	5.96×10^{-7}	2.22×10^{-7}	5.19×10^{-7}	0.0048	0.0091
8	1.17×10^{-7}	2.05×10^{-7}	1.58×10^{-7}	2.32×10^{-7}	0.0041	0.0054
6	9.76×10^{-8}	1.40×10^{-7}	1.49×10^{-7}	1.72×10^{-7}	0.0038	0.0045
4	6.29×10^{-8}	1.52×10^{-7}	7.74×10^{-8}	1.61×10^{-7}	0.0031	0.0048
2	1.29×10^{-6}	3.29×10^{-6}	1.37×10^{-6}	3.42×10^{-6}	0.0136	0.0217

Given are minimum values of training and validation error as well as the L2 error. The minima were reached around the last 50 epochs of the training. The L2 error is evaluated with the model at the last epoch

is too low. Overfitting occurs with 4 layers only after the 1000th epoch and is of smaller magnitude compared to the other three models that show overfitting.

For rarefied case **R**, the lowest validation error of 1.61×10^{-7} is also reached with 4 layers. On the other hand, the lowest L2 error of 0.0031 and the lowest training error of 1.40×10^{-7} are reached with 6 layers. Contrary to the previously discussed hydrodynamic case, the training error and L2 errors are of lower magnitude for 6 layers, except for the validation error. Looking at Fig. 17(right), it is observed that the model with 6 layers starts to overfit after the 1500 epochs, yielding a decreasing training error and a stagnating validation error. Hence the model improved in the optimization task which additionally improves the L2 error. Its generalization ability, measured by the validation error, did not improve and is larger than the validation error reached with 4 layers. This concludes a model with 4 layers constitutes the best-performing.

Qualitatively, the overall training for both rarefaction levels is very stable. Training and validation errors do not diverge excessively and converge early in training. Separation of training and validation error occurs prominently for the hydrodynamic solution.

The width of the two remaining hidden layers is examined in the following. For both the hydrodynamic and the rarefied regime five experiments are conducted, lowering the hidden units of the hidden layers from fifty to ten. Note that the decoder is chosen to be structurally a reflection of the encoder. Therefore only one parameter

is changed. Results for the hydrodynamic case **H** and the rarefied case **R** are shown in Table 6. Note that the contribution of over-and underfitting is negligible and therefore the training error is omitted. A model with 30 hidden units in encoder and decoder performs best with the hydrodynamic case **H** and reaches a validation error of 1.77×10^{-08}. The corresponding L2 error is equal to 1.5×10^{-3} with a shrinkage factor of 0.015. Overall, the loss of each experiment with the hydrodynamic case **H** is quite similar and ranges from 1.77×10^{-8} to 5.11×10^{-8}. The L2 error behaves similarly and is even equal for 50 and 30 layers. A model with 40 hidden units performs best for the rarefied case **R**. The corresponding validation error is 1.65×10^{-8} with $L2 = 1.4 \times 10^{-3}$, which is smaller than for the hydrodynamic case **H**. The shrinkage factor here is 0.125. In all experiments, a model with 10 hidden nodes performs worst. Training and validation errors over 4000 epochs for both experiments can be seen in Fig. 18.

Next, the mini-batch size is analyzed. Results are displayed in Table 7. Experiments are conducted with mini-batch sizes of 2, 4, 8, 16, 32. The smallest batch size of 2 yields the lowest validation error of 1.15×10^{-8} with corresponding $L2 = 0.0012$ at epoch 4956 for the hydrodynamic case **H**. The lowest validation error with 6.30×10^{-9} is achieved for the rarefied case **R** at epoch 4534 with a batch size of 4. The corresponding L2 error equals 0.001. Small batch sizes have a regularizing effect on the training and therefore are beneficial to generalization. At the same time, the lower the batch size is, the more unstable is the

Table 6 Results for the variation of the width

Hidden units	Validation error		L2		Shrinkage factor	
	H	**R**	**H**	**R**	**H**	**R**
50	1.91×10^{-8}	5.05×10^{-8}	0.0015	0.0025	0.06	0.01
40	2.65×10^{-08}	1.65×10^{-8}	0.0018	0.0014	0.075	0.125
30	1.77×10^{-08}	3.40×10^{-8}	0.0015	0.0021	0.015	0.0167
20	2.50×10^{-08}	5.25×10^{-8}	0.0017	0.0027	0.1	0.25
10	5.11×10^{-08}	3.97×10^{-7}	0.0025	0.0077	0.3	0.5

Given are the minimum value of the validation error as well as the L2 error. The minima were reached around the last 50 epochs of the training, the L2 error is evaluated with the model at the last epoch

Table 7 Results for the variation of the batch sizes

Batch size	Validation error		L2		Epoch	
	H	**R**	**H**	**R**	**H**	**R**
32	5.40×10^{-8}	2.17×10^{-8}	0.0024	0.0017	4998	4992
16	1.95×10^{-8}	2.06×10^{-8}	0.0015	0.0016	4999	5000
8	2.25×10^{-8}	1.03×10^{-8}	0.0017	0.0012	4965	4961
4	1.52×10^{-8}	6.30×10^{-9}	0.0013	0.0010	3956	4534
2	1.15×10^{-8}	9.18×10^{-9}	0.0012	0.0013	4956	4872

Given are the minimum value of validation error as well as the corresponding epoch. Additionally, the L2 error is given but evaluated with the model at the last epoch

Table 8 Results for the variation of the activation functions for the hidden-/code layers

Activ. hidden/code	Validation error		L2		Epoch	
	H	**R**	**H**	**R**	**H**	**R**
ReLU/ReLU	9.79×10^{-9}	7.18×10^{-9}	0.0010	0.0009	5000	4998
ELU/ELU	4.44×10^{-9}	1.11×10^{-8}	0.0008	0.0012	5000	5000
Tanh/Tanh	7.83×10^{-9}	2.58×10^{-8}	0.0011	0.0018	5000	5000
SiLU/SiLU	7.69×10^{-9}	1.37×10^{-8}	0.0011	0.0013	5000	5000
LeakyReLU/LeakyReLU	1.86×10^{-8}	9.39×10^{-9}	0.0015	0.0010	5000	4997
ELU/Tanh	5.49×10^{-9}	1.87×10^{-8}	0.0008	0.0014	5000	5000
LeakyReLU/Tanh	1.00×10^{-8}	1.42×10^{-8}	0.0010	0.0012	4997	4992
ELU/SiLU	8.11×10^{-9}	1.93×10^{-8}	0.0011	0.0015	5000	5000

Given are the minimum value of validation error as well as the corresponding epoch and the L2 error. The L2 error is evaluated with the models saved when the minimum validation error was achieved during training

training as seen in Fig. 19. The oscillations that begin with batch sizes of 8 and lower, which make the training unstable, can be cured with a lower learning rate as soon as training starts to tremble. Additionally, small batch sizes drastically increase training time, thus a batch size as low as 2 is not used for the experiments. In conclusion, a batch size of 4 is chosen. Furthermore, a reduction of the learning rate from 1×10^{-4} to 1×10^{-5} is applied after the 3000th epoch.

Eight experiments with different activation functions, ReLU, ELU, Tanh, SiLU, and LeakyReLU, are performed. The experiment designs and results are given in Table 8 for hidden and code activations. With the hydrodynamic case **H**, combining ELU and ELU for hidden and code activation, respectively, yields the best result for the

validation error with 4.44×10^{-9} and a corresponding L2 error of 0.0008. These values are achieved at the last epoch. For the rarefied case **R**, a combination of ReLU and ReLU for hidden and code activation, respectively, produces a validation error of 7.18×10^{-9} and a corresponding L2 error of 0.0009. Both are reached close to the last epoch. Note that all models reach their lowest loss at or very close to the last epoch. The reason is the stable training after the 3000th epoch, where the learning rate is lowered to 1×10^{-5} as seen in Fig. 20. This measure shows in all experiments an immediate success for learning. Both validation and training errors fall at the 3001st epoch and only decrease slightly thereafter. This behavior clearly shows that the updates to the free parameters θ were too
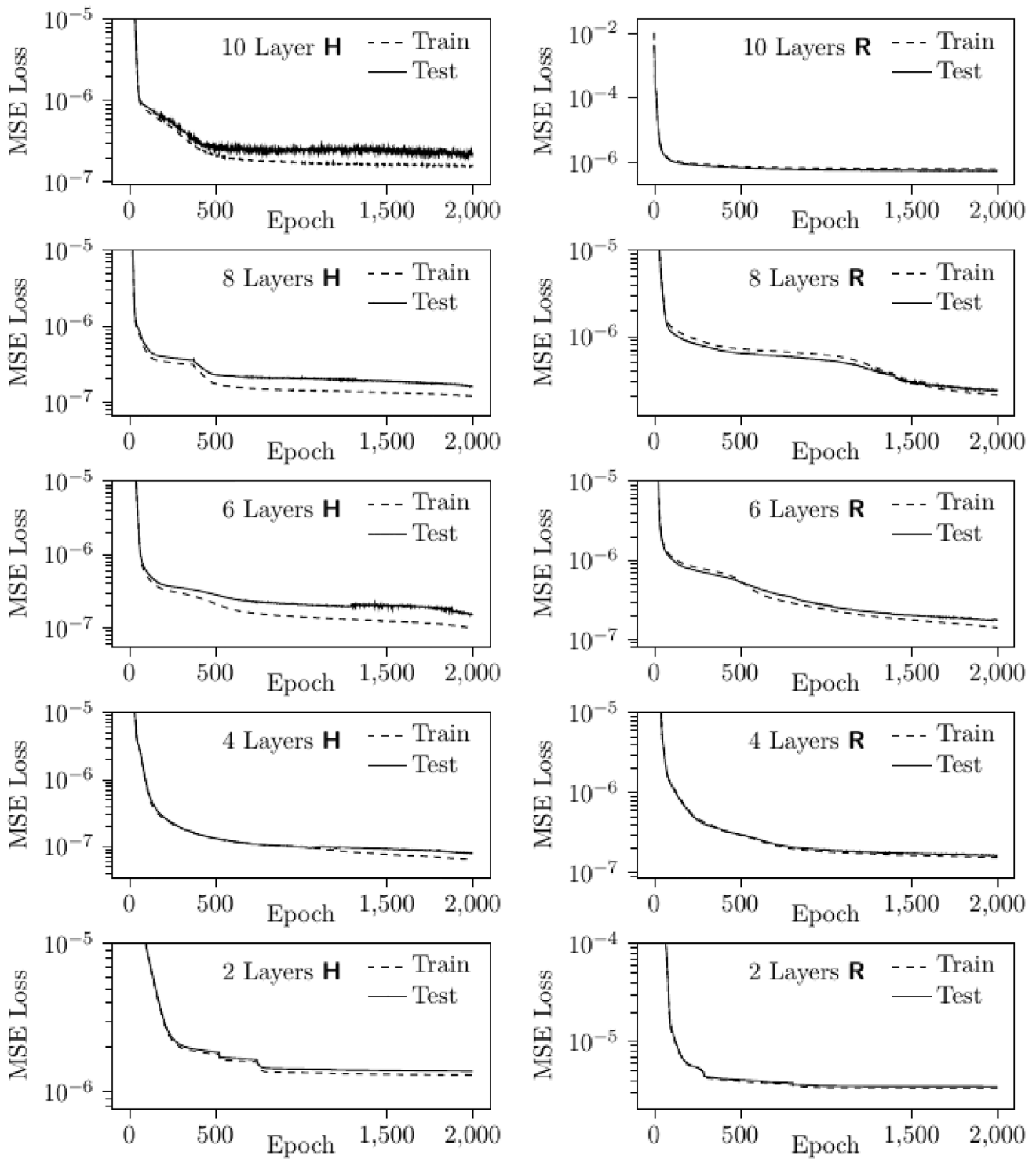
Fig. 17 Five experiments over the different depths with the hydrodynamic case **H** left and the rarefied case **R** right. The number of layers used for every experiment is given. Training and validation loss are shown over 2000 epochs
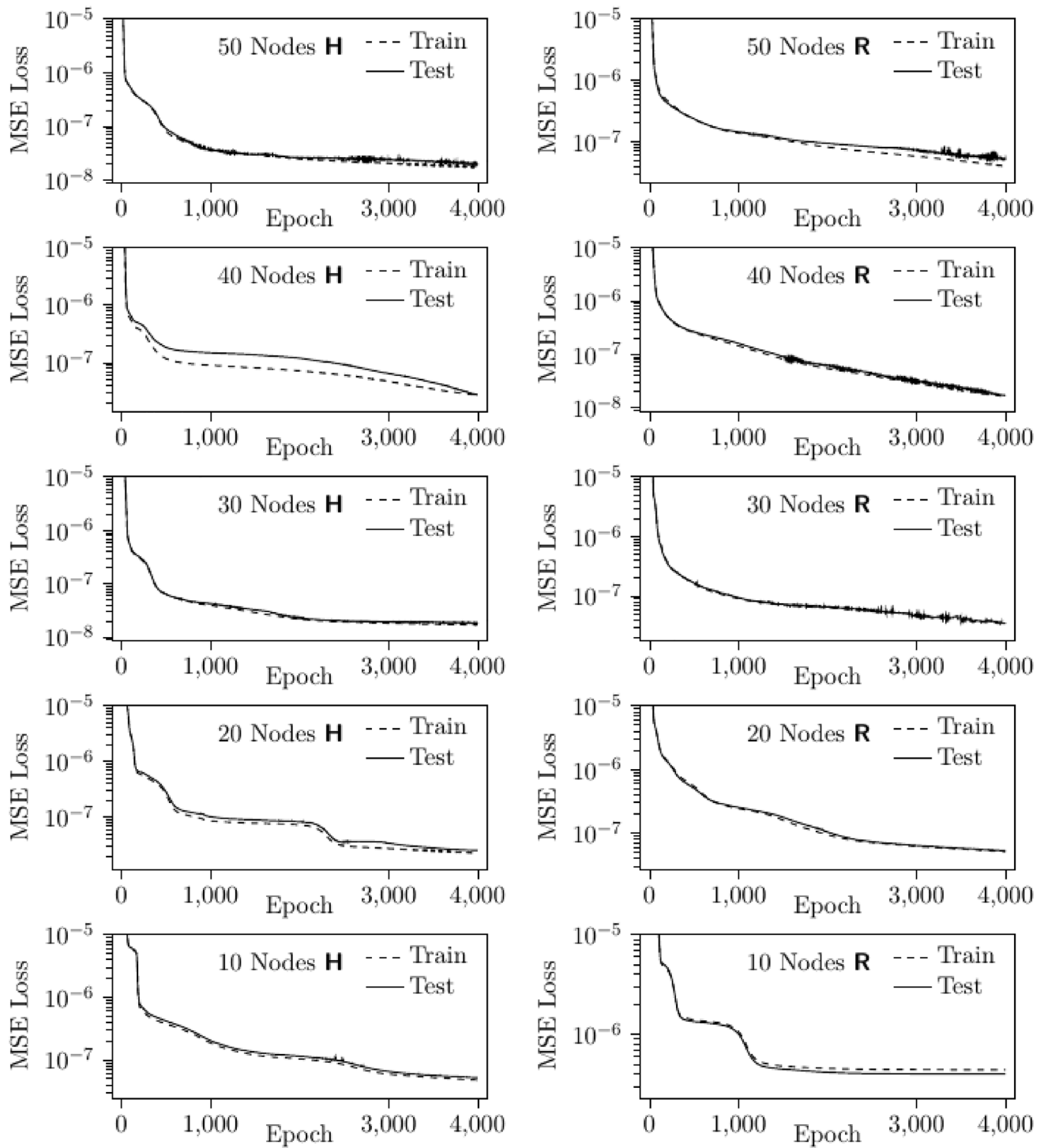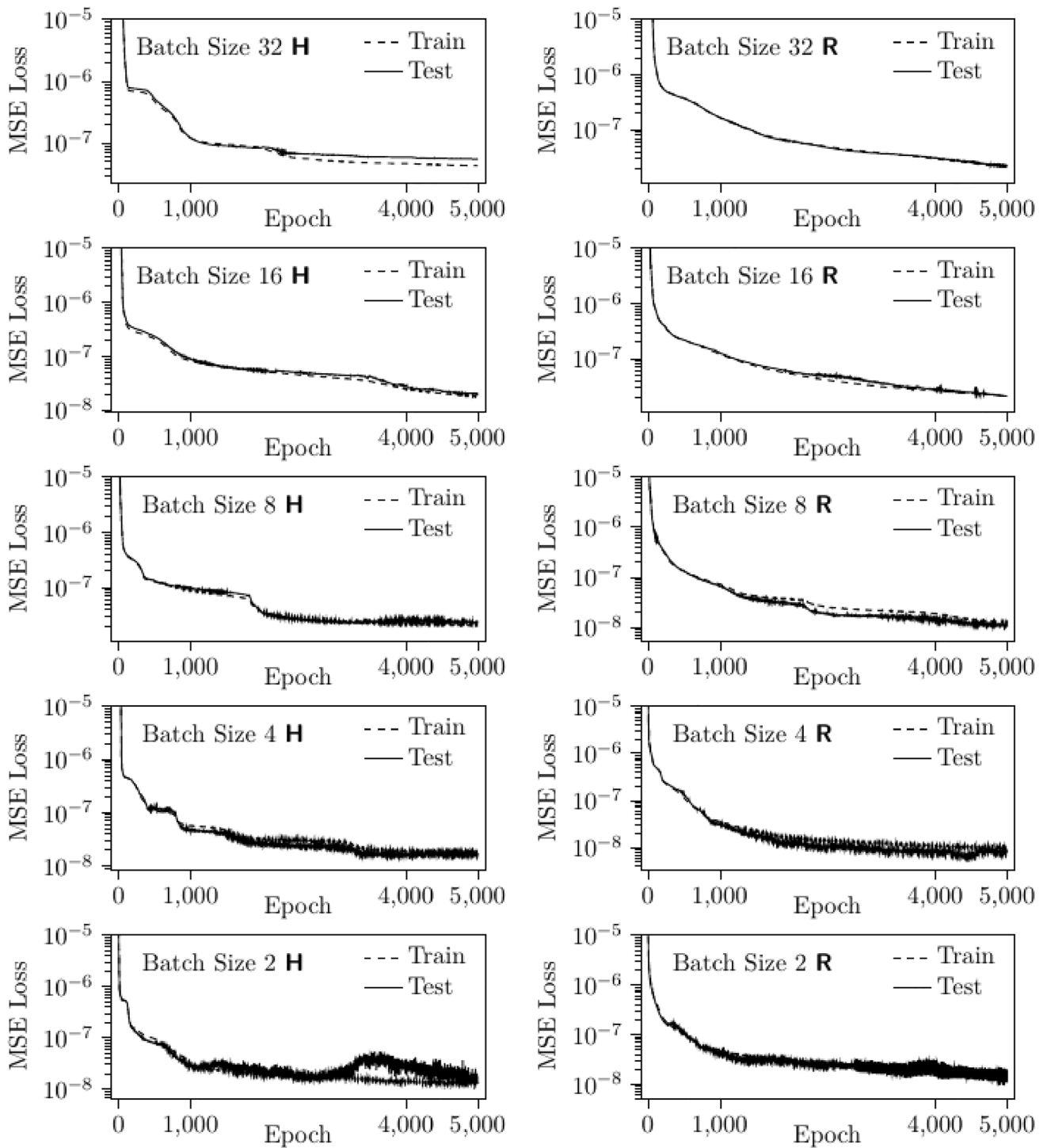
Fig. 18 Five experiments over different widths with the hydrodynamic case **H** left and the rarefied case **R** right. The number of nodes used for every experiment is given. Training and validation loss are shown over 4000 epochs

Fig. 19 Five experiments over different batch sizes with the hydrodynamic case **H** left and the rarefied case **R** right. The batch size used for every experiment is given. Training and validation loss are shown over 5000 epochs
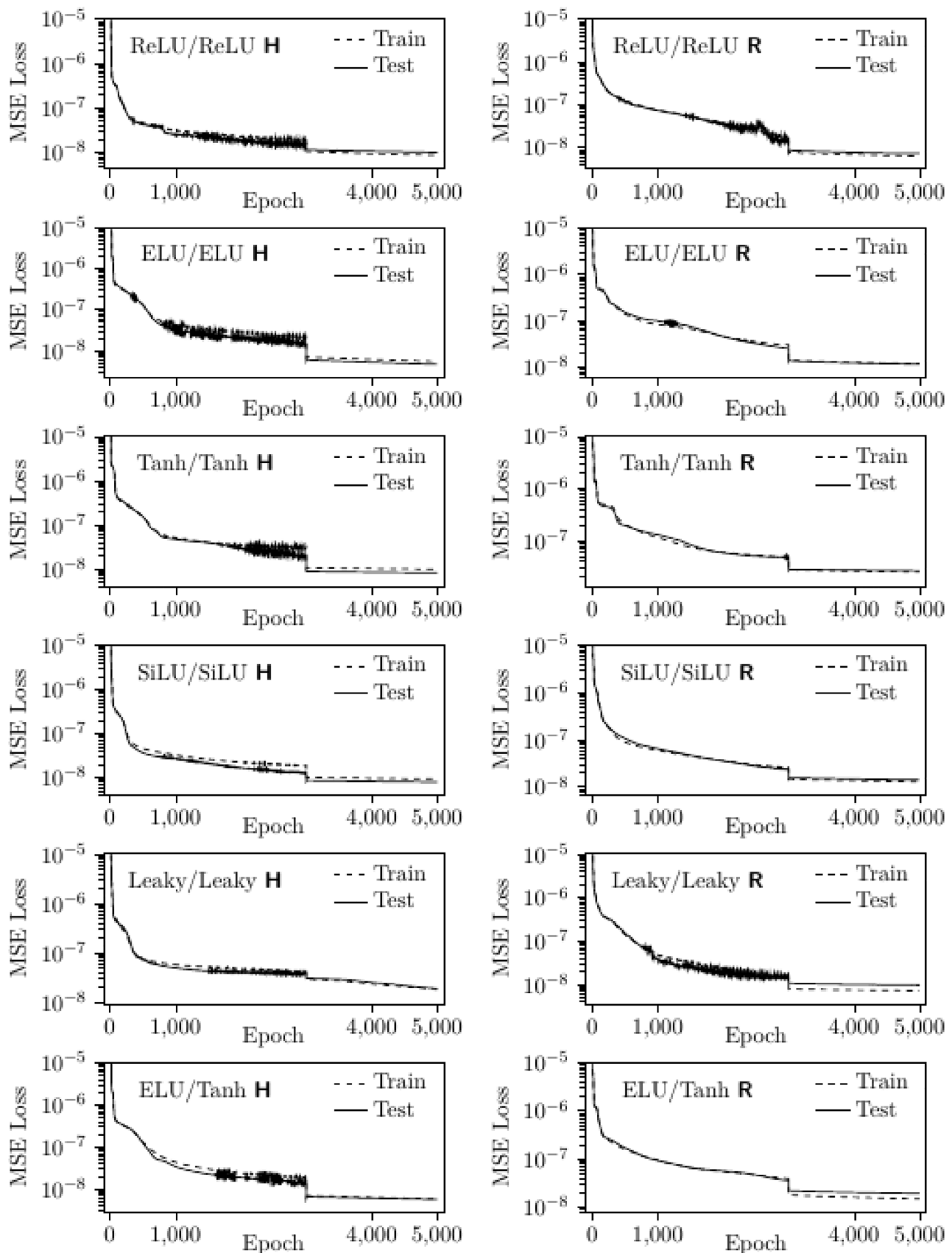
◀**Fig. 20** Six experiments with different combinations of activation functions for the hydrodynamic case **H** left and the rarefied case **R** right. Shown are training and validation error over 5000 epochs

big, which prohibitively slowed down or even prevented the learning process. Small updates to θ made all models quickly reach a minimum.

The final hyperparameters for both input data are summarized below in Table 2. From the initial models to the final models, the decrease in the validation error gained $\approx 1.5 \times 10^{-7}$ for hydrodynamic case **H** and $\approx 7.2 \times 10^{-8}$ for rarefied case **R** which amount to 93% of the initial values for both models.

Author contributions All authors contributed to this publication. The author's contributions differ in the following points: JK: initial idea, methodology, setup of test cases, reference data, writing. PK: initial idea, methodology, implementation, visualization, writing. JR: initial idea, supervision. ZS: implementation, training, visualization.

Code and data availability All data and scripts to reproduce the results can be obtained from the authors on reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Bernard F, Iollo A, Riffaud S (2018) Reduced-order model for the BGK equation based on POD and optimal transport. J Comput Phys 373:545–570

Bhatnagar PL, Gross EP, Krook M (1954) A model for collision processes in gases. 1. Small amplitude processes in charged and neutral one-component systems. Phys Rev 94:511–525

Brull S, Prigent C (2020) Local discrete velocity grids for multi-species rarefied flow simulations. Commun Comput Phys 28(4):1274–1304

Brunton SL, Kutz JN (2019) Data driven science and engineering. Cambridge University Press

Cranmer M (2023) Interpretable machine learning for science with PySR and SymbolicRegression.jl. arXiv preprint arXiv:2305.01582

Debrabant K, Samaey G, Zieliński P (2017) A micro-macro acceleration method for the Monte Carlo simulation of stochastic differential equations. SIAM J Numer Anal 55(6):2745–2786

Degond P, Dimarco G, Pareschi L (2011) The moment-guided Monte Carlo method. Int J Numer Methods Fluids 67(2):189–213

Eckart C, Young G (1936) The approximation of one matrix by another of lower rank. Psychometrika 1(3):211–218

Einkemmer L (2019) A low-rank algorithm for weakly compressible flow. SIAM J Sci Comput 41(5):A2795–A2814

Einkemmer L, Hu J, Ying L (2021a) An efficient dynamical low-rank algorithm for the Boltzmann-BGK equation close to the compressible viscous flow regime. SIAM J Sci Comput 43(5):B1057–B1080

Einkemmer L, Hu J, Wang Y (2021b) An asymptotic-preserving dynamical low-rank method for the multi-scale multi-dimensional linear transport equation. J Comput Phys 439:110353

Fan Y, Koellermeier J, Li J, Li R, Torrilhon M (2016) Model reduction of kinetic equations by operator projection. J Stat Phys 162(2):457–486

Fan F, Xiong J, Li M, Wang G (2020) On interpretability of artificial neural networks: a survey. IEEE Trans Radiat Plasma Med Sci 5:741–760

Garcia AL, Bell JB, Crutchfield WY, Alder BJ (1999) Adaptive mesh and algorithm refinement using direct simulation Monte Carlo. J Comput Phys 154(1):134–155

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press

Han Z, Zhao J, Leung H, Ma KF, Wang W (2019) A review of deep learning models for time series prediction. IEEE Sens J 21(6):7833–7848

Koch O, Lubich C (2007) Dynamical low-rank approximation. SIAM J Matrix Anal Appl 29(2):434–454

Koellermeier J, Torrilhon M (2017) Numerical study of partially conservative moment equations in kinetic theory. Commun Comput Phys 21(4):981–1011

Koellermeier J, Torrilhon M (2018) Two-dimensional simulation of rarefied gas flows using quadrature-based moment equations. Multiscale Model Simul 16(2):1059–1084

Koellermeier J, Krah P, Kusch J (2023) Macro-micro decomposition for consistent and conservative model order reduction of hyperbolic shallow water moment equations: a study using POD-Galerkin and dynamical low Rank approximation. arXiv preprint arXiv:2302.01391

Kowalski J, Torrilhon M (2019) Moment approximations and model cascades for shallow flow. Commun Comput Phys 25(3):669–702

Krah P, Yin X-Y, Bergmann J, Nave J-C, Schneider K (2023) A characteristic mapping method for Vlasov-Poisson with extreme resolution properties. arXiv preprint arXiv:2311.09379

Kunisch K, Volkwein S (1999) Control of the burgers equation by a reduced-order approach using proper orthogonal decomposition. J Optim Theory Appl 102:345–371

Lee K, Carlberg KT (2020) Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. J Comput Phys 404:108973

LeVeque RJ (2002) Finite volume methods for hyperbolic problems. Cambridge texts in applied mathematics. Cambridge University Press

Maes V, Dekeyser W, Koellermeier J, Baelmans M, Samaey G (2023) Hilbert expansion based fluid models for kinetic equations

describing neutral particles in the plasma edge of a fusion device. Phys Plasmas 30(6):063907

McClarren RG, Hauck CD (2010) Robust and accurate filtered spherical harmonics expansions for radiative transfer. J Comput Phys 229(16):5597–5614

Mieussens L (2000) Discrete velocity model and implicit scheme for the BGK equation of rarefied gas dynamics. Math Models Methods Appl Sci 10(08):1121–1149

Mieussens L, Baranger C, Claudel J, Herouard N (2012) Locally refined discrete velocity grids for deterministic rarefied flow simulations. AIP Conf Proc 1501(1):389

Minaee S, Boykov Y, Porikli F, Plaza A, Kehtarnavaz N, Terzopoulos D (2021) Image segmentation using deep learning: a survey. IEEE Trans Pattern Anal Mach Intell 44(7):3523–3542

Mirsky L (1960) Symmetric gauge functions and unitarily invariant norms. Q J Math 11(1):50–59

Pinkus A (1999) Approximation theory of the MLP model in neural networks. Acta Numer 8:143–195

Sirovich L (1987) Turbulence and the dynamics of coherent structures. Part I-III. Q Appl Math 45(3):561–571

Sod GA (1978) Review. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. J Comput Phys 27(1):1–31

Struchtrup H, Torrilhon M (2008) Higher-order effects in rarefied channel flows. Phys Rev E 78(4):46301

Tian C, Fei L, Zheng W, Xu Y, Zuo W, Lin C-W (2020) Deep learning on image denoising: an overview. Neural Netw 131:251–275

Torrilhon M (2015) Convergence study of moment approximations for boundary value problems of the Boltzmann-BGK equation. Commun Comput Phys 18(3):529–557

Torrilhon M (2016) Modeling nonequilibrium gas flow based on moment equations. Annu Rev Fluid Mech 48(1):429–458

Van Der Maaten L, Postma E, Van den Herik J et al (2009) Dimensionality reduction: a comparative. J Mach Learn Res 10(66–71):13