

# Distribution learning for fast generative flow sampling

By D. A. Bezgin<sup>†</sup>, J. M. Winter<sup>†</sup> AND N. A. Adams<sup>‡</sup>

We explore the use of machine learning (ML) to capture velocity probability distribution functions (PDFs) in the context of the lattice Boltzmann method (LBM) and the closure of the Reynolds-averaged Navier-Stokes (RANS) equations. For the LBM, the physics of the governing equations are modeled through a collision process that describes the relaxation of the instantaneous velocity distribution function toward an equilibrium. We employ ML surrogates to approximate the collision process directly in distribution space, and find that they give accurate and generalizable results if physical constraints such as equivariance and conservation are satisfied. In the context of RANS modeling, exact closure of the Reynolds stress tensor (RST) is possible if the PDF of the fluctuating quantities is known. We find that diffusion models, trained to map a uniform Gaussian distribution into the data PDF via a reversible Markov chain of parameterized conditionals, accurately capture velocity PDFs and derived quantities such as the RST.

---

## 1. Introduction

Probability distribution functions (PDFs) appear naturally in fluid mechanics, for example, as configuration-space distributions in the lattice Boltzmann method (LBM) or as PDFs of fluctuating turbulent quantities in the closure of Reynolds-averaged Navier-Stokes (RANS) equations.

The LBM (Krüger *et al.* 2017) has attracted significant attention due to its versatility and parallel efficiency. It consists of two fundamental steps: collision and streaming. The collision operator encodes the underlying physics into the algorithm. This project focuses on modeling the collision step of the LBM using neural networks. Machine learning (ML)-based approaches have the potential to improve the computational efficiency and provide a general framework for developing collision models to recover different governing equations. This approach aligns with the growing trend of integrating ML techniques into the LBM (Hennigh 2017; Bedrunka *et al.* 2021; Miller *et al.* 2022; Zhao *et al.* 2023; Ataei & Salehipour 2024). Previous studies (Corbetta *et al.* 2023; Ortali *et al.* 2024) developed ML surrogates for the calculation of the equilibrium distribution function, a substep of the entire collision operator. This work introduces a novel methodology in which the entire postcollision state is calculated through an ML surrogate, offering a more versatile and comprehensive solution.

RANS equations are commonly used for the simulation of turbulent flows in engineering applications. As a result of the recent interest in ML, ML-based closure models for the RANS equations have been explored, e.g., by Ling *et al.* (2016) and Cruz *et al.* (2019). Most of these models directly predict the Reynolds stress tensor (RST) or its divergence, namely the Reynolds force vector (RFV). Instead of modeling the RST or RFV, one

<sup>†</sup> Both authors have contributed equally.

<sup>‡</sup> Chair of Aerodynamics and Fluid Mechanics, Technical University of Munich, Germany

can model the PDF of the fluctuating velocities directly. So-called PDF-based RANS closure modeling has been explored by Jenny *et al.* (2001), who discretized the PDF transport by a set of particles that are evolved according to a Langevin equation. Here, we propose to use ML for modeling the PDF of the fluctuating velocities. In particular, we explore generative models that allow for fast sampling of fluctuating fields and direct computation of the RST.

## 2. Machine learning–based collision models for the lattice-Boltzmann method

The LBM represents macroscopic flow behavior through discretized single-particle distribution functions  $f_i(\mathbf{x}, t)$ , which are defined over discrete spatial coordinate  $\mathbf{x}$  and time  $t$ . The redistribution of particle populations at the mesoscopic scale is governed by the lattice Boltzmann equation (Krüger *et al.* 2017; Chen *et al.* 1992)

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = \Omega(f_i(\mathbf{x}, t)), \quad (2.1)$$

where  $f_i(\mathbf{x}, t)$  is the discretized distribution function,  $\mathbf{c}_i$  is the particle velocity, and  $\Delta t$  is the time-step size. The velocity space is discretized into a finite set of discrete particle velocities  $\mathbf{c}_i$  using a velocity set, without affecting the conservation laws at the macroscopic level. The macroscopic fluid density and momentum are

$$\rho(\mathbf{x}, t) = \sum_i f_i(\mathbf{x}, t), \quad \text{and} \quad \rho \mathbf{u}(\mathbf{x}, t) = \sum_i \mathbf{c}_i f_i(\mathbf{x}, t), \quad (2.2)$$

respectively. The first term on the left-hand side in Eq. (2.1) accounts for the streaming operation. The right-hand side term  $\Omega(f_i(\mathbf{x}, t))$  is the collision operator and represents the effect of fluid viscosity at the molecular level through particle collisions. The collision operator using the Bhatnagar-Gross-Krook (BGK) formulation with single relaxation time (Bhatnagar *et al.* 1954) reads

$$\Omega_i^{BGK}(f) = -\frac{f_i(\mathbf{x}, t) - f_i^{eq}(\mathbf{x}, t)}{\tau} \Delta t \quad (2.3)$$

with the equilibrium distribution function  $f_i^{eq}(\mathbf{x}, t)$  and relaxation time  $\tau$ , which determine the relaxation rate toward the equilibrium. The relaxation rate is directly related to the kinematic shear viscosity  $\nu$  through

$$\nu = c_s^2 \left( \tau - \frac{\Delta t}{2} \right), \quad (2.4)$$

where  $c_s = 1/\sqrt{3}$  is the lattice speed of sound. The equilibrium distribution function that recovers the Navier-Stokes equations reads

$$f_i^{eq}(\mathbf{x}, t) = w_i \rho \left( 1 + \frac{\mathbf{u} \cdot \mathbf{c}_i}{c_s^2} + \frac{(\mathbf{u} \cdot \mathbf{c}_i)^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right) \quad (2.5)$$

where  $w_i$  are weights specific to the velocity set. In the scope of this work, we use the D2Q9 velocity set (Krüger *et al.* 2017) sketched in Figure 1(a).

### 2.1. Machine learning model

The goal of this work is to learn ML surrogate models for the postcollision state

$$f_i^*(\mathbf{x}, t) = f_i(\mathbf{x}, t) \left( 1 - \frac{\Delta t}{\tau} \right) + f_i^{eq}(\mathbf{x}, t) \frac{\Delta t}{\tau} \quad (2.6)$$

---

Hidden layer sizes	[40, 40]
Learning rate	$2.5 \times 10^{-4}$
Learning rate scheduler	Reduce learning rate on plateau
Loss function	L1
Epochs	200
Batch size	1000
Activation function	GELU
Optimizer	Adam
Size of the data set	$1 \times 10^{-5}$
Train, val, test split ratio	0.8, 0.1, 0.1

---

TABLE 1. Hyperparameters used for training the model.

which follows from inserting Eq. (2.3) into Eq. (2.1). Given a discretized distribution function in the precollision state  $f_i^{pre} = \{f_0^{pre}, \dots, f_q^{pre}\}$ , where  $q$  is the number of discrete velocities of the velocity set and the relaxation time  $\tau$ , our goal is to find a model that predicts the postcollision state  $f_i^* = \{f_0^*, \dots, f_q^*\}$ . The D2Q9 velocity set, for which  $q = 9$  holds, is sketched in Figure 1(a) along with the respective discrete velocities. Following the approach of Corbetta *et al.* (2023); Ortali *et al.* (2024), we employ a multilayer perceptron (MLP) to model the postcollision state.

Corbetta *et al.* (2023) and Ortali *et al.* (2024) highlight that, for achieving accurate and physically consistent results, it is crucial to satisfy the scale equivariance of the collision operator, mass and momentum invariance, and rotation and reflection equivariance according to the eighth-order dihedral symmetry group. The first two properties are enforced using the approach described by Corbetta *et al.* (2023). The third is ensured using the group-averaging procedure outlined by Ortali *et al.* (2024).

The model is trained with a synthetic data set that allows generalization for different test cases. Following the approach of Corbetta *et al.* (2023), we generate pairs of precollision states and corresponding equilibrium states. Subsequently, we sample the relaxation time from a uniform distribution  $\tau \sim \mathcal{U}(0.8, 1.2)$ . The lower and upper values for  $\tau$  are chosen to represent a typical range that ensures stable and accurate simulations. Given the precollision state, equilibrium state, and relaxation time, the postcollision state can be calculated using Eq. (2.6).

The hyperparameters used for training are listed in Table 1, and a sketch of the MLP architecture is shown in Figure 1(b). To assess the accuracy of the trained model, we plot the relative L1 error,  $\epsilon_i = |f_{i,\text{true}} - f_{i,\text{pred}}|/f_{i,\text{true}}$ , for each discrete velocity  $i$  of the D2Q9 velocity set as a box plot in Figure 1(c). The relative L1 error converges to an acceptable range for all discrete velocities. We can identify three distinct error levels. These correspond to the zeroth discrete velocity, discrete velocities 1 to 4, and discrete velocities 5 to 8, respectively. This can be attributed to the structure of the D2Q9 velocity set shown in Figure 1(a). The zeroth discrete velocity is the resting velocity. Discrete velocities 1 to 4 correspond to directions along the major coordinate axes, and discrete velocities 5 to 8 correspond to diagonal directions.

## 2.2. Results

We validate the accuracy of the trained collision model for two test cases, the two-dimensional Taylor-Green vortex and the lid-driven cavity case. (Corbetta *et al.* 2023) use the same cases for validation.

The two-dimensional Taylor-Green vortex is set up using periodic boundary conditions

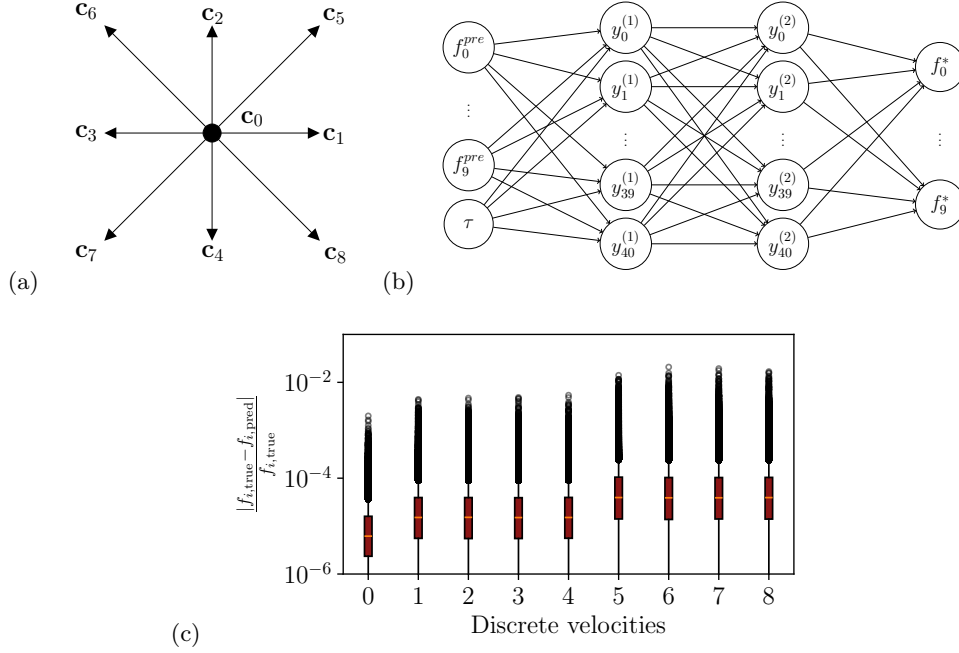


FIGURE 1. (a) Sketch of the D2Q9 velocity with discrete velocity directions  $c_i$ . (b) Sketch of the multilayer perceptron architecture. Inputs are the precollision states  $f_i^{pre}$  and the relaxation time  $\tau$ , and outputs are the postcollision states  $f_i^*$ , where  $i = 1, \dots, 9$  for D2Q9. (c) Box plot evaluating the accuracy of the trained model using the relative L1 error for each discrete velocity separately. The orange line corresponds to the mean error, the red bar indicates its standard deviations, and black dots represent outliers.

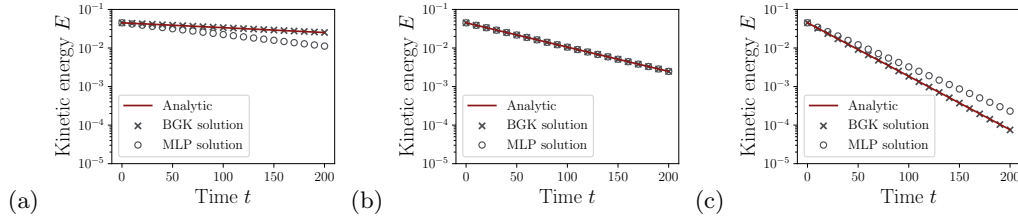


FIGURE 2. Decay in kinetic energy for the two-dimensional Taylor-Green vortex. The relaxation times are (a) 0.6, (b) 1.0, and (c) 1.6.

and the initial velocity field

$$u_x(x, y) = u_0 \cos(x) \sin(y), \quad u_y(x, y) = -u_0 \cos(y) \sin(x), \quad x, y \in [0, 2\pi], \quad (2.7)$$

with  $u_0 = 0.3 \cdot 10^{-2}$ . We simulate the problem using using  $100 \times 100$  cells and consider three different relaxation times:  $\tau = \{0.6, 1.0, 1.6\}$ . Note that, from these relaxation times, only  $\tau = 1.0$  is covered by the data set, whereas  $\tau = 0.6$  and  $1.6$  are not covered. Figure 2 shows the decay of the kinetic energy and compares results obtained using the MLP collision model with reference results obtained using the BGK collision operator and the analytical solution. The reference solution using the BGK collision model perfectly matches the analytical solution for all relaxation times. The MLP solution matches the reference and analytical solutions for  $\tau = 1.0$ , which is covered by the data set. For the

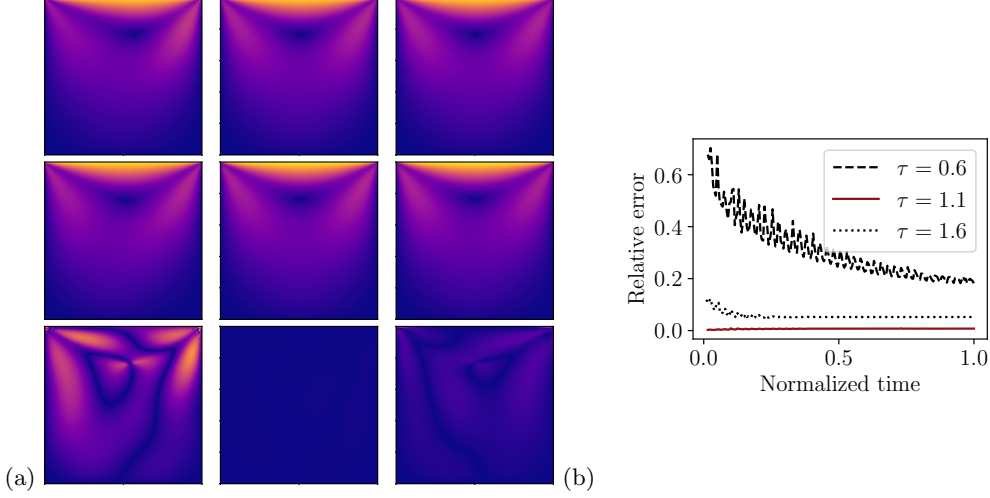


FIGURE 3. (a) Velocity field of the lid-driven cavity test case at the last time step. The color scale is from 0.0 (purple) to 1.0 (yellow). From left to right, the relaxation times are  $\tau = \{0.6, 1.1, 1.6\}$ . The top and middle rows show results using the classical or neural network collision model, respectively. In the bottom row, we show the absolute L1 error of the velocity field. The color scale is from 0.0 (purple) to 0.1 (yellow). (b) Temporal evolution of the mean relative error to the BGK reference solution for the lid-driven cavity test case.

unseen relaxation times  $\tau = \{0.6, 1.6\}$ , the MLP solution deviates from the reference and analytical solutions. However, these simulations also remain stable. As expected, the decay obtained with  $\tau = 0.6$  behaves as if a higher relaxation time was used for the simulation because the data set considers only higher relaxation times. Conversely, the simulation with  $\tau = 1.6$  behaves as if a lower relaxation time was used.

The two-dimensional lid-driven cavity flow case considers a wall-bounded setup, where the top wall moves with a constant velocity  $u_0 = 1.0$  and the remaining walls are at rest. No-slip boundary conditions are applied at all walls. The problem is initialized with zero velocity. The resolution is  $100 \times 100$ . We run the problem for  $\tau = \{0.6, 1.1, 1.6\}$  and compare the results obtained using the BGK collision model and the MLP model in Figure 3. Note that, for the simulated relaxation times only  $\tau = 1.1$  is covered by the data set, whereas  $\tau = \{0.6, 1.6\}$  are unseen. Figure 3(a) demonstrates that the learned collision operator performs well for a relaxation time covered by the data set and shows deficiencies for unseen relaxation times. This is confirmed by Figure 3(b), which depicts the temporal evolution of the mean relative error to the BGK reference solution for all simulated relaxation times.

### 3. Reynolds-averaged Navier-Stokes closure by generative sampling

The incompressible RANS equations are given by

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_j \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial \overline{u'_i u'_j}}{\partial x_j}, \quad \text{and} \quad \frac{\partial \bar{u}_i}{\partial x_i} = 0, \quad (3.1)$$

where  $\bar{u}_i$  is the Reynolds-averaged velocity,  $u'_i = u_i - \bar{u}_i$  is the fluctuating velocity,  $p$  is the averaged pressure, and  $\rho$  is the density. The RANS equations require closure due to the RST  $\overline{u'_i u'_j}$ . A common ansatz for closure of the RANS equations is by an eddy-viscosity

approach in which the RST is assumed to be related to the strain tensor of the averaged velocity field.

One could directly close the RST if one were able to sample from the true distribution of fluctuating velocities,  $\mathbf{u}' \sim q(\mathbf{u}')$ , where  $\mathbf{u}' = [u'_1, u'_2, u'_3]^T$  is the vector of fluctuating velocities. Then, one could obtain the RST by averaging over an ensemble of independently and identically distributed samples. Such a closure model would not rely on any ad hoc assumptions.

Obtaining the true distribution of the fluctuating quantities is far from trivial, and a closed-form expression is generally not available. Generative models, however, are data-driven methods for modeling data distributions of observables and allow sampling from these. Examples of generative models are variational autoencoders, generative adversarial networks, and normalizing flows. A more recent class of generative models consists of so-called diffusion models, which have shown success in generating high-quality samples from distributions of observables. In this work, we explore their potential for modeling the PDF of the fluctuating velocities in turbulent flows.

### 3.1. Diffusion models

Diffusion models (Sohl-Dickstein *et al.* 2015; Ho *et al.* 2020) learn to reverse a diffusion process in order to generate realistic samples from a given data distribution. They consist of a forward process, which iteratively adds Gaussian noise to the real sample, and a parameterized reverse process, which iteratively denoises the sample.

Let  $\mathbf{x}_0 \sim q(\mathbf{x})$  denote a sample from the true distribution  $q$ . The forward process consists of  $T$  steps,  $\mathbf{x}_1, \dots, \mathbf{x}_T$ , where in each step a small amount of Gaussian noise is added to the datum. The conditional distribution for  $q_t$  is therefore given by

$$q(\mathbf{x}_t | \mathbf{x}_{t-1}) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{1 - \beta_t} \mathbf{x}_{t-1}, \beta_t \mathbf{I}\right), \quad (3.2)$$

where  $\mathcal{N}$  denotes the normal distribution, and  $\beta_t \in (0, 1)$ , the so-called variance scheduler, is a hyperparameter of the forward process. The Markov chain of the forward process is

$$q(\mathbf{x}_{1:T} | \mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_{t-1}). \quad (3.3)$$

As  $T \rightarrow \infty$ ,  $\mathbf{x}_T$  approaches an isotropic Gaussian distribution. By reparameterizing Eq. (3.3), the closed-form expression for sampling  $\mathbf{x}_t$  is obtained

$$q(\mathbf{x}_t | \mathbf{x}_0) = \mathcal{N}\left(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}\right). \quad (3.4)$$

Here,  $\alpha_t = 1 - \beta_t$  and  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ .

The reverse process  $q(\mathbf{x}_{t-1} | \mathbf{x}_t)$  is modeled by the conditional probabilities

$$p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t)\right), \quad (3.5)$$

so that

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_t), \quad (3.6)$$

where  $p(\mathbf{x}_T) \sim \mathcal{N}(0, \mathbf{I})$ .

The training objective can be derived from the variational lower bound

$$L_t = E_{t, \mathbf{x}_0, \boldsymbol{\epsilon}_t} \left[ \|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\epsilon}_t, t)\|^2 \right]. \quad (3.7)$$

Note that a reparameterization was used in the derivation of Eq. (3.7), so that the model  $\epsilon_\theta$  predicts the noise  $\epsilon_t$  given the input  $\mathbf{x}_t$ ; see Ho *et al.* (2020) for details.

In the present work, we use Markov chains of length  $T = 1000$  for forward and reverse processes. The variance  $\beta_t$  follows a cosine-based variance schedule. The reverse process is represented by a U-Net architecture (Ronneberger *et al.* 2015) with three feature map resolutions ( $64 \times 64$ ,  $32 \times 32$ , and  $16 \times 16$ ). Each resolution level consists of two convolutional ResNet blocks followed by an attention block (Vaswani *et al.* 2017). As the diffusion model in this present work generates global samples of, for example, the entire xy slice, the attention mechanism facilitates spatially coherent turbulent structures in the generated realizations of the flow field; see also the inclusion of an attention layer in the diffusion model of Ho *et al.* (2020). Group normalization is employed throughout. Parameters are shared across steps in the reverse process. For training, the Adam optimizer is used with a constant learning rate of  $1 \times 10^{-3}$ .

### 3.2. Data set

In this work, we use direct numerical simulation (DNS) data of turbulent channel flow at a nominal Reynolds number  $Re_\tau = 180$  and a nominal bulk Mach number  $Ma_b = 0.2$  for training. Simulations are conducted with the JAX-Fluids solver (Bezgin *et al.* 2023, 2024). JAX-Fluids is a finite-volume solver that discretizes the compressible Navier-Stokes equations on a Cartesian mesh. The spatial discretization consists of a TENO6-A scheme (Fu *et al.* 2019) combined with an HLLC (Toro 2009) approximate Riemann solver. The TVD-RK3 scheme is used for temporal integration. Hyperbolic tangent mesh stretching is used in the wall-normal direction to provide sufficient spatial resolution in near-wall regions. The mesh in the streamwise and spanwise directions is uniform.

The computational domain  $[0, 4\pi h] \times [-h, h] \times [0, 4/3\pi h]$  is resolved by  $128 \times 128 \times 128$  finite-volume cells, which corresponds to  $\Delta x^+ = 17.54$ ,  $\Delta y^+ = [0.85, 4.62]$ , and  $\Delta z^+ = 5.85$ . The actual Reynolds number in the simulation is  $Re_\tau = 178.61$ . Figure 4 (a) compares our results with the reference DNS data from Moser *et al.* (1999). Good agreement is observed. Figure 4 (b) visualizes two samples of the fluctuating streamwise velocity field; that is,  $u_1 = u_1 - \bar{u}_1$ . In total, 21 three-dimensional snapshots of the fluctuating fields were collected, resulting in 2688 xy slices for training. Thus, the training data have shape  $(N_{\text{samples}}, C, H, W) = (2688, 3, 128, 128)$ , where the velocity components are stacked along the channel dimension  $C$ , and  $H$  and  $W$  are the spatial resolution. The training data are min-max normalized per channel dimension (i.e., min-max normalization is performed for each velocity component separately).

### 3.3. Results

The trained diffusion model is evaluated in *a priori* tests. We draw  $N = 100$  samples from the diffusion model. Each sample is a single realization of a two-dimensional xy slice of the turbulent flow field and has the shape  $(3, 128, 128)$ . We compute the RST as well as the PDF of the velocity fluctuations based on the generated ensemble. Figure 5 compares the PDF of the fluctuating streamwise velocity for the generated samples with DNS data at three different wall-normal positions. The generated samples recover the shape of the target PDFs well. Due to the limited number of generated samples, we observe a slight deviation in the tails of the target PDF. Figure 6 compares the generated RST with the RST of the DNS data for  $N = \{1, 10, 100\}$  samples. The quality of the RST generated using the diffusion model increases with the number of samples. Overall, the RST is recovered with reasonable accuracy.

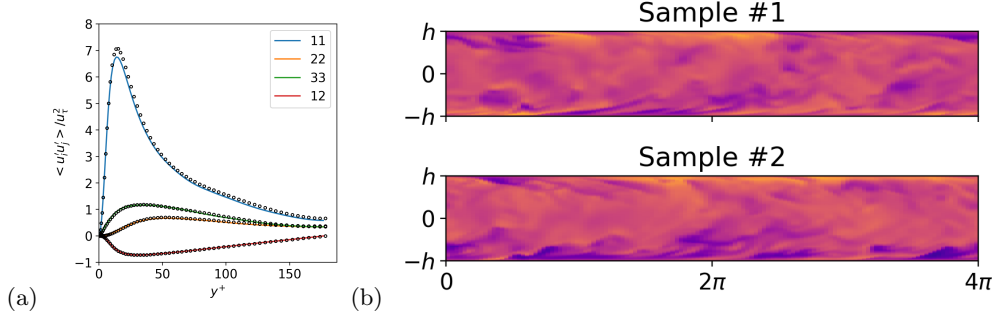


FIGURE 4. (a) Reynolds stress components. Comparison of the data obtained by JAX-Fluids (lines) with reference DNS results by Moser *et al.* (1999) (circles). (b) DNS samples of the fluctuating streamwise velocity component  $u_1' = u_1 - \bar{u}_1$ .

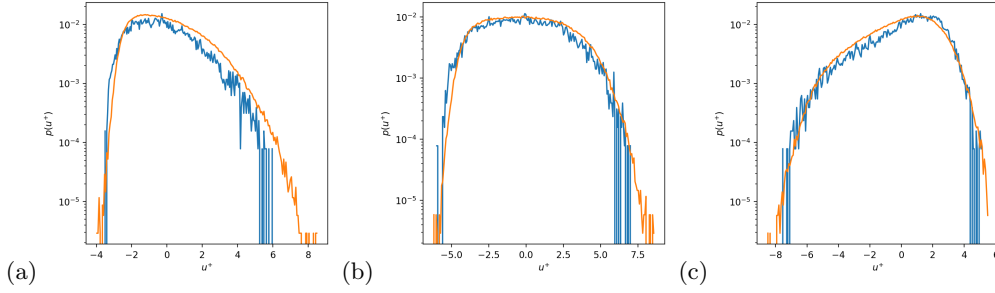


FIGURE 5. Probability density functions (PDFs) of the streamwise fluctuating velocity component at different locations: (a)  $y^+ = 5.20$ , (b)  $y^+ = 9.79$ , and (c)  $y^+ = 30.79$ . Orange line: PDFs based on DNS data. Blue line: PDFs based on 100 generated samples from the diffusion model.

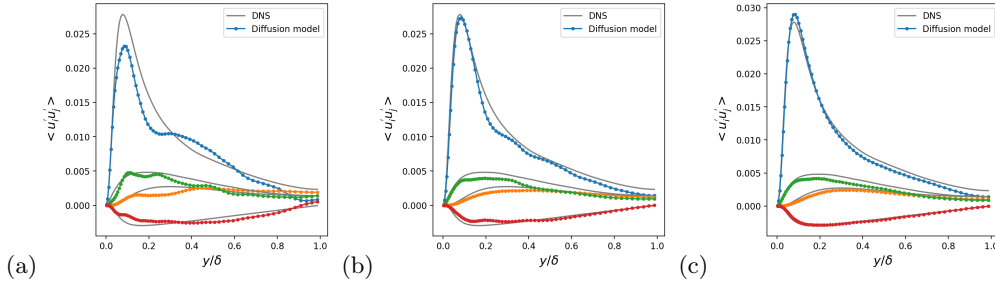


FIGURE 6. Reynolds stress tensor for (a) 1, (b) 10, and (c) 100 samples generated by the diffusion model.

#### 4. Conclusions

In the context of the LBM, this work presents a novel approach for replacing the collision step using an ML surrogate. The trained model provides accurate results when applied to problems with relaxation times covered by the data set. As expected, the accuracy of the model suffers when unseen relaxation times are used. However, the simulation also remains stable for these cases. Future work will focus on a more extensive evaluation of this model, including a detailed analysis of its generalizability with respect to the relaxation time, and extensions to more complicated collision models such as two-relaxation-time or multi-relaxation-time models.

Additionally, this work introduces a novel approach for directly closing the RANS

equations by generative sampling. A diffusion model learns the PDF of the fluctuating velocities. This allows for straightforward computation of the RST using ensemble averaging, without introducing any ad hoc assumptions regarding the closure form. *A priori* tests show that diffusion models are capable of generating realistic samples that recover the true velocity PDF. This work is limited to turbulent channel flow at a single Reynolds number. Future works will expand the training data set and explore conditional diffusion models.

#### Acknowledgments

We gratefully acknowledge fruitful discussions with Aakash Patil, Diego Rossinelli, and Gianluca Iaccarino over the course of the CTR summer program.

#### REFERENCES

- ATAEI, M. & SALEHIPOUR, H. 2024 XLB: A differentiable massively parallel lattice Boltzmann library in Python. *Comp. Phys. Commun.* **300**, 109187.
- BEDRUNKA, M. C., WILDE, D., KLIEMANK, M., REITH, D., FOYSI, H. & KRÄMER, A. 2021 Lettuce: PyTorch-based lattice Boltzmann framework. In *High Performance Computing: ISC High Performance Digital 2021 International Workshops, Frankfurt am Main, Germany, June 24–July 2, 2021, Revised Selected Papers 36*, pp. 40–55. Springer.
- BEZGIN, D. A., BUHENDWA, A. B. & ADAMS, N. A. 2023 JAX-Fluids: A fully-differentiable high-order computational fluid dynamics solver for compressible two-phase flows. *Comp. Phys. Commun.* **282**, 108527.
- BEZGIN, D. A., BUHENDWA, A. B. & ADAMS, N. A. 2024 JAX-Fluids 2.0: Towards HPC for differentiable cfd of compressible two-phase flows. *arxiv:2402.05193* .
- BHATNAGAR, P., GROSS, E. & KROOK, M. 1954 A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems. *Phys. Rev.* **94** (3), 511 – 525.
- CHEN, H., CHEN, S. & MATTHAEUS, W. H. 1992 Recovery of the Navier-Stokes equations using a lattice-gas Boltzmann method. *Phys. Rev. A* **45** (8), R5339–R5342.
- CORBETTA, A., GABBANA, A., GYRYA, V., LIVESCU, D., PRINS, J. & TOSCHI, F. 2023 Toward learning lattice Boltzmann collision operators. *Eur. Phys. J. E* **46**, 10.
- CRUZ, M. A., THOMPSON, R. L., SAMPAIO, L. E. & BACCHI, R. D. 2019 The use of the Reynolds force vector in a physics informed machine learning approach for predictive turbulence modeling. *Comput. Fluids* **192**, 104258.
- FU, L., HU, X. Y. & ADAMS, N. A. 2019 Improved five- and six-point targeted essentially nonoscillatory schemes with adaptive dissipation. *AIAA J.* **57**, 1143–1158.
- HENNIGH, O. 2017 Lat-Net: Compressing lattice Boltzmann flow simulations using deep neural networks. *arXiv:1705.09036 [stat.ML]* .
- HO, J., JAIN, A. & ABBEEL, P. 2020 Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Red Hook, NY: Curran Associates Inc.
- JENNY, P., POPE, S., MURADOGLU, M. & CAUGHEY, D. 2001 A hybrid algorithm for the joint PDF equation of turbulent reactive flows. *J. Comput. Phys.* **166**, 218–252.
- KRÜGER, T., KUSUMAATMAJA, H., KUZMIN, A., SHARDT, O., SILVA, G. & VIGGEN, E. M. 2017 *The Lattice Boltzmann Method*. SpringerLink.
- LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016 Reynolds averaged turbulence mod-

- elling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166.
- MILLER, S. T., ROBERTS, N. V., BOND, S. D. & CYR, E. C. 2022 Neural-network based collision operators for the Boltzmann equation. *J. Comput. Phys.* **470**, 111541.
- MOSER, R. D., KIM, J. & MANSOUR, N. N. 1999 Direct numerical simulation of turbulent channel flow up to  $Re = 590$ . *Phys. Fluids* **11**, 943–945.
- ORTALI, G., GABBANA, A., ATMODIMEDJO, I. & CORBETTA, A. 2024 Enhancing lattice kinetic schemes for fluid dynamics with lattice-equivariant neural networks. *arxiv:2405.13850* .
- RONNEBERGER, O., FISCHER, P. & BROX, T. 2015 U-Net: Convolutional networks for biomedical image segmentation. In *Image Computing and Computer-Assisted Intervention (MICCAI 2015): 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III*, pp. 234–241. Springer.
- SOHL-DICKSTEIN, J., WEISS, E., MAHESWARANATHAN, N. & GANGULI, S. 2015 Deep unsupervised learning using nonequilibrium thermodynamics. *Proc. Mach. Learn. Res.* **37**, 2256–2265.
- TORO, E. F. 2009 Riemann Solvers and Numerical Methods for Fluid Dynamics. *Springer. 3rd ed.* .
- VASWANI, A., SHAZEER, N., PARMAR, N., USZKOREIT, J., JONES, L., GOMEZ, A. N., KAISER, L. & POLOSUKHIN, I. 2017 Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010. Red Hook, NY.
- ZHAO, Y., MENG, F. & LU, X. 2023 Improvement of lattice Boltzmann methods based on gated recurrent unit neural network. *Signal Image Video P.* **17**, 1–9.