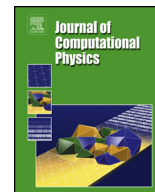




Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

RelaxNet: A structure-preserving neural network to approximate the Boltzmann collision operator

Tianbai Xiao^{a,*}, Martin Frank^b^a State Key Laboratory of High Temperature Gas Dynamics, Institute of Mechanics, Chinese Academy of Sciences, Beijing, China^b Steinbuch Centre for Computing, Karlsruhe Institute of Technology, Karlsruhe, Germany

ARTICLE INFO

Article history:

Received 17 November 2022

Received in revised form 6 May 2023

Accepted 19 June 2023

Available online 24 June 2023

Keywords:

Kinetic theory

Computational fluid dynamics

Scientific machine learning

Artificial neural network

ABSTRACT

The extremely high dimensionality and nonlinearity in the Boltzmann equation bring tremendous difficulties to the study of rarefied gas dynamics. This paper addresses a neural network-based surrogate model that provides a structure-preserving approximation for the fivefold collision integral. The notion originates from the similarity in structure between the BGK-type relaxation model and residual neural network (ResNet) when a particle distribution function is treated as the input to the neural network function. Therefore, we extend the ResNet architecture and construct what we call the relaxation neural network (RelaxNet). Specifically, two feed-forward neural networks with physics-informed connections and activations are introduced as building blocks in RelaxNet, which provide bounded and physically realizable approximations of the equilibrium distribution and velocity-dependent relaxation time respectively. The evaluation of the collision term is thus significantly accelerated due to the fact that the convolution in the fivefold integral is replaced by tensor multiplication in the neural network. We fuse the mechanical advection operator and the RelaxNet-based collision operator into a unified model named the universal Boltzmann equation (UBE). We prove that UBE preserves the key structural properties in a many-particle system, i.e., positivity, conservation, invariance, H-theorem, and correct fluid dynamic limit. These properties promise that RelaxNet is superior to strategies that naively approximate the right-hand side of the Boltzmann equation using a machine learning model. A novel sampling strategy based on closure hierarchies of the moment system of the Boltzmann equation is developed to generate reliable and unbiased sampling for the supervised learning. The construction of the RelaxNet-based UBE and its solution algorithm are demonstrated in detail. Several numerical experiments, where the ground-truth datasets are produced by the Shakhov model, velocity-dependent ν -BGK model, and the full Boltzmann equation, are investigated. The capability of the current approach for simulating non-equilibrium flow physics is validated through satisfactory in- and out-of-distribution performance. The open-source codes to reproduce the numerical results are available under the MIT license [1].

© 2023 Elsevier Inc. All rights reserved.

* Corresponding author.

E-mail address: txiao@imech.ac.cn (T. Xiao).

1. Introduction

The advances in machine learning are increasingly incorporated in computational physics to enable versatile data-driven modeling and simulation. Interestingly, not so long ago, many were convinced that there are significant differences between scientific computing and machine learning tasks. For example, scientific computing generally processes smaller but more computationally intensive amounts of data. These data from different locations are expected to be linked by physical principles, whereas classical machine learning mainly deals with discrete and localized data. However, recent progress, e.g., physics-informed neural networks [2], suggests that there is ample room for building data-efficient and physics-enhanced machine learning approaches for scientific applications at the intersection of classical numerical methods.

Among possible applications, a key use of scientific machine learning (SciML) is to help establish reliable physical models. This approach is especially crucial when first-principle models are missing, e.g., fitting potential-energy hypersurfaces whose topographical features are sufficiently close to those of real but unknown surfaces such that the simulation results from molecular dynamics (MD) or Monte-Carlo (MC) methods are experimentally meaningful [3,4]. Another promising direction is to seek simplified models for high-dimensional and nonlinear terms. An appropriate surrogate model can significantly reduce the dimensionality and complexity of differential and integral equations, thereby improving computational efficiency while maintaining satisfactory accuracy [5,6].

The SciML-integrated model is expected to be as universally accurate as possible to the ground-truth physics. It should state fundamental physical laws, e.g., the conservation of mass in fluid dynamic equations. It is desirable to preserve the structural properties under physical constraints, e.g., symmetry, invariance, and non-decreasing entropy. Different from the brute-force solution, this requires us to design physics-oriented architectures, parameters, and connections in the machine learning model, or seek efficient regularization approaches [7–10]. Ideally, one would like to perform a small set of experiments under idealized situations and obtain a model that works under more general conditions [11]. In other words, we expect the trained model to extrapolate well for both in- and out-of-distribution cases [12] and the generalization gap to be as small as possible [13]. This further requires that the model should have clear physical meaning and interpretability rather than pure black-box nature.

In this paper, we focus on the kinetic theory of gases. Lying at its core, the Boltzmann equation describes the evolution of a many-particle system in a statistical manner, which serves as a cornerstone of many hydrodynamic and thermodynamic theories [14,15]. The Boltzmann equation is an integro-differential equation defined in a high-dimensional phase space. The extremely high dimensionality and nonlinearity pose a notorious challenge for theoretical analysis and numerical simulation. Starting from the introduction of the relaxation-type model [16], tremendous efforts have been made to seek a sufficiently accurate simplified model of the Boltzmann equation. To further improve the accuracy in highly dissipative regions and preserve the correct fluid dynamic limit, without changing the form of the relaxation model, explorations have been conducted to modify the objective equilibrium function [17,18] and the relaxation frequency [19,20]. Although these improved models can provide the correct Prandtl number, the accuracy in describing strongly non-equilibrium flows is less satisfactory [21].

The burgeoning scientific machine learning offers an alternative for establishing simplified models of the Boltzmann equation. The existence of the multifold integral term precludes direct use of automatic differentiation-based techniques [2,22]. Preliminary attempts have been made to build simplified surrogate models for the Boltzmann collision integral. The underlying techniques of related work involve neural differential equations [23], reduced order modeling [24], and spectral representations [25]. Despite the good results obtained by these methods, very limited proof has been provided showing that the numerical method preserves the key structures, e.g., the H-theorem [26], of a many-particle system. Therefore, the preservation of such physical constraints must rely on the reliability of the dataset and the quality of training. It is notoriously known that both tasks are difficult to be perfect, and the generalization performance of machine learning models based on this paradigm can be questionable.

In this paper, we exploit the structural similarity of the BGK-type relaxation model and the residual neural network (ResNet) to construct what we call the relaxation neural network (RelaxNet), which plays as a surrogate model for the Boltzmann collision integral with quantitative interpretability. Specifically, two feed-forward neural networks with physics-informed connections and bounded activation functions are introduced as building blocks to provide physically realizable approximations of the parameterized equilibrium distribution and velocity-dependent relaxation time. The evaluation of collision term is thus significantly accelerated since the computation of convolutions in the fivefold integral is replaced by the tensor multiplication, addition, and activation in the neural network. The advection operator (left-hand side) of the original Boltzmann equation and the neural network-based collision term are fused into a trainable framework, i.e., the so-called universal Boltzmann equation (UBE) [6,23]. We prove that the RelaxNet-based UBE preserves key structural properties. The positivity, invariance, correct fluid dynamic limit, and hyperbolicity of advection operator hold independent of trainable parameters. The conservation and H-theorem are preserved up to the training error. The implementation of the RelaxNet-based UBE, training strategy, and solution algorithm are presented in detail.

The paper is organized as follows. In Section 2, we introduce some fundamental concepts in the kinetic theory of gases. Section 3 presents the idea and architecture of RelaxNet as well as the training method. Section 4 describes the methodology for generating datasets for training and testing. Section 5 details the numerical solution algorithm for the RelaxNet-based universal Boltzmann equation. Section 6 contains the numerical experiments for both spatially homogeneous and inhomogeneous cases to validate the current methodology. The last section is the conclusion. The nomenclature of this paper can be found in Table 1.

Table 1
Nomenclature.

$t, \mathbf{x}, \mathbf{v}$	time, space, and particle velocity variables
f	particle distribution function
\mathcal{Q}, \mathcal{A}	collision and advection operators of the Boltzmann equation
ν	relaxation frequency
ψ	collision invariants
H	function used in the H-theorem
\mathcal{M}	Maxwellian distribution function
ρ, \mathbf{V}, T, E	density, bulk velocity, temperature, and energy
\mathbf{c}	peculiar velocity
m	molecular mass
k	Boltzmann constant
\mathcal{E}	equilibrium state in the relaxation model
σ	activation function
\mathcal{E} -net	subnet in RelaxNet to approximate equilibrium state
τ -net	subnet in RelaxNet to approximate relaxation time
D	dimension
τ	relaxation time
$\bar{\nu}, \bar{\tau}$	mean relaxation frequency and time
μ	viscosity
p	pressure
\mathcal{M}^+	correction term for equilibrium distribution
α, β	outputs of \mathcal{E} -net and τ -net
ELU	Exponential Linear Unit function
θ	trainable parameters
NN_θ	RelaxNet function
θ_1, θ_2	The parts of θ belonging to \mathcal{E} -net and τ -net
C	cost function
λ, ξ	regularization strengths
ε	minimum value of cost function
$\hat{\alpha}$	Coefficients of Maxwellian in summation form of collision invariants
$\tilde{\alpha}$	product of α and $\hat{\alpha}$
\mathbf{P}, \mathbf{q}	pressure tensor and heat flux
\mathbf{m}	moment variables
ϕ	basis functions
N_m	polynomial degree of ϕ
γ	Lagrange multipliers of the entropy closure problem
\mathcal{H}	Heaviside step function
N_i	number of samples
$f_{\text{ref}}, \mathcal{Q}_{\text{ref}}$	referenced distribution function and collision term in the dataset
N_h	number of hidden layers
CFL	Courant–Friedrichs–Lewy number

2. Kinetic theory

The kinetic theory of gases describes the evolution of a many-particle system in terms of the particle distribution function $f(t, \mathbf{x}, \mathbf{v})$, where $t \in \Gamma \subseteq \mathbb{R}^+$ is time variable, $\mathbf{x} \in \mathcal{D} \subseteq \mathbb{R}^3$ is space variable, and $\mathbf{v} \in \mathbb{R}^3$ is particle velocity. For dilute monatomic gas in the absence of external force, the Boltzmann equation can be written as

$$\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f = \mathcal{Q}(f, f) = \int_{\mathbb{R}^3} \int_{\mathbb{S}^2} [f(\mathbf{v}') f(\mathbf{v}'_*) - f(\mathbf{v}) f(\mathbf{v}_*)] \mathcal{B}(\cos \theta, g) d\Omega d\mathbf{v}_*, \quad (1)$$

where $\{\mathbf{v}, \mathbf{v}_*\}$ denotes the pre-collision velocities of two classes of colliding particles, and $\{\mathbf{v}', \mathbf{v}'_*\}$ is the corresponding post-collision velocities. The collision kernel $\mathcal{B}(\cos \theta, g)$ measures the probability of collisions in different directions, where θ is the deflection angle and $g = |\mathbf{g}| = |\mathbf{v} - \mathbf{v}_*|$ is the magnitude of relative pre-collision velocity. The solid angle Ω denotes the unit vector along the relative post-collision velocity $\mathbf{v}' - \mathbf{v}'_*$, and the deflection angle satisfies the relation $\theta = \Omega \cdot \mathbf{g}/g$. Note that if we define the collision frequency as

$$\nu(\mathbf{v}) = \int_{\mathbb{R}^3} \int_{\mathbb{S}^2} f(\mathbf{v}_*) \mathcal{B}(\cos \theta, g) d\Omega d\mathbf{v}_*, \quad (2)$$

the collision term in Eq. (1) can be written as a combination of gain and loss, i.e.,

$$\begin{aligned} \mathcal{Q}(f, f) &= \mathcal{Q}^+(f, f) + \mathcal{Q}^-(f), \\ \mathcal{Q}^+ &= \int_{\mathbb{R}^3} \int_{\mathbb{S}^2} f(\mathbf{v}') f(\mathbf{v}'_*) \mathcal{B}(\cos \theta, g) d\Omega d\mathbf{v}_*, \quad \mathcal{Q}^- = \nu(\mathbf{v}) f(\mathbf{v}). \end{aligned} \quad (3)$$

The Boltzmann equation possesses structural properties that reflect the physical constraints of a many-particle system. Such structure is manifested in the properties of the collision operator \mathcal{Q} and the advection operator $\mathcal{A} = \partial_t + \mathbf{v} \cdot \nabla_{\mathbf{x}}$, and it plays a key role in design of the analytical and numerical solution algorithms of the Boltzmann equation. These structural properties can be highlighted as follows [27].

I *Invariant range*: There exists a set $B \subseteq [0, \infty)$ consistent with the physical bounds of f such that $\text{Range}(f(t, \cdot, \cdot)) \subseteq B$ if $\text{Range}(f(0, \cdot, \cdot)) \subseteq B$. A minimum requirement for the invariant range is $f \geq 0$ since f represents the density of particles.

II *Conservation*: There exists the collision invariants $\boldsymbol{\psi} = (1, \mathbf{v}, \mathbf{v}^2/2)^T$ such that

$$\langle \boldsymbol{\psi} \mathcal{Q}(g, g) \rangle = 0, \quad \forall g \in \text{Dom}(\mathcal{Q}), \tag{4}$$

where $\langle \cdot \rangle = \int_{\mathbb{R}^3} \cdot d\mathbf{v}$ denotes the integral over particle velocity space. This property leads to the conservation laws

$$\partial_t \langle \boldsymbol{\psi} f \rangle + \nabla_{\mathbf{x}} \cdot \langle \mathbf{v} \boldsymbol{\psi} f \rangle = 0, \tag{5}$$

where $\mathbf{W}(t, \mathbf{x}) = (\rho, \rho \mathbf{V}, \rho E)^T = \langle \boldsymbol{\psi} f \rangle$ are named as conservative variables in fluid mechanics. These velocity-space integrals are also commonly referred as moments of particle distribution function f .

III *Hyperbolicity*: For each fixed \mathbf{v} , the advection operator \mathcal{A} is hyperbolic over $(t, \mathbf{x}) \in \Gamma \times \mathcal{D}$.

IV *H-theorem*: There exists a differentiable function $H(f) = -f \log f$ such that

$$\langle H'(g) \mathcal{Q}(g, g) \rangle \leq 0, \quad \forall g \in \text{Dom}(\mathcal{Q}) \text{ s.t. } \text{Range}(g) \subseteq \mathcal{D}, \tag{6}$$

which further results in the dissipation law

$$\partial_t \langle H(f) \rangle + \nabla_{\mathbf{x}} \cdot \langle \mathbf{v} H(f) \rangle \leq 0. \tag{7}$$

The H-theorem implies that H is a Lyapunov function for the Boltzmann equation, and the logarithm of its minimizer \mathcal{M} must be a linear combination of the collision invariants $\boldsymbol{\psi}$, which can be further proved to have the following form, i.e., the Maxwellian distribution function [28],

$$\mathcal{M} := \rho \left(\frac{m}{2\pi kT} \right)^{3/2} \exp\left(-\frac{m}{2kT} \mathbf{c}^2 \right), \tag{8}$$

where m is molecular mass, k is the Boltzmann constant, T is temperature, and $\mathbf{c} = \mathbf{v} - \mathbf{V}$ is the peculiar velocity.

V *Galilean invariance*: There exists the Galilean transformation defined by

$$\mathcal{G}(g)(t, \mathbf{x}, \mathbf{v}) = g(t, \mathbf{x} - \boldsymbol{\omega}t, \mathbf{v} - \boldsymbol{\omega}), \tag{9}$$

where $\boldsymbol{\omega} \in \mathbb{R}^3$ is a translational velocity, that commute with the advection and collision operators, i.e.,

$$\begin{aligned} \mathcal{A}(\mathcal{G}(g)) &= \mathcal{G}(\mathcal{A}(g)), \quad \forall g \in \text{Dom}(\mathcal{A}), \\ \mathcal{Q}(\mathcal{G}(g)) &= \mathcal{G}(\mathcal{Q}(g)), \quad \forall g \in \text{Dom}(\mathcal{Q}). \end{aligned} \tag{10}$$

As a consequence, the transformed particle distribution function $g = \mathcal{G}(f)$ still satisfies Eq. (1) if f is a solution of it.

Following the structure in Eq. (3) and the H-theorem, several relaxation models have been developed to simplify the computation of the fivefold integral on the right-hand side of the Boltzmann equation. These simplified models take the form

$$\mathcal{Q}(f) = \nu(\mathcal{E} - f), \tag{11}$$

where \mathcal{E} denotes the target equilibrium state and ν is the relaxation frequency. In the Bhatnagar–Gross–Krook (BGK) model [16], the equilibrium distribution \mathcal{E} is set as Maxwellian, and ν is independent of \mathbf{v} , which implies that particles with different velocities share the same probability of collision. Although the BGK model can satisfy the structural properties shown in Section 2, the underlying assumption becomes particularly unreasonable when the magnitude of ν in rarefied gases is moderate, which makes the BGK model inaccurate in describing highly dissipative flows. One direction to improve the BGK model is to introduce \mathbf{v} -dependent relaxation frequency in Eq. (11) [19,20]. Following the principle that high-speed particles should have a greater probability of collision, these models attempt to recover the true collision frequency in Eq. (2). Another idea is to supplement correction terms in \mathcal{E} based on high-order moments of f , e.g., stress tensor [17] and heat flux [18], which builds a multi-level relaxation mechanism towards Maxwellian. It is shown that both strategies above can provide the correct Prandtl number. However, no model has yet provided as satisfactory the solution as the full Boltzmann equation in describing the evolution of gases far away from equilibrium.

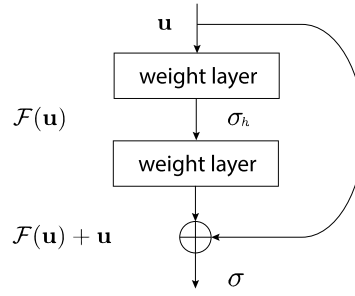


Fig. 1. Schematic diagram of a building block in the residual neural network (ResNet).

3. Relaxation neural network

3.1. Deep residual learning

Our idea is to build a surrogate model of the nonlinear function described by the Boltzmann equation based on a neural network. To preserve as many structural properties as shown in Section 2, we prefer a physics-oriented architecture rather than a naive feed-forward neural network.

We choose to construct the model on the basis of residual neural network (ResNet). To clarify this, we briefly introduce here the basic principles of ResNets. The concept of residual learning was proposed to mitigate the degradation problem in deep neural networks, i.e., as the network depth increases, the accuracy gets saturated and then degrades rapidly. This phenomenon implies that it is non-trivial to learn a deeper model based on its shallower counterpart in which the added layers only perform identity mapping. Therefore, instead of building each few stacked layers to fit a desired underlying mapping directly, one can explicitly let these layers fit a residual mapping. It has been shown that it is easier to optimize the residual mapping than the original unreferenced mapping [29].

Fig. 1 shows the schematic diagram of the building block in a ResNet. We denote $\mathcal{H}(\mathbf{u})$ as the underlying mapping to be fit by a few stacked layers and \mathbf{u} as the input to the first layer. If a continuous function can be approximated asymptotically by multiple nonlinear layers, then we can equivalently conclude that they can asymptotically approximate the residual function, i.e., $\mathcal{F}(\mathbf{u}) = \mathcal{H}(\mathbf{u}) - \mathbf{u}$, provided that the input and output are of the same dimensions. The original function thus becomes $\mathcal{H}(\mathbf{u}) = \mathcal{F}(\mathbf{u}) + \mathbf{u}$, and its formulation is realized by adding shortcut connections [30] to feedforward or convolutional neural networks. The shortcut can skip one or several hidden layers and performs identity mapping. The nonlinear activation function inside the hidden layers is denoted as σ_h . The addition operation can be regarded as the connection function which combines the contributions from \mathcal{F} and identity mapping of \mathbf{u} . The residual block is further activated with σ , and the final output can be written as

$$\mathcal{N}(\mathbf{u}) = \sigma(\mathcal{F}(\mathbf{u}) + \mathbf{u}). \quad (12)$$

It is noticeable that the relaxation model of the Boltzmann equation in Eq. (11) has a similar structure to the ResNet. Specifically, if the particle distribution function is considered as the input to the solver of relaxation term, then the equilibrium distribution function $\mathcal{E} = \mathcal{E}(f)$ plays an equivalent role to that of \mathcal{H} . Next, unlike in neural networks where addition is employed as the connection function, subtraction is used in the relaxation model. Finally, the relaxation frequency can be viewed as an activation function acting on the equilibrium function \mathcal{E} and the shortcut connection from f . The building block of the solver of relaxation term can thus be written as

$$\mathcal{Q}(f) = \nu(\mathcal{E}(f) - f). \quad (13)$$

The structural similarity between the relaxation model of Boltzmann equation in Eq. (13) and ResNet in Eq. (12) can be easily noticed. This provides us an opportunity to build a physics-oriented neural network based on the relaxation model and preserve the structural properties of the Boltzmann equation.

3.2. Relaxation neural network

Following the spirit of the relaxation model and ResNet, we build a novel neural network model which is named as relaxation neural network (RelaxNet). The neural network is built on top of the discrete velocity formulation of the Boltzmann equation, i.e., the particle distribution function is discretized in the velocity space and the macroscopic variables as its moments can be evaluated by numerical quadrature. The schematic diagram of the model with forward pass is shown in Fig. 2. As can be seen, RelaxNet consists of two feedforward subnets and the corresponding activation and connection functions. We describe these components in detail below.

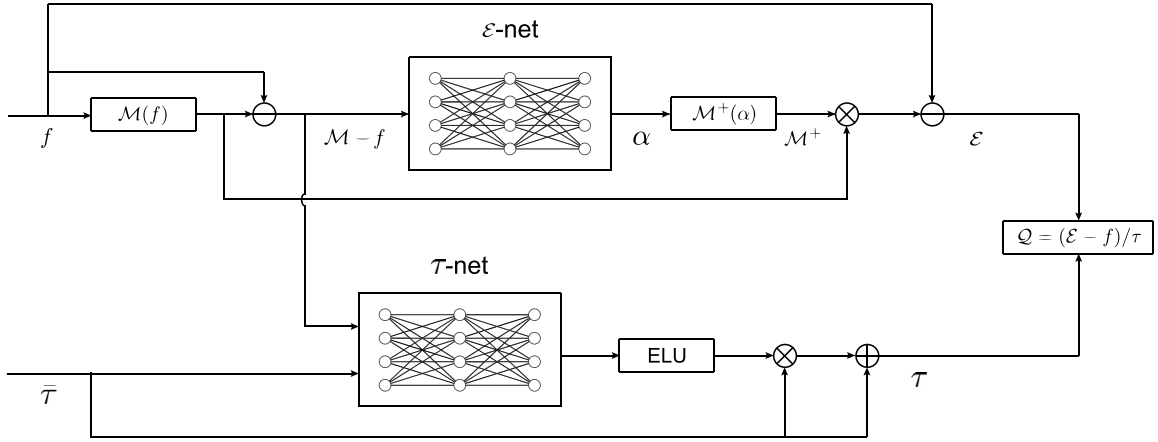


Fig. 2. Schematic diagram of forward pass in the relaxation neural network (RelaxNet).

(1) Neural networks

- \mathcal{E} -net: feedforward network $\mathcal{N}_{\mathcal{E}} : \mathbb{R}^{N_v} \rightarrow \mathbb{R}^{D+2}$, where N_v is the number of quadrature points to discretize f in the velocity space and $D \in \{1, 2, 3\}$ is the dimension of the flow problem of interest. The neural network takes the difference between Maxwellian and distribution function, i.e., $\mathcal{M} - f$, as input. The output of neural network is a vector $\alpha = (\alpha_1, \alpha_2, \alpha_3)^T$, where α_2 is a vector of dimension D . The hyperbolic tangent (\tanh) function is employed as the activation function for the hidden layers. We require the \mathcal{E} -net to be a bias-free neural network, i.e., it has only weights without biases in each layer.
- τ -net: feedforward network $\mathcal{N}_{\tau} : \mathbb{R}^{N_v+1} \rightarrow \mathbb{R}^{N_v}$. The input to the neural network $(\mathcal{M} - f, \bar{\tau})^T$ is a vector of dimension $N_v + 1$. Here $\bar{\tau}$ is the mean relaxation time calculated by

$$\bar{\tau} = \frac{\mu}{p}, \tag{14}$$

where μ is the viscosity that can be decided from a molecular model, and p is the pressure. The relationship between mean relaxation time and frequency is $\bar{\nu}\bar{\tau} = 1$. The output of neural network β is a vector of the same dimension as f . The hyperbolic tangent (\tanh) function is used as the activation function for the hidden layers. We likewise require the τ -net to be a bias-free neural network.

(2) Structure functions

- $\mathcal{M}(f)$: Maxwellian function described in Eq. (8), where the macroscopic variables can be evaluated as

$$\rho = \langle f \rangle, \quad \mathbf{V} = \frac{\langle \mathbf{v}f \rangle}{\langle f \rangle}, \quad T = \frac{m}{3k\rho} \langle \mathbf{c}^2 f \rangle. \tag{15}$$

Note that the above integrals can only be approximated via numerical quadrature when the distribution function has been discretized with N_v collocation points, i.e., $\langle a \rangle \simeq \sum w_q a_q$, where a denotes any quantity of interest.

- $\mathcal{M}^+(\alpha)$: Constructor of the correction term for equilibrium distribution via α , i.e.,

$$\mathcal{M}^+ = \exp(\alpha \cdot \psi) = \exp\left(\alpha_1 + \alpha_2 \cdot \mathbf{v} + \frac{1}{2}\alpha_3 \mathbf{v}^2\right), \tag{16}$$

(3) Connection and activation functions

- ELU: Exponential Linear Unit function defined as

$$R(z) = \begin{cases} z, & z > 0, \\ e^z - 1, & z \leq 0. \end{cases} \tag{17}$$

- \oplus : Addition.
- \ominus : Subtraction.
- \otimes : Multiplication.

Given the RelaxNet defined above, a neural network enhanced Boltzmann equation can be formulated as

$$\begin{aligned} \partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f &= \text{NN}_{\theta}(f, \bar{\tau}) = \nu(\mathcal{E} - f), \quad f = \{f(v_i)\}_{i=1:N_v}, \\ \mathcal{E} &= \mathcal{M}\mathcal{M}_{\theta_1}^+, \quad \nu = \frac{1}{\tau} = \frac{1}{\bar{\tau}(1 + \text{ELU}(\beta_{\theta_2}))}, \end{aligned} \tag{18}$$

where $\theta = (\theta_1, \theta_2)^T$ denotes trainable parameters in the RelaxNet model. Such an equation is consistent of the idea of universal differential equation [6], where the mechanical advection operator $\mathcal{A} = \partial_t + \mathbf{v} \cdot \nabla_{\mathbf{x}}$ and the neural collision operator NN_θ constitute a differentiable surrogate model. Therefore, we can call Eq. (18) as the universal Boltzmann equation (UBE). Note that since the number of input neurons in the RelaxNet needs to correspond to the number of collocation points N_v , the UBE holds pointwise. The continuous formulation in Eq. (18) should be understood as the asymptotic limit with $N_v \rightarrow \infty$.

3.3. Training strategy

UBE in Eq. (18) is a data-driven model and it forms a typical supervised training task. Given a dataset consisting of reference solutions to the Boltzmann equation, an optimization problem can be set up which minimizes the difference between the current prediction of UBE model and ground-truth solution. We define the cost function as

$$C(\theta) = \frac{1}{N_t N_x} \sum_{n,i,j} \|\text{NN}_\theta(\hat{f}_{i,j}^n, \hat{\tau}_i^n) - \mathcal{Q}_{\text{ref}}(\hat{f}_{i,j}^n)\| + \frac{\lambda}{N_t N_x} \sum_{n,i} \|\langle \psi | \text{NN}_\theta(\hat{f}_{i,j}^n, \hat{\tau}_i^n) \rangle\| + \xi \sum_l \|\theta^l\|, \quad (19)$$

where $\|\cdot\|$ denotes the Euclidean distance, \hat{f} denotes the particle distribution function generated in the dataset and $\hat{\tau}$ is the mean relaxation time with respect to it. The superscript and subscripts $\{n, i, j\}$ are the indices of the solution collocation points at $\{t^n, \mathbf{x}_i, \mathbf{v}_j\}$. Note that here t^n does not represent any actual physical time, but only the ordering of samples in the dataset. The generation of datasets used for the above function will be detailed in Section 4.

The latter two terms in Eq. (19) serve as regularization terms to mitigate overfitting. The second term in Eq. (19) comes naturally from the structural property II shown in Section 2, i.e., the collision term should conserve mass, momentum and energy in the Boltzmann equation. This physics-informed regularization [31] leads to more accurate preservation of conservation laws, and improves the physical interpretability of the model. The last term in Eq. (19) sums over the squared weight parameters of the network, where θ^l denotes the weight parameters of l -th layers and L is the total number of layers. The parameters $\{\lambda, \xi\}$ are the regularization strengths. We require

$$\lambda N_t N_x > 1, \quad \xi N_t N_x N_v > 1, \quad (20)$$

where $\{N_t, N_x, N_v\}$ are the number of time steps, physical cells, and velocity points respectively in the dataset.

The cost function in Eq. (19) can then be minimized by different optimization algorithms. In this paper, we adopt the Adam [32] method which is an improvement of the stochastic gradient descent method with adaptive moment estimation. The gradient of Eq. (19) with respect to θ is computed with the help of reverse-mode automatic differentiation (AD). Consider a smooth function $y = \mathcal{F}(x)$, the reverse-mode AD computes the dual (conjugate-transpose) matrix of Jacobian $\mathcal{J} = \nabla \mathcal{F}$ at $x = x_0$ with the chain rule

$$(\mathcal{J}(\mathcal{F})(x_0))^* = (\mathcal{J}(G_1)(x_0))^* \times \cdots \times (\mathcal{J}(G_k)(x_{k-1}))^*, \quad (21)$$

with $x_i := G_i(x_{i-1})$ for $i = 1, \dots, k-1$. The implementation of the reverse-mode AD can be found in the open-source library Zygote [33].

3.4. Structure of RelaxNet

The universal Boltzmann equation based on RelaxNet preserve the structural properties illustrated in Section 2. We prove these properties in detail below. Note that since the RelaxNet has a finite number of input neurons (with respect to the number of quadrature points N_v), the following analysis is based on a discrete perspective. The formal continuous description can be understood as the asymptotic limit with $N_v \rightarrow \infty$.

Theorem 1. *Let f be the solution of the RelaxNet-based universal Boltzmann equation given in Eq. (18), then the range of f satisfies*

$$\text{Range}(f(t, \mathbf{x}, \mathbf{v})) \subseteq B = \{f : 0 \leq f < \infty\}, \quad (22)$$

given $\text{Range}(f(0, \mathbf{x}, \mathbf{v})) \subseteq B$.

Proof. We define shifted f along characteristics as

$$f^\#(t, \mathbf{x}, \mathbf{v}) = f(t, \mathbf{x} + \mathbf{v}t, \mathbf{v}). \quad (23)$$

Then UBE in Eq. (18) can be written as

$$\frac{d}{dt} f^\# + \nu f^\# = \nu \mathcal{E}^\#, \quad (24)$$

where d/dt denotes the full derivative. Given the initial value

$$f(0, \mathbf{x}, \mathbf{v}) = f_0(\mathbf{x}, \mathbf{v}), \tag{25}$$

then the solution of the initial value problem consisting of Eq. (24) and (25) can be written as [34]

$$f^\#(t, \mathbf{x}, \mathbf{v}) = f_0(\mathbf{x}, \mathbf{v})e^{-\nu t} + \int_0^t e^{-\nu(t-s)} \nu \mathcal{E}^\#(s, \mathbf{x}, \mathbf{v}) ds. \tag{26}$$

Proving Theorem 1 is equivalent to proving that $\text{Range}(f^\#) \subseteq B$. As shown in Section 3.2, \mathcal{E} and ν can be expressed as

$$\begin{aligned} \mathcal{E} &= \mathcal{M}\mathcal{M}^+, \quad \mathcal{M}^+ = \exp(\boldsymbol{\alpha}_{\theta_1} \cdot \boldsymbol{\psi}), \\ \nu &= \frac{1}{\tau}, \quad \tau = \bar{\tau}(1 + \text{ELU}(\boldsymbol{\beta}_{\theta_2})), \end{aligned} \tag{27}$$

where $\theta = (\theta_1, \theta_2)^T$ denotes all the trainable parameters. With the help of the physics-informed activation functions, i.e., ELU and ReLU, it is clear that \mathcal{E} and ν in Eq. (27) are non-negative independent of θ . Besides, since the hidden layers inside \mathcal{E} -net and τ -net are activated by the tanh function, the following relation holds,

$$\text{Range}(\mathcal{E}) \subseteq B, \quad \text{Range}(\nu) \subseteq B \tag{28}$$

given any finite layers and parameters. So far, Theorem 1 is proved. \square

Theorem 2. Let NN_θ be the optimized RelaxNet, then the moments of NN_θ with respect to the collision variants $\boldsymbol{\psi}$ satisfy

$$\|\langle \boldsymbol{\psi} \text{NN}_\theta(\mathbf{g}, \bar{\tau}) \rangle\| \leq \varepsilon, \quad \forall (\mathbf{g}, \bar{\tau}) \in \text{Dom}(\text{NN}_\theta), \tag{29}$$

where ε is the minimum value of the cost function.

Proof. The optimized cost function in Eq. (19) satisfy

$$C(\theta) = \sum_{n,i,j} \|\text{NN}_\theta - \mathcal{Q}\|(t^n, \mathbf{x}_i, \mathbf{v}_j) + \lambda \sum_{n,i} \|\langle \boldsymbol{\psi} \text{NN}_\theta \rangle\|(t^n, \mathbf{x}_i) + \xi \sum_l^L \|\theta^l\| = \varepsilon, \tag{30}$$

which results in

$$\lambda \sum_{n,i} \|\langle \boldsymbol{\psi} \text{NN}_\theta \rangle\|(t^n, \mathbf{x}_i) \leq \varepsilon. \tag{31}$$

Due to the requirement in Eq. (20), we have

$$\|\langle \boldsymbol{\psi} \text{NN}_\theta \rangle\| \leq \varepsilon, \tag{32}$$

at any solution point in the dataset. \square

Proposition 3. For each fixed \mathbf{v} , the advection operator \mathcal{A} in the RelaxNet-based universal Boltzmann equation is hyperbolic over $(t, \mathbf{x}) \in \Gamma \times \mathcal{D}$.

Proof. As RelaxNet surrogates only the right-hand side of the Boltzmann equation, this theorem naturally holds according to Property (3) in Section 2. \square

Theorem 4. The function $H(f) = -f \log f$ satisfies

$$\partial_t \langle H(f) \rangle + \nabla_{\mathbf{x}} \cdot \langle \mathbf{v} H(f) \rangle \leq \sum_i \chi_i \varepsilon, \tag{33}$$

where $\boldsymbol{\chi} = \sum_i \chi_i$ is a vector of the same dimension of $\boldsymbol{\alpha}$. The H-theorem holds asymptotically as $\varepsilon \rightarrow 0$.

Proof. Substituting $H'(f) = -(1 + \log f)$ into the RelaxNet-based UBE in Eq. (18) and considering Eq. (27) yield

$$\langle (\partial_t f + \mathbf{v} \cdot \nabla_{\mathbf{x}} f)(1 + \log f) \rangle = \partial_t \langle H(f) \rangle + \nabla_{\mathbf{x}} \cdot \langle \mathbf{v} H(f) \rangle = \langle \nu(\mathcal{E} - f)(1 + \log f) \rangle. \tag{34}$$

Notice that the Maxwellian distribution in Eq. (8) can be written as

$$\mathcal{M} = \exp(\hat{\boldsymbol{\alpha}} \cdot \boldsymbol{\psi}), \quad (35)$$

where

$$\begin{aligned} \hat{\boldsymbol{\alpha}} &= (\hat{\alpha}_1, \hat{\alpha}_2, \hat{\alpha}_3)^T, \\ \hat{\alpha}_1 &= \log \left(\rho \left(\frac{m}{2\pi kT} \right)^{D/2} - \frac{m\mathbf{V}^2}{2kT} \right), \quad \hat{\alpha}_2 = \frac{m\mathbf{V}}{kT}, \quad \hat{\alpha}_3 = -\frac{m}{kT}. \end{aligned} \quad (36)$$

Given the structure function \mathcal{M}^+ defined in Eq. (16), the equilibrium state can be written as

$$\mathcal{E} = \mathcal{M}\mathcal{M}^+ = \exp(\tilde{\boldsymbol{\alpha}} \cdot \boldsymbol{\psi}) \quad (37)$$

where $\tilde{\alpha}_i = \alpha_i + \hat{\alpha}_i$. From Theorem 2, we obtain

$$\langle \nu(\mathcal{E} - f) \log \mathcal{E} \rangle = \left\langle \nu(\mathcal{E} - f) \left(\tilde{\alpha}_1 + \tilde{\boldsymbol{\alpha}}_2 \cdot \mathbf{v} + \frac{1}{2} \tilde{\alpha}_3 \mathbf{v}^2 \right) \right\rangle \leq \sum_i \tilde{\alpha}_i \varepsilon. \quad (38)$$

The right-hand side of Eq. (34) can be written as

$$\langle \nu(\mathcal{E} - f)(1 + \log f) \rangle = \langle \nu(\mathcal{E} - f) \rangle + \langle \nu(\mathcal{E} - f) \log f \rangle \leq \varepsilon + \langle \nu(\mathcal{E} - f) \log f \rangle, \quad (39)$$

where the last inequality comes from Theorem 2. We rewrite the last term of the above equation as

$$\langle \nu(\mathcal{E} - f) \log f \rangle = \langle \nu(\mathcal{E} - f)(\log f - \log \mathcal{E}) \rangle + \langle \nu(\mathcal{E} - f) \log \mathcal{E} \rangle. \quad (40)$$

Since the logarithm is a monotonically increasing function, the first in the above equation is non-positive. Considering Eq. (34), (38), and (40), we end up with

$$\partial_t \langle H(f) \rangle + \nabla_{\mathbf{x}} \cdot \langle \mathbf{v}H(f) \rangle \leq \left(1 + \sum_i \tilde{\alpha}_i \right) \varepsilon. \quad (41)$$

So far, Theorem 4 is proved. \square

Proposition 5. *The RelaxNet-based universal Boltzmann equation is Galilean invariant if the values of distribution function at quadrature points remains unchanged.*

Proof. We assume f is a solution to UBE in Eq. (18) and consider the Galilean transformation defined in Eq. (9), i.e.,

$$g(t, \mathbf{x}, \mathbf{v}) = \mathcal{G}(f) = f(t, \mathbf{x} - \boldsymbol{\omega}t, \mathbf{v} - \boldsymbol{\omega}). \quad (42)$$

As the advection operator in UBE is the same as the original Boltzmann equation, the commute naturally holds, i.e.,

$$\mathcal{A}(\mathcal{G}(f)) = \mathcal{G}(\mathcal{A}(f)). \quad (43)$$

For the collision operator, from the definition of macroscopic variables $\mathbf{W} = \langle \boldsymbol{\psi} f \rangle$, we have

$$\rho_g = \rho_f, \quad T_g = T_f, \quad \mathbf{V}_g = \mathbf{V}_f + \boldsymbol{\omega}, \quad (44)$$

where the subscripts f and g denote the flow quantities associated with the distribution function before and after the transformation, respectively. Given the definition of the Maxwellian in Eq. (8) and mean relaxation time in Eq. (14), we have

$$\mathcal{M}_g = \mathcal{G}(\mathcal{M}_f), \quad \bar{\tau}_g = \bar{\tau}_f, \quad (45)$$

which yields

$$(\mathcal{M}_g - g) = \mathcal{G}(\mathcal{M}_f - f). \quad (46)$$

Our goal is to prove

$$\frac{\mathcal{M}_g \mathcal{M}_g^+ - g}{\tau_g} = \mathcal{G} \left(\frac{\mathcal{M}_f \mathcal{M}_f^+ - f}{\tau_f} \right). \quad (47)$$

The remaining parts in Eq. (18), i.e., \mathcal{M}^+ and τ , are determined by \mathcal{E} -net and τ -net. Note that the number of input neurons needs to correspond to the number of quadrature points N_v . As defined in Section 3.2, since the value of velocities does not appear in the input value of both neural networks, the predictions of \mathcal{E} -net and τ -net remain pointwise unchanged

as long as the input distribution function $\{f, g\}$ and mean relaxation time $\bar{\tau}$ are the same at each quadrature point. This requires the quadrature points (at least those on which the value of the distribution function is greater ε) to undergo the same underlying Galilean transformation \mathcal{G} instead of being mere Eulerian grid points. After such conditions are met, we arrive at

$$\mathcal{Q}(\mathcal{G}(f), \bar{\tau}) = \mathcal{G}(\mathcal{Q}(f, \bar{\tau})). \quad (48)$$

Therefore, g is also a solution to Eq. (18). \square

Theorem 6. *The RelaxNet-based universal Boltzmann equation preserves the correct continuum limit.*

Proof. Based on Theorem 2, we obtain the transport equations for conservative variables

$$\partial_t \langle \boldsymbol{\psi} f \rangle + \nabla_{\mathbf{x}} \cdot \langle \mathbf{v} \boldsymbol{\psi} f \rangle \leq \varepsilon, \quad (49)$$

or specifically,

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho \mathbf{V} \\ \rho E \end{pmatrix} + \nabla_{\mathbf{x}} \cdot \begin{pmatrix} \rho \mathbf{V} \\ \rho \mathbf{V} \mathbf{V} \\ \rho E \mathbf{V} \end{pmatrix} \leq -\nabla_{\mathbf{x}} \cdot \begin{pmatrix} 0 \\ \mathbf{P} \\ \mathbf{P} \cdot \mathbf{V} + \mathbf{q} \end{pmatrix} + \begin{pmatrix} \varepsilon \\ \varepsilon \\ \varepsilon \end{pmatrix}, \quad (50)$$

where

$$\mathbf{P} = \langle \mathbf{c} \mathbf{c} f \rangle, \quad \mathbf{q} = \frac{1}{2} \langle \mathbf{c} \mathbf{c}^2 f \rangle, \quad (51)$$

denote the pressure tensor and heat flux respectively. The UBE in Eq. (18) can be rewritten as

$$f = \mathcal{E} - \tau (\mathcal{E}_t + \mathbf{v} \cdot \nabla_{\mathbf{x}} \mathcal{E}) + O(\tau^2), \quad (52)$$

where $\tau = 1/\nu$ is the relaxation time. In the continuum limit, infinitely many intermolecular interactions happen in a unit of time, i.e., $\bar{\nu} \rightarrow \infty$ and $\bar{\tau} \rightarrow 0$. From Eq. (27), we have $\tau \rightarrow 0$ and thus $f \rightarrow \mathcal{E}$. We next prove that \mathcal{E} converges to \mathcal{M} in the continuum limit. Note that the difference between Maxwellian and distribution function, i.e., $\mathcal{M} - f$, are used as input to the bias-free \mathcal{E} -net in RelaxNet. As a result, the \mathcal{E} -net produces $\boldsymbol{\alpha} \rightarrow \mathbf{0}$ and $\mathcal{M}^+ \rightarrow 1$, and thus we have

$$\mathcal{E} = \mathcal{M} \mathcal{M}^+ \sim \mathcal{M} \quad (\text{as } f \rightarrow \mathcal{M}). \quad (53)$$

In addition, given $\langle \boldsymbol{\psi} (\mathcal{M} - f) \rangle = \mathbf{0}$, the residuals in Eq. (49) converge to zero,

$$\varepsilon \rightarrow 0 \quad (\text{as } f \rightarrow \mathcal{M}), \quad (54)$$

and we recover the exact Euler equations independent of θ , i.e.,

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho \mathbf{V} \\ \rho E \end{pmatrix} + \nabla_{\mathbf{x}} \cdot \begin{pmatrix} \rho \mathbf{V} \\ \rho \mathbf{V} \mathbf{V} \\ \rho E \mathbf{V} \end{pmatrix} = -\nabla_{\mathbf{x}} \cdot \begin{pmatrix} 0 \\ p \mathbf{I} \\ p \mathbf{V} \end{pmatrix}, \quad (55)$$

given the following relationship as $f = \mathcal{M}$,

$$\mathbf{P} = p \mathbf{I}, \quad \mathbf{q} = \mathbf{0}. \quad \square \quad (56)$$

4. Data sampling

As presented in Section 3.3, a dataset with ground-truth distribution functions f_{ref} and collision terms Q_{ref} is needed in Eq. (19) to perform the supervised learning task. We consider the space of f as a sample space under the constraint,

$$f(t, \mathbf{x}, \mathbf{v}) \in B = \{f : 0 \leq f < \infty\}, \quad (57)$$

where B can be referred as the physically realizable set. A data-distribution p_f can be incorporated to generate f from B .

A common strategy for data-driven modeling is to first perform deterministic simulations and extract solutions in the intermediate steps as sample data. One disadvantage of this approach is that p_f is implicitly determined by the chosen test cases and can thus be heavily biased. It may not be possible to cover enough different particle distributions and flow regimes. The high expense of CFD simulations makes it challenging to establish an all-round database.

In this paper, we provide an alternative sampling strategy to generate particle distribution functions and corresponding collision terms. The theoretical basis of this approach is the closure hierarchies of moment system of the Boltzmann equation [35]. In the following we briefly explain the principle and implementation of the approach.

4.1. Generation of distribution function

A closure strategy of the Boltzmann equation aims to reconstruct the particle distribution function f from moments

$$\mathbf{m} = \langle \phi f \rangle, \quad (58)$$

under the constraint

$$f(t, \mathbf{x}, \mathbf{v}) \in B_M = \{f : f \geq 0, \|\langle \phi f \rangle\| < \infty\}, \quad (59)$$

where $\phi(\mathbf{v}) \in \mathbb{R}^{N_m}$ is a set of basis functions. Here we choose the basis in such a way that the first moments in \mathbf{m} are consistent with the conservative variables $\mathbf{W} = (\rho, \rho\mathbf{V}, \rho E)^T$. Therefore, we can write ϕ as

$$\phi(\mathbf{v}) = (\psi, \tilde{\phi}(\mathbf{v}))^T, \quad (60)$$

where $\psi = (1, \mathbf{v}, \mathbf{v}^2/2)^T$, and $\tilde{\phi}(\mathbf{v})$ are chosen as monomials and mixed polynomials of \mathbf{v} up to degree N_m . In the current work, we choose $N_m = 4$, thus \mathbf{m} resembles Levermore's 14 moment system [36], i.e.

$$\mathbf{m}(\mathbf{v}) = \left(1, \mathbf{v}, |\mathbf{v}|^2/2, \mathbf{v} \otimes \mathbf{v} - |\mathbf{v}|^2\mathbf{I}, |\mathbf{v}|^2\mathbf{v}, |\mathbf{v}|^4\right)^T. \quad (61)$$

The entropy closure seeks for a unique map $\mathbf{m} \mapsto f$ by forming an optimization problem. The objective function of the optimization problem is built with the help of a convex entropy function. For classical Maxwell-Statistics statistics, the minimal entropy closure problem writes

$$\min_{g \in B_m} (g \log(g) - g), \quad \text{s.t. } \mathbf{m} = \langle \phi g \rangle. \quad (62)$$

The solution of the above optimization problem can be expressed as

$$f_u(\mathbf{v}) = \exp(\boldsymbol{\gamma}_u \cdot \phi), \quad (63)$$

where $\boldsymbol{\gamma}_u \in \mathbb{R}^{N_m}$ is a vector of Lagrange multipliers of the dual formulation of the optimization problem, which reads

$$\boldsymbol{\gamma}_u = \operatorname{argmin}_{\boldsymbol{\gamma} \in \mathbb{R}^{N_m}} \{ \langle \exp(\boldsymbol{\gamma} \cdot \phi) \rangle - \boldsymbol{\gamma} \cdot \mathbf{m} \}. \quad (64)$$

The idea is to sample particle distribution functions from Eq. (63). Note that the condition number of the minimal entropy closure at a moment \mathbf{u} can be computed via the positive semi-definite Hessian of the dual problem, i.e.,

$$\mathbf{H}_u = \langle \phi \phi \exp(\boldsymbol{\gamma}_u \cdot \phi) \rangle, \quad (65)$$

where $\phi \phi$ denotes the union of ϕ . As analyzed in [37,38], generally the lower the condition number, the less the reconstructed particle distribution differs from the Maxwellian. Therefore, by controlling the condition number of \mathbf{H}_u , we can generate distribution functions that are close and far from equilibrium. We let the first three elements of $\boldsymbol{\gamma}$ be consistent with $\hat{\boldsymbol{\alpha}}$ in Eq. (36), and sample γ_n for $n > 3$ with a prescribed probability density. In this work, we use the condition number threshold as $c = 10^{-3}$ for the basis in Eq. (61).

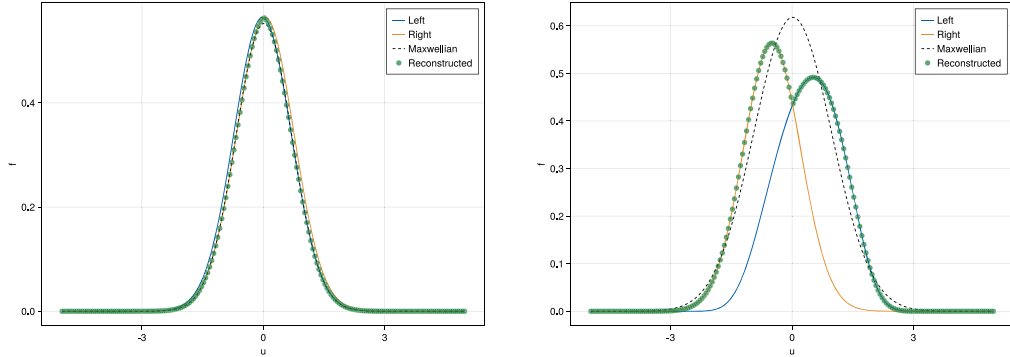
It is worthwhile to mention that the condition number cannot be arbitrarily large due to the constraint of realizability. The realizable set of the minimal entropy problem in Eq. (62) writes

$$\mathcal{R} = \{ \mathbf{m} : \mathbf{m} = \langle \phi g \rangle, g \in B_m \}. \quad (66)$$

It is known that the minimal entropy problem near the boundary of \mathcal{R} , denoted as $\partial\mathcal{R}$, may become ill-conditioned [39]. Therefore, we keep the condition number within a certain range, and employ an upwind reconstruction to enable the capability to generate highly non-equilibrium distribution functions that may exist in highly dissipative flows. The distribution functions produced from Eq. (63) are divided randomly into two parts, and the referenced f is reconstructed as

$$f_{\text{ref}} = f_L \mathcal{H}(\mathbf{n} \cdot \mathbf{v}) + f_R (1 - \mathcal{H}(\mathbf{n} \cdot \mathbf{v})), \quad (67)$$

where $\{f_L, f_R\}$ are the distribution functions from these two parts, \mathbf{n} is a randomly generated unit vector, and $\mathcal{H}(x)$ is the heaviside step function. Fig. 3 shows typical near-equilibrium and non-equilibrium particle distribution functions reconstructed from Eq. (67) using the degree $N_m = 4$. As the polynomial degree increases, the non-equilibrium effect is expected to be stronger. The detailed sampling strategy is summarized in Algorithm 1. Note that the sampling strategy here can work together with classical simulation-based data samplers, where the data produced by a specific simulation can be employed as reinforcement data on top of the pre-generated data. We will explain this further in Section 5.



(a) Near-equilibrium reconstructed distribution function (b) Non-equilibrium reconstruction distribution function

Fig. 3. Typical one-dimensional near-equilibrium and non-equilibrium particle distribution functions generated by the sampling strategy in Section 4.1 with the polynomial degree $N_m = 4$. The values of velocity u and distribution function f are normalized by $(2RT_0)^{1/2}$ and $\rho_0/(2RT_0)^{1/2}$, respectively.

Algorithm 1: Sampling of reference particle distribution functions.

Input: Number of dataset entries N_i , maximum moment order N_m , and spatial dimension D
 Basis functions ϕ
 Range of density, velocity, and temperature to formulate $\hat{\alpha}$
 Standard deviation σ to sample γ
 Condition number threshold c

Result: Dataset of reference distribution functions f

```

for  $i = 1, \dots, 2N_i$  do
    Sample  $\rho \in [\rho_{min}, \rho_{max}]$ 
    Sample  $\mathbf{V} \in [\mathbf{V}_{min}, \mathbf{V}_{max}]$ 
    Sample  $T \in [T_{min}, T_{max}]$ 
    Compute mean for  $(\gamma_{u,1}, \gamma_{u,2}, \gamma_{u,3})^T = \hat{\alpha}$ 
    do
        Sample  $\gamma_{u,n}, n = 4, \dots, N_m$ 
        Compute  $\mathbf{H}_u(\gamma_u)$ 
    while  $\sigma_{\mathbf{H}_u(\gamma_u)} < c$ 
    Compute  $f_{ent,i} = \exp(\gamma_u \cdot \phi)$ 
end
Divide  $f_{ent}$  into  $f_L$  and  $f_R$  equally
for  $i = 1, \dots, N_i$  do
    Sample unit vector  $\mathbf{n}$ 
    Reconstruct  $f_{ref,i} = f_{L,i} \mathcal{H}(\mathbf{n} \cdot \mathbf{v}) + f_{R,i} (1 - \mathcal{H}(\mathbf{n} \cdot \mathbf{v}))$ 
end
    
```

4.2. Computation of collision term

After the particle distribution functions are collected, the collision terms need to be computed and used in the cost function Eq. (19). In this paper, we consider three different collision terms for the Boltzmann and its model equations, which are briefly introduced below.

(1) *Full Boltzmann collision integral with fast spectral method:* The full Boltzmann term can be transformed into Carleman-type [40],

$$\begin{aligned}
 \mathcal{Q}(f, f) &= \int_{\mathbb{R}^3} \int_{\mathbb{S}^2} \Theta \mathbf{g} \left[f \left(\mathbf{v} + \frac{\mathbf{g}\Omega - \mathbf{g}}{2} \right) f \left(\mathbf{v}_* - \frac{\mathbf{g}\Omega - \mathbf{g}}{2} \right) - f(\mathbf{v}) f(\mathbf{v}_*) \right] d\Omega d\mathbf{v}_* \\
 &= 2 \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \Theta \delta(2\mathbf{y} \cdot \mathbf{g} + \mathbf{y}^2) \left[f \left(\mathbf{v} + \frac{\mathbf{y}}{2} \right) f \left(\mathbf{v}_1 - \frac{\mathbf{y}}{2} \right) - f(\mathbf{v}) f(\mathbf{v}_*) \right] d\mathbf{y} d\mathbf{v}_* \\
 &= 4 \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \Theta \delta(\mathbf{y} \cdot \mathbf{z}) [f(\mathbf{v} + \mathbf{y}) f(\mathbf{v} + \mathbf{z}) - f(\mathbf{v}) f(\mathbf{v} + \mathbf{y} + \mathbf{z})] d\mathbf{y} d\mathbf{z},
 \end{aligned} \tag{68}$$

which can be solved with a discrete Fourier transform-based spectral method with a computational cost of $O(N^3 \log N)$ [41]. Specifically, the particle distribution function and collision term discretized with N_v quadrature points are expanded into the Fourier series,

$$\begin{aligned}
 f(t, \mathbf{x}, \mathbf{v}) &= \sum_{k=-N_v/2}^{N_v/2-1} f_k(t, \mathbf{x}) \exp(i\xi_k \cdot \mathbf{v}), \\
 f_k &= \frac{1}{L^D} \langle f(t, \mathbf{x}, \mathbf{v}) \exp(-i\xi_k \cdot \mathbf{v}) \rangle, \\
 Q(t, \mathbf{x}, \mathbf{v}) &= \sum_{k=-N_v/2}^{N_v/2-1} Q_k(t, \mathbf{x}) \exp(i\xi_k \cdot \mathbf{v}), \\
 Q_k &= \sum_{l,m=-N_v/2, (l+m=k)}^{N_v/2-1} f_l f_m [\beta(l, m) - \beta(m, m)],
 \end{aligned} \tag{69}$$

where L is the span of velocity space in each dimension, and β is the kernel mode, and the convolutions defined in Eq. (68) can be solved in the frequency domain. The details of this method can be found in [41], and its numerical implementation is available in [1].

- (2) *Shakhov relaxation model*: The Shakhov relaxation model builds a heat flux-based correction term on top of the BGK model to ensure that the Chapman-Enskog expansion of the model has the correct Prandtl number [18]. The equilibrium state in the Shakhov model writes

$$\mathcal{E} = \mathcal{M} \left[1 + (1 - \text{Pr}) \mathbf{c} \cdot \mathbf{q} \left(\frac{\mathbf{c}^2}{RT} - 5 \right) / (5pRT) \right], \tag{70}$$

where Pr is the Prandtl number, \mathbf{c} is the peculiar velocity, \mathbf{q} is the heat flux, and $R = k/m$ is the gas constant. It has been shown that the Shakhov model is more accurate in predicting highly non-equilibrium flows in normal shock wave compared to the BGK model [42]. With the inclusion of heat flux, the Shakhov model loses some of the structural properties shown in Section 2. For example, non-negative solution of particle distribution function can appear, and the H-theorem is no longer strictly valid in the Shakhov model. Using the Shakhov model to construct the dataset is to validate the ability of RelaxNet to correctly approximate \mathcal{E} .

- (3) *ν -BGK model*: Another strategy to improve the BGK model is to introduce velocity-dependent relaxation frequency. Here we choose the ν -BGK model [19], where the relaxation term writes

$$Q = \nu(\mathbf{v})(\mathcal{M} - f). \tag{71}$$

Different curves of ν as a function of \mathbf{v} can be fitted to provide the correct Prandtl number of monatomic gas. Here we adopt the relaxation frequency defined in [43], which is an improvement over the frequencies provided in [19] and can provide more accurate solution in non-equilibrium flows. In this model, the relaxation frequency is defined as

$$\begin{aligned}
 \nu(|\mathbf{v}|) &= A \frac{p}{\mu} \left[\nu_{eq}^0(\xi) + 2\nu_{eq}^0(\xi) \right], \\
 \nu_{eq}^0(\xi) &= \frac{3}{2} \left[\exp(-\xi^2) + \frac{\sqrt{\pi}}{2} \left(\frac{1}{\xi} + 2\xi \right) \text{erf}(\xi) \right],
 \end{aligned} \tag{72}$$

where $\xi = |\mathbf{c}|/\sqrt{2kT/m}$, and A is a parameter depending on the viscosity index. Using the ν -BGK model to construct the dataset is to validate the ability of RelaxNet to correctly approximate velocity-dependent ν .

5. Solution algorithm

5.1. Update algorithm

The finite volume method is employed to build the solution algorithm of the RelaxNet-based universal Boltzmann equation. We denote the mean value of particle distribution function, in a control volume as

$$f(t^n, \mathbf{x}_i, \mathbf{v}_j) = f_{i,j}^n = \frac{1}{\Omega_i(\mathbf{x})\Omega_j(\mathbf{v})} \int_{\Omega_i} \int_{\Omega_j} f(t^n, \mathbf{x}, \mathbf{v}) d\mathbf{x}d\mathbf{v}, \tag{73}$$

where Ω_i and Ω_j are cell areas in the discrete physical and velocity space. The update algorithm of distribution function in each control volume can be written as

$$f_{i,j}^{n+1} = f_{i,j}^n + \frac{1}{\Omega_i} \int_{t^n}^{t^{n+1}} \sum_{r=1}^{n_f} F_r \Delta S_r dt + \int_{t^n}^{t^{n+1}} \text{NN}_\theta(f_i, \bar{\tau}_i)_j dt, \quad (74)$$

where F_r is the time-dependent flux function of distribution function at r -th cell interface, ΔS_r is the interface area, and n_f is the number of interfaces. Details of the update algorithm can be found in [44].

5.2. Numerical flux

The numerical flux in Eq. (74) is evaluated from the reconstructed particle distribution function at the cell interface. At the cell interface $\mathbf{x}_{i+1/2}$, the distribution function is constructed in an upwind fashion,

$$f_{i+1/2,j} = f_{i+1/2,j}^L \mathcal{H}(\mathbf{n}_{i+1/2} \cdot \mathbf{v}_j) + f_{i+1/2,j}^R (1 - \mathcal{H}(\mathbf{n}_{i+1/2} \cdot \mathbf{v}_j)), \quad (75)$$

where $\mathbf{n}_{i+1/2}$ is the unit normal vector of the interface, and $\mathcal{H}(x)$ is the heaviside step function. The left and right value of distribution function $f_{i+1/2,j}^{L,R}$ can be obtained through reconstruction and interpolation. For example, a second-order interpolation results in

$$\begin{aligned} f_{i+1/2,j}^L &= f_{i,j} + \nabla_{\mathbf{x}} f_{i,j} \cdot (\mathbf{x}_{i+1/2} - \mathbf{x}_i), \\ f_{i+1/2,j}^R &= f_{i+1,j} + \nabla_{\mathbf{x}} f_{i+1,j} \cdot (\mathbf{x}_{i+1/2} - \mathbf{x}_{i+1}), \end{aligned} \quad (76)$$

where $\nabla_{\mathbf{x}} f$ is the reconstructed gradient with slope limiters. The principle of higher-order interpolation can be found in [45]. The numerical flux of particle distribution function can then be evaluated as

$$F_{i+1/2,j} = f_{i+1/2,j} \mathbf{n}_{i+1/2} \cdot \mathbf{v}_j. \quad (77)$$

5.3. Collision term

The RelaxNet-based collision operator in each cell reads

$$\mathcal{Q}_{i,j} = \text{NN}_\theta(f_i, \bar{\tau}_i)_j. \quad (78)$$

According to the kinetic theory of gases, the mean relaxation time is defined as $\bar{\tau} = \mu/p$ where p is pressure. In this paper, we adopt the variational hard-sphere (VHS) molecule model to compute the viscosity coefficient, i.e.,

$$\mu = \mu_{\text{ref}} \left(\frac{T}{T_{\text{ref}}} \right)^\omega, \quad (79)$$

where μ_{ref} and T_{ref} are the viscosity and temperature in the reference state, and ω is the viscosity index.

Once the parameters in RelaxNet get optimized, the collision term can be integrated with the help of time-integral algorithms. While the forward Euler method is the most straightforward integrator for solving Eq. (74), it is feasible to adopt other higher-order methods, e.g., Tsitouras' 5/4 Runge-Kutta method [46]. When the time step is much larger than mean relaxation time, it is preferable to employ implicit [47] and implicit-explicit (IMEX) methods [48] to improve numerical stability of the scheme.

Note that the update procedure in Eq. (74) produces case-specific distribution functions that may be beyond the dataset used for training and testing. The newly generated distribution functions can be supplemented to the dataset on-the-fly by computing the collision term $\mathcal{Q}_{i,j} = \mathcal{Q}(f_{i,j})$ with the methods shown in Section 4.2. We use the rise of residuals as a criterion to generate supplementary data. A schematic diagram of the solution algorithm of the RelaxNet-based UBE is presented in Fig. 4.

6. Numerical experiments

In this section, we present numerical experiments to validate the RelaxNet-based universal Boltzmann equation and its solution algorithm. The dimensionless variables are introduced in the simulation, i.e.,

$$\begin{aligned} \tilde{t} &= \frac{t}{L_0/(2RT_0)^{1/2}}, \quad \tilde{\mathbf{x}} = \frac{\mathbf{x}}{L_0}, \quad \tilde{\rho} = \frac{\rho}{\rho_0}, \quad \tilde{\mathbf{v}} = \frac{\mathbf{v}}{(2RT_0)^{1/2}}, \quad \tilde{\mathbf{V}} = \frac{\mathbf{V}}{(2RT_0)^{1/2}}, \quad \tilde{T} = \frac{T}{T_0}, \\ \tilde{E} &= \frac{E}{2RT_0}, \quad \tilde{\mathbf{P}} = \frac{\mathbf{P}}{2\rho_0RT_0}, \quad \tilde{\mathbf{q}} = \frac{\mathbf{q}}{\rho_0(2RT_0)^{3/2}}, \quad \tilde{\mu} = \frac{\mu}{\rho_0L_0(2RT_0)^{1/2}}, \quad \tilde{f} = \frac{f}{\rho_0/(2RT_0)^{D/2}}. \end{aligned}$$

The subscript zero represents the reference state. For brevity, the tilde notation for dimensionless variables will be removed henceforth. In all simulations we employ monatomic hard-sphere gas, where the viscosity coefficient can be determined by Eq. (79).

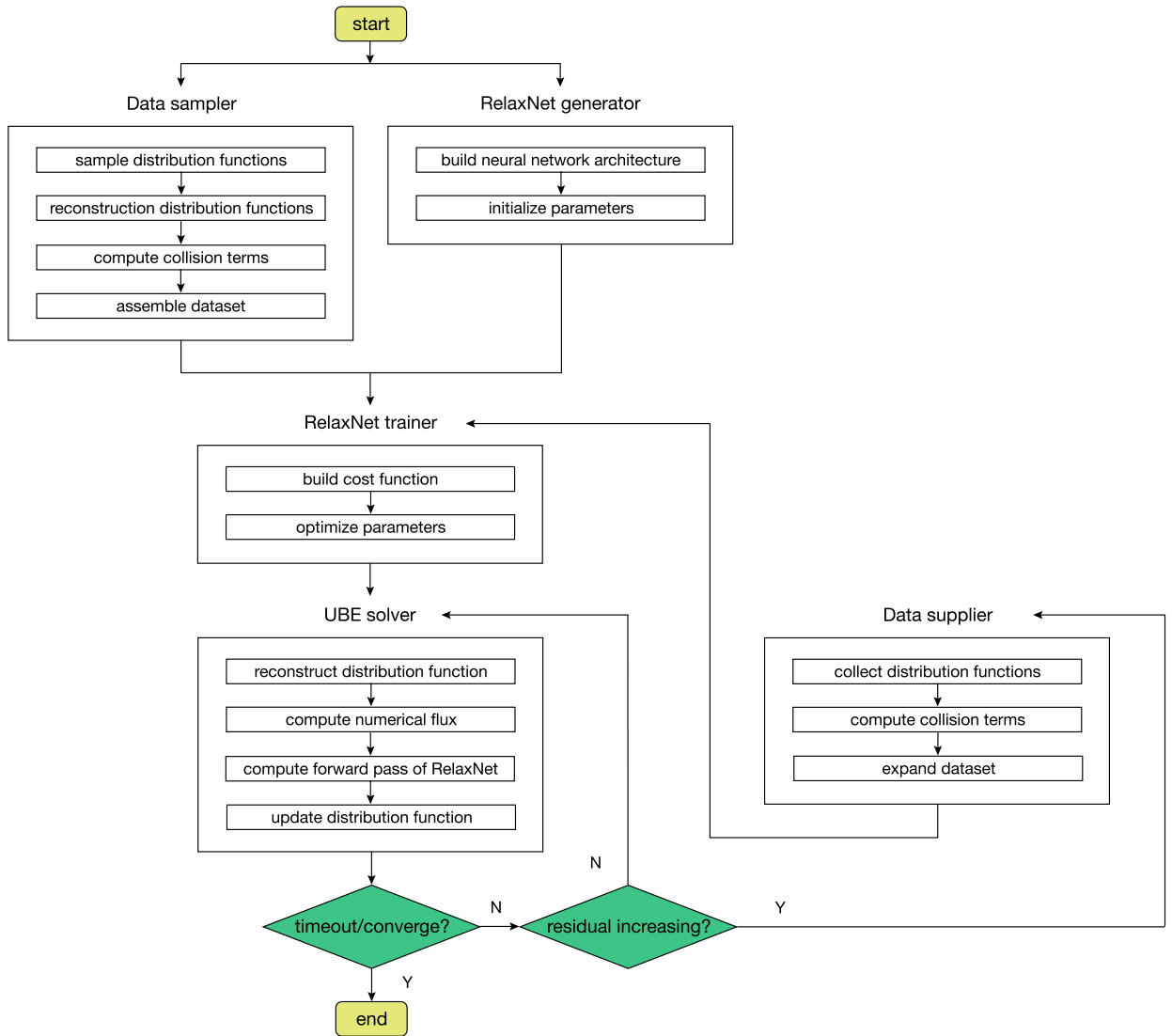


Fig. 4. Schematic diagram of solution algorithm of the RelaxNet-based universal Boltzmann equation.

Table 2
Computational setup for sampling particle distribution functions in the data generation problem.

N_i	N_m	D	ρ_s	U_s	T_s	γ^4	γ^5
3000	4	1	$\mathcal{U}(0, 2)$	$\mathcal{U}(-1.5, 1.5)$	$\mathcal{U}(0, 4)$	$\mathcal{N}(0, 0.005)$	$\mathcal{T}(0, 0.005, -\infty, 0)$

6.1. Data generation

To further explain the principle of the data sampling strategy described in Section 4 and illustrate its superiority, we first compare the distribution of the data generated by Algorithm 1 with the data collected directly from numerical simulation results of the Boltzmann equation. The computational setup for Algorithm 1 is presented in Table 2, where $\{\rho_s, U_s, T_s\}$ is the macroscopic variables used for sampling distribution functions, γ is the Lagrange multipliers in Eq. (62), and $\mathcal{U}(a, b)$, $\mathcal{N}(\mu, \sigma)$, and $\mathcal{T}(\mu, \sigma, a, b)$ denote uniform, normal, and truncated normal distributions, respectively.

Two problems are considered below as illustrative examples to produce simulation-based datasets.

- (1) Wave propagation problem. The initial particle distribution function is set to Maxwellian everywhere in correspondence with the following flow field,

Table 3
Computational setup in the wave propagation problem.

t	x	N_x	u	N_u	Quadrature	Kn	A_s
(0, 5]	[0, 1]	200	[-8, 8]	80	Rectangular	10^{-3}	$\mathcal{U}(0.5, 1.5)$
a_s	b_s	c_s	Reference	Integrator	Boundary	CFL	
$\mathcal{U}(0, 0.2)$	$\mathcal{U}(-1, 1)$	$\mathcal{U}(0.5, 1.5)$	Shakhov	RK4	Periodic	0.5	

Table 4
Computational setup in the Sod shock tube problem.

t	x	N_x	u	N_u	Quadrature	Kn	ρ_{L_s}
(0, 0.2]	[0, 1]	200	[-8, 8]	80	Rectangular	10^{-4}	$\mathcal{U}(0, 2)$
ρ_{R_s}	T_{L_s}	T_{R_s}	Reference	Integrator	Boundary	CFL	
$\mathcal{U}(0, 0.25)$	$\mathcal{U}(1, 3)$	$\mathcal{U}(0, 1.25)$	Shakhov	IMEX	Dirichlet	0.5	

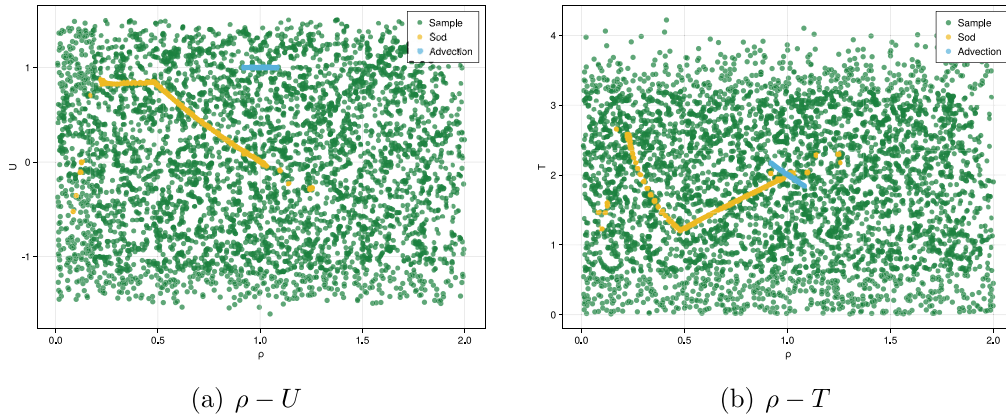


Fig. 5. Phase diagrams of the data generated by the algorithmic sampler and a single simulation (green points are the results generated by the algorithmic sampler, while yellow and blue points are generated by the simulation results from the wave propagation problem and Sod problem, respectively, Section 6.1). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\begin{pmatrix} \rho \\ U \\ p \end{pmatrix}_{t=0} = \begin{pmatrix} A(1 + a \sin(2\pi x)) \\ b \\ c \end{pmatrix}.$$

The detailed computational setup is presented in Table 3.

- (2) Sod shock tube problem. The initial particle distribution function is set to Maxwellian everywhere in correspondence with the one-dimensional Riemann problem,

$$\begin{pmatrix} \rho \\ U \\ p \end{pmatrix}_{t=0, x_L} = \begin{pmatrix} \rho_L \\ 0 \\ p_L \end{pmatrix}, \quad \begin{pmatrix} \rho \\ U \\ p \end{pmatrix}_{t=0, x_R} = \begin{pmatrix} \rho_R \\ 0 \\ p_R \end{pmatrix}.$$

The detailed computational setup is presented in Table 4.

We compare the datasets generated from the algorithmic sampler described in Section 4 and the above two numerical examples with the same sample size $N_i = 3000$. Two cases are considered. In the first case, the input parameters of two examples are fixed as $\{A, a, b, c\} = \{1, 0.1, 1, 1\}$ and $\{\rho_L, \rho_R, T_L, T_R\} = \{1, 0.125, 2, 0.625\}$ to analyze the dataset produced by a single numerical simulation. In the second case, we sample the parameters ten times according to the uniform distribution of the intervals given in Table 3 and 4 to characterize the datasets collected from multiple numerical simulations with various inputs.

Fig. 5 and 6 present the phase diagrams between macroscopic variables generated by the algorithmic sampler using Algorithm 1 and the simulation results. As shown in Fig. 5, the samples from a single numerical simulation can suffer from a strong bias. Such a phenomenon is caused by the physical constraints in a particular flow problem. In the wave propagation problem, the density and temperature are linearly coupled due to the isobaric pressure. In the Sod shock tube problem, the density, velocity and temperature are correlated around wave structures, i.e., the rarefaction wave, contact discontinuity, and shock wave. The discontinuous flow field in the Sod problem makes the samples show a even sparser distribution in the phase diagram. As particle distribution functions generated from multiple sampling of parameters and

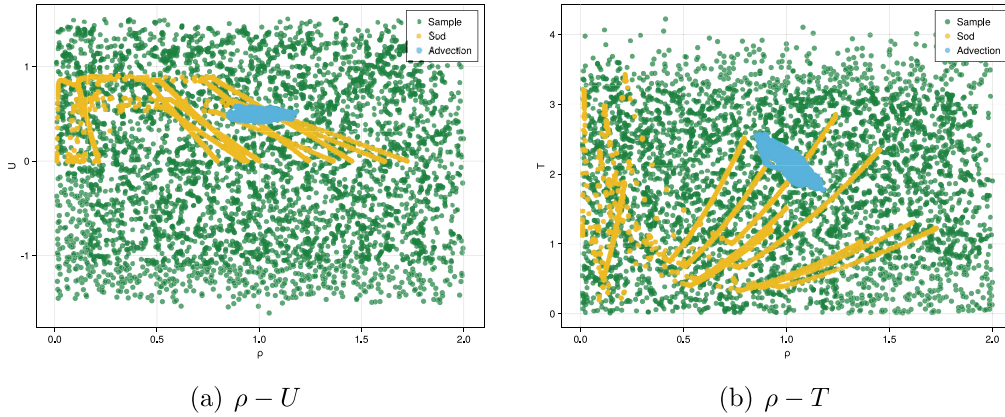


Fig. 6. Phase diagrams of the data generated by the algorithmic sampler and multiple simulations (green points are the results generated by the algorithmic sampler, while yellow and blue points are generated by the simulation results from the wave propagation problem and Sod problem, respectively, Section 6.1).

Table 5
Computational cost in the data generation problem.

	Time	Allocations	Memory
Wave	197.12 s	1.49×10^9	485.94 GB
Sod	41.46 s	4.18×10^6	13.15 GB
Sampler	0.03 s	7.27×10^3	13.33 MB

numerical simulations are added to the dataset, as presented in Fig. 6, a wider range of flow variables can be obtained. From this trend, it may be feasible to generate reliable datasets through numerical simulations if the sampling ranges of parameters are further extended, but this obviously requires the use of appropriate physical problems and adequate sample size. To accomplish the goal, the use of trial-and-error processes is inevitable.

In contrast, the algorithmic sampler generates a dataset of particle distribution function with a much wider range of densities, velocities and temperatures. The obtained samples are homogeneously distributed without a preferred direction. Therefore, the predefined domain of $\{\rho, U, T\}$ is fully covered with a relatively small sample size. Note that this also implies a broader and unbiased distribution of particle distribution functions, since there is a one-to-one correspondence between distribution functions and macroscopic quantities. Compared with the simulation-based sampling, the broader dataset provided by the algorithmic sampler reduces the risk of overfitting and helps improve the generalization of the neural network model. Table 5 provides the computation time and allocations for the three methods (including the garbage collection with Julia programming language [49]). Since the presented sampling strategy does not require a full simulation of the Boltzmann equation with multiple cases or initial conditions, the computational cost is significantly reduced by three to five orders of magnitude.

6.2. Homogeneous relaxation

In this numerical experiment, we study the homogeneous relaxation problem of non-equilibrium particle distribution functions. To fully validate the ability of RelaxNet to approximate the solution of the Boltzmann and its model equations, three collision models, i.e., the Shakhov model, the velocity-dependent ν -BGK model, and the full Boltzmann integral model described in Section 4.2, are employed to provide ground-truth datasets for training the neural network.

6.2.1. Shakhov model

First we consider the non-equilibrium distribution function in the initial state,

$$f(t=0, u) = \frac{1}{2\sqrt{\pi}} \left(\exp\left(-\frac{(u-1.5)^2}{2}\right) + \frac{7}{10} \exp\left(-\frac{(u+1.5)^2}{2}\right) \right). \quad (80)$$

We construct RelaxNet in such a way that there are two hidden layers of the same dimension as the input layer in both \mathcal{E} -net and τ -net. A training set with the same sample size $N_i = 10000$ is generated by Algorithm 1 to train RelaxNet, which is then used to solve the universal Boltzmann equation. The test set with 1000 samples is generated independently by the same Algorithm. The setups of the neural network, training and computation are provided in Table 6, where $N_h = 2$ is the number of hidden layers and Tsit5 denotes Tsitouras' 5/4 Runge-Kutta method [46].

The values of cost function in Eq. (19) for both training and test datasets during the training process are plotted in Fig. 7. After the parameters of RelaxNet are optimized, the initial value problem of the UBE model is solved. Note that the

Table 6
Computational setup for the homogeneous relaxation problem based on the Shakhov model.

D	u	N_u	N_i	ρ_s	U_s
1	$[-8, 8]$	80	10000	$\mathcal{U}(0.1, 10)$	$\mathcal{U}(-1, 1)$
V_s	W_s	T_s	γ_4	γ_5	Kn_s
0	0	$\mathcal{U}(0.1, 5)$	$\mathcal{N}(0, 0.005)$	$\mathcal{T}(0, 0.005, -\infty, 0)$	$\mathcal{U}(0.001, 1)$
N_h	N_m	Reference	Optimizer	t	Kn
2	4	Boltzmann	Adam	$(0, 8]$	1
Integrator	Quadrature	CFL			
Tsit5	Rectangular	0.3			

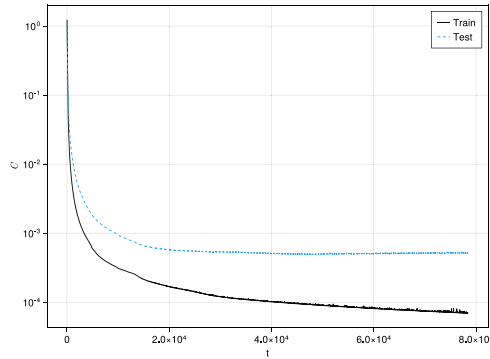


Fig. 7. Values of cost function versus optimization time in the training and test datasets for the homogeneous relaxation problem (Section 6.2.1).

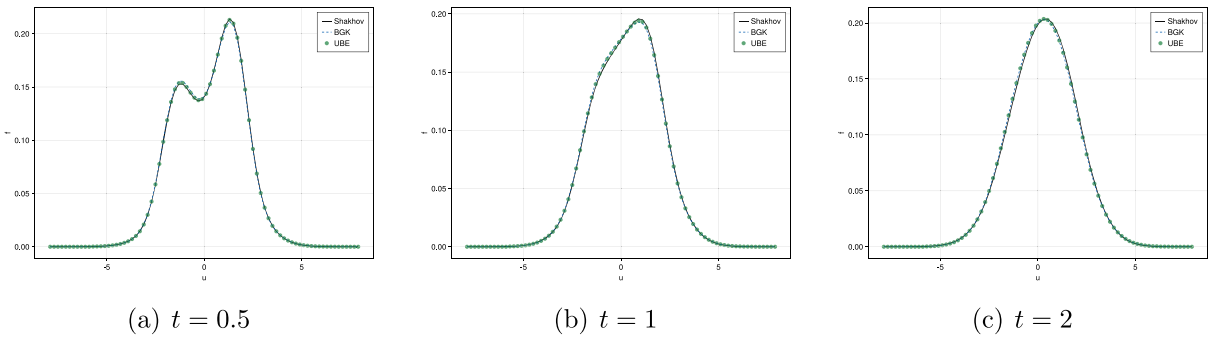


Fig. 8. Particle distribution functions at different time instants in the homogeneous relaxation problem (Shakhov model is used as reference solution, Section 6.2.1).

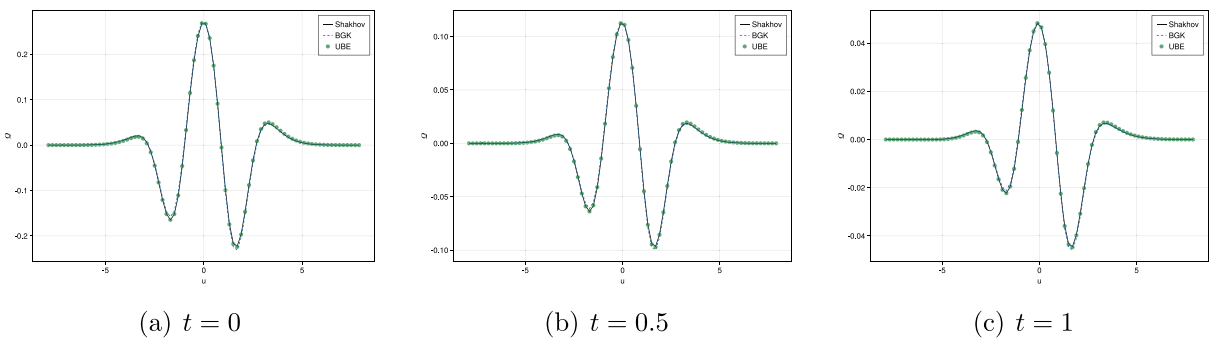


Fig. 9. Collision terms at different time instants in the homogeneous relaxation problem (Shakhov model as reference solution, Section 6.2.1).

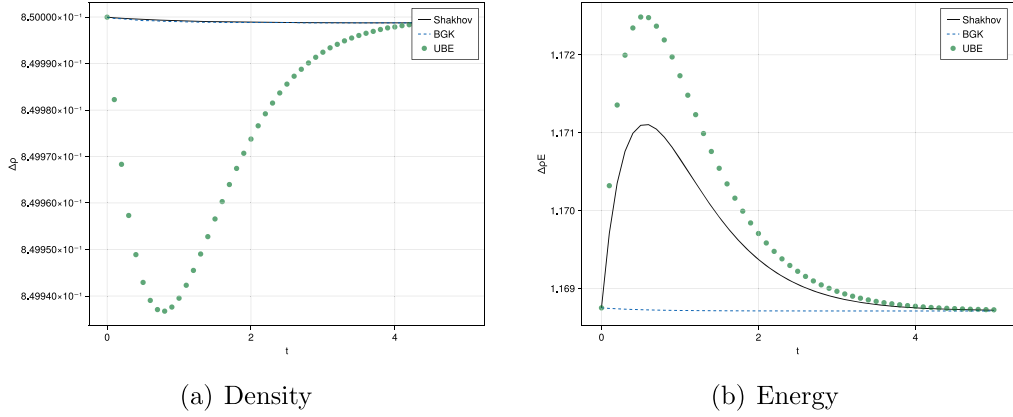


Fig. 10. Evolution of total density and energy with time in the homogeneous relaxation problem (Shakhov model as reference solution, Section 6.2.1).

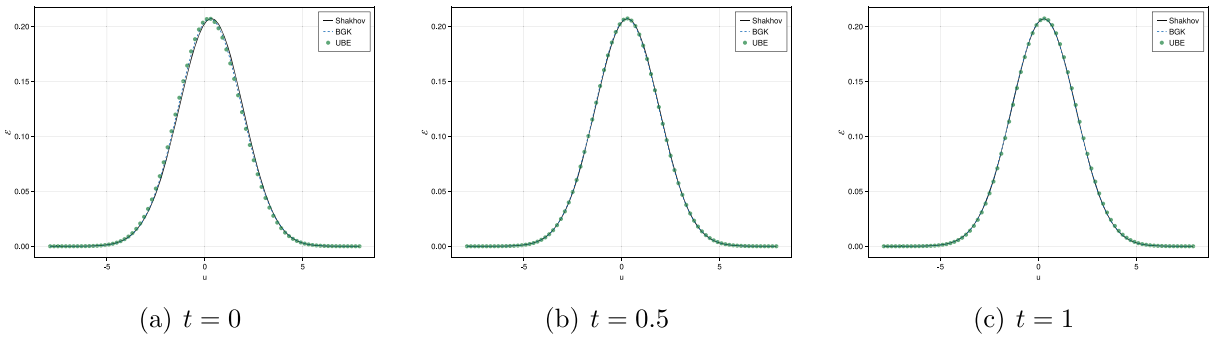


Fig. 11. Equilibrium functions at different time instants in the homogeneous relaxation problem (Shakhov model as reference solution, Section 6.2.1).

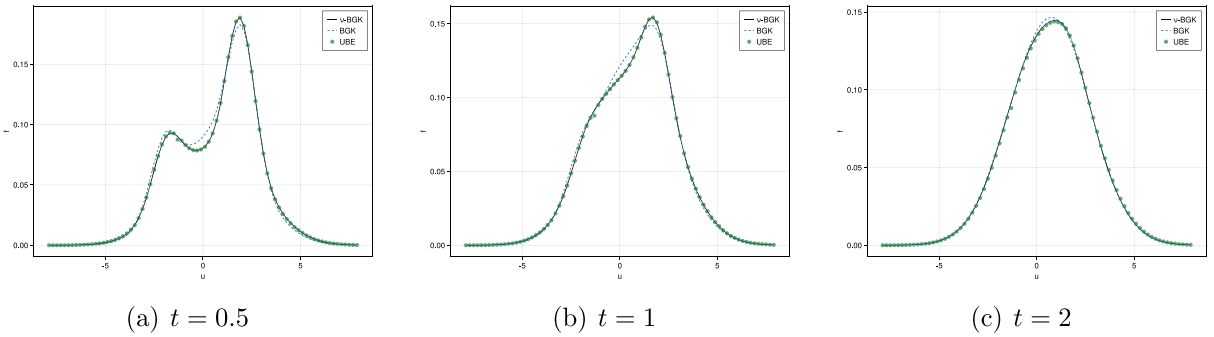


Fig. 12. Particle distribution functions at different time instants in the homogeneous relaxation problem (ν -BGK model as reference solution, Section 6.2.2).

bimodal distribution in Eq. (80) can fall outside the training set, so this example can act as a validation set. Fig. 8 presents the solutions of particle distribution functions at different time instants computed by the BGK, Shakhov, and UBE model. It can be seen that the UBE provide the solution consistent with the Shakhov model. This is attributed to the accurate approximation of the collision term of the Boltzmann equation, which is shown in Fig. 9. Fig. 10 presents the evolution of macroscopic density and energy with time. It is clear that the conservation of mass is approximately satisfied (the mass deviate only by around 10^{-4}), which provides a corroboration of Theorem 2. The error in energy slightly exceeds the range of the Shakhov reference solution. Due to the interpretability of RelaxNet, we can open the black box of forward pass in the UBE model and explain the issue. Fig. 11 presents the equilibrium state \mathcal{E} approximated by RelaxNet at different time instants, which intuitively explains the mechanism of UBE. Note that the Shakhov model does not satisfy H -theorem and the logarithm of equilibrium distribution corrected by heat flux cannot be written as a sum of collision invariants. As a result, the approximation of RelaxNet deviates a bit from the reference Shakhov solution.

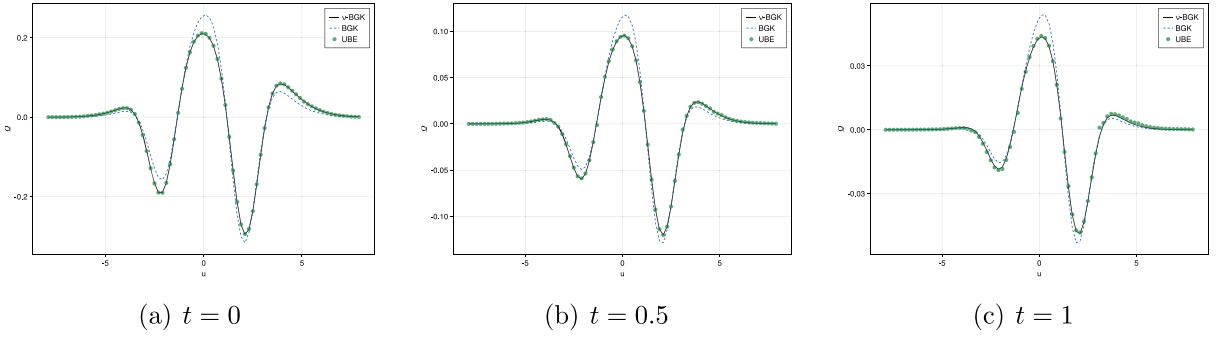


Fig. 13. Collision terms at different time instants in the homogeneous relaxation problem (ν -BGK model as reference solution, Section 6.2.2).

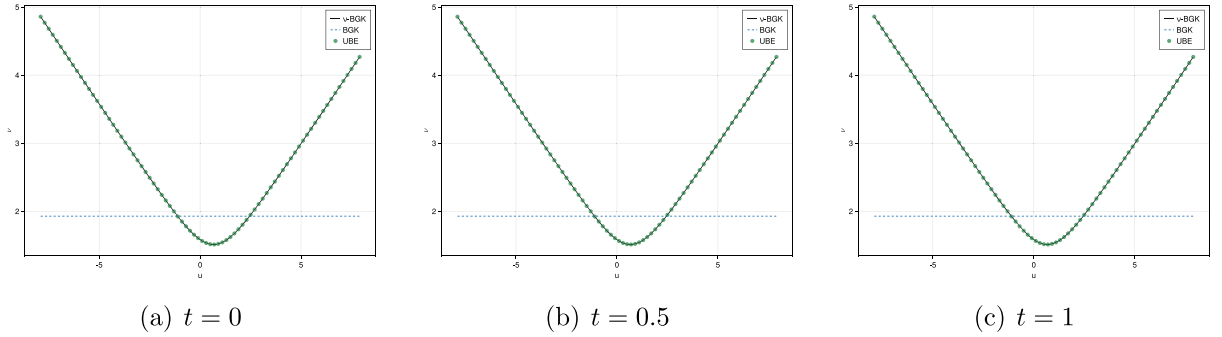


Fig. 14. Relaxation frequencies at different time instants in the homogeneous relaxation problem (ν -BGK model as reference solution, Section 6.2.2).

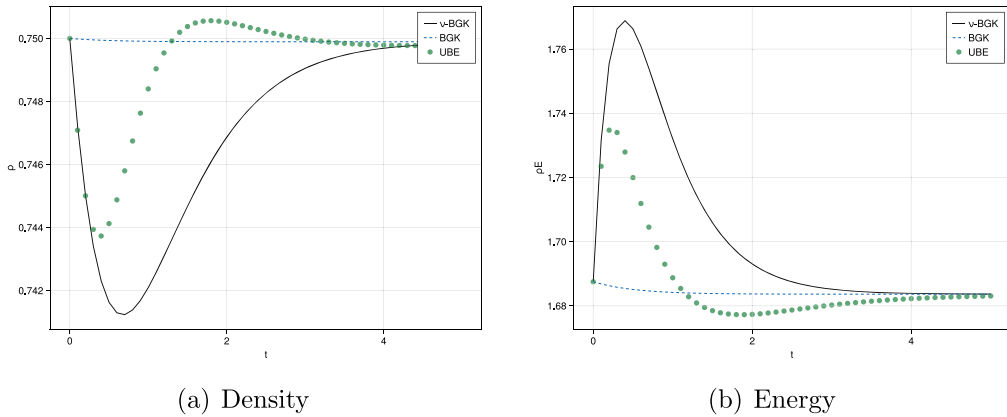


Fig. 15. Evolution of total density and energy with time in the homogeneous relaxation problem (ν -BGK model as reference solution, Section 6.2.2).

6.2.2. ν -BGK model

Next we consider the same setup as in Section 6.2.1, except that the velocity-dependent ν -BGK model shown in Section 4.2 is employed to produce reference solution. Fig. 12 presents the solutions of particle distribution functions at different time instants computed by the BGK, the ν -BGK, and the UBE model. It is clear that there is a more significant difference between the BGK and ν -BGK solutions, while the solution provided by the UBE model is in excellent agreement with the reference ν -BGK solution. As we explained \mathcal{E} in Section 6.2.1, RelaxNet allows us to quantitatively interpret the relaxation frequency ν in the UBE model. Fig. 13 and 14 show the collision terms and relaxation frequencies used in RelaxNet, and compare them with the BGK and ν -BGK models. It is clear that the dependence between relaxation frequencies and particle velocities are precisely recovered, bringing a more accurate approximation of the collision term. Fig. 15 presents the evolution of macroscopic density and energy with time, which corroborates Theorem 2. It can be seen that although the parameters in RelaxNet are optimized based on the solution of ν -BGK model in a supervised learning manner, the physics-informed regularization in the cost function in Eq. (19) allows UBE to better preserve the conservation property, which provides a corroboration of Theorem 2.

Table 7
Computational setup for the homogeneous relaxation problem based on the full Boltzmann model.

D	\mathbf{v}	N_u	N_v	N_w	N_i
3	$[-8, 8]^3$	80	28	28	10000
ρ_s	U_s	V_s	W_s	T_s	γ_4
$\mathcal{U}(0.1, 10)$	$\mathcal{U}(-1, 1)$	0	0	$\mathcal{U}(0.1, 5)$	$\mathcal{N}(0, 0.005)$
γ_s	Kn_s	N_h	N_m	Reference	Optimizer
$\mathcal{T}(0, 0.005, -\infty, 0)$	$\mathcal{U}(0.001, 1)$	2	4	Boltzmann	Adam
t	Kn	Integrator	Quadrature	CFL	
$(0, 8]$	1	Tsit5	Rectangular	0.3	

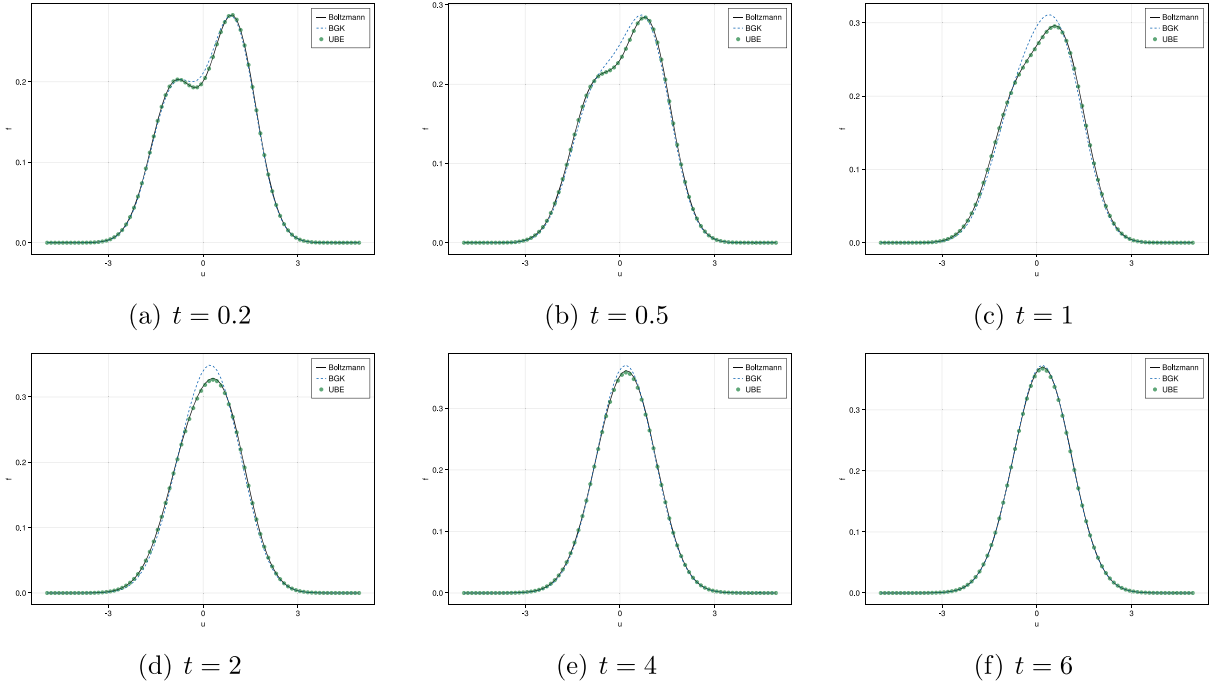


Fig. 16. Particle distribution functions at different time instants in the homogeneous relaxation problem (full Boltzmann model as reference solution, Section 6.2.3).

6.2.3. Full Boltzmann model

We then turn to the full Boltzmann equation. In this case, the initial particle distribution function is set as

$$f(t = 0, u, v, w) = \frac{1}{\sqrt{\pi}} \left(\exp(-(u - 1)^2) + \frac{7}{10} \exp(-(u + 1)^2) \right) \exp(-v^2) \exp(-w^2).$$

RelaxNet is constructed in the same way as in Section 6.2.1 and 6.2.2. Following Algorithm 1, the fast spectral method introduced in Section 4.2 is employed to generate the dataset used for training RelaxNet. The computational setup is presented in Table 7.

Fig. 16 provides the evolution of particle distribution functions with time. The variable used for display is the reduced distribution function on u axis, i.e.,

$$h(t, u) = \int_{\mathbb{R}^2} f(t, u, v, w) dv dw.$$

As shown, UBE based on RelaxNet provides a solution consistent with the full Boltzmann equation, while the solution of the BGK model shows significant deviations. Fig. 17 presents the collision terms computed by the three approaches at different time instants. Fig. 18 shows the equilibrium distributions and relaxation frequencies approximated by RelaxNet, which opens the black box of neural network and quantitatively illustrates the mechanism that allows the relaxation model to recover full Boltzmann solution by choosing appropriate equilibrium distribution functions and velocity-dependent relaxation frequencies. It can be seen that the difference between the equilibrium distribution in RelaxNet and the Maxwellian

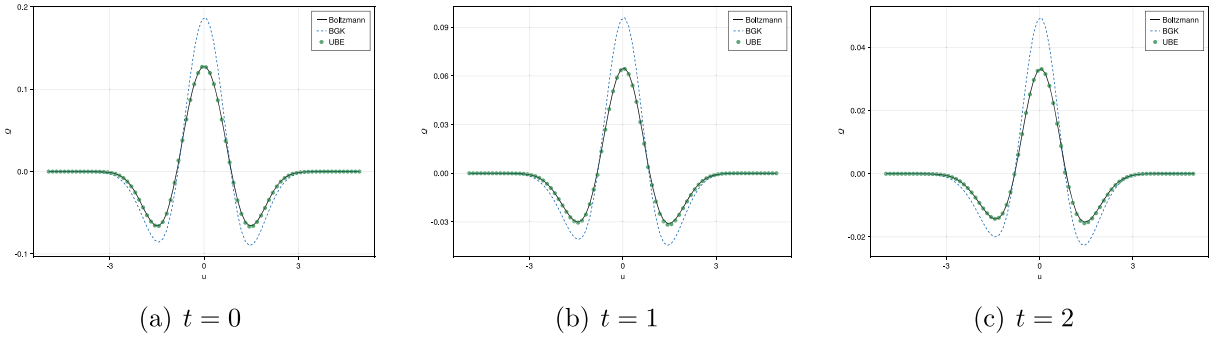


Fig. 17. Collision terms at different time instants in the homogeneous relaxation problem (full Boltzmann model as reference solution, Section 6.2.3).

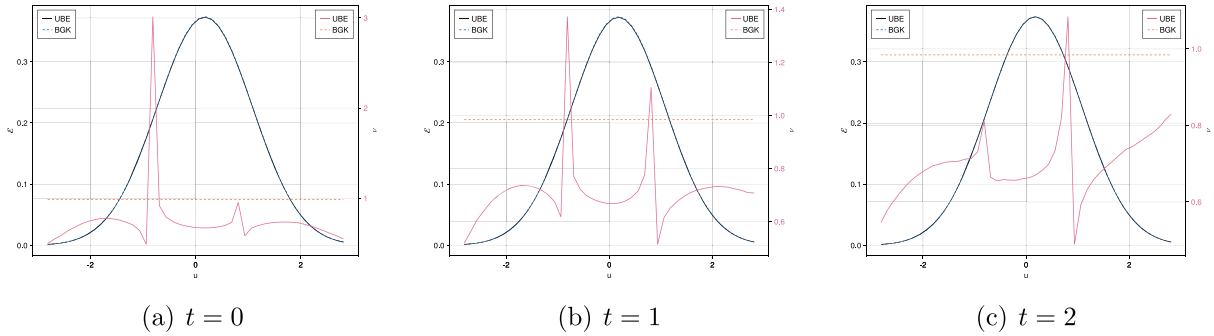


Fig. 18. Equilibrium (dark colors) and relaxation frequencies (light colors) at different time instants in the homogeneous relaxation problem (full Boltzmann model as reference solution, Section 6.2.3).

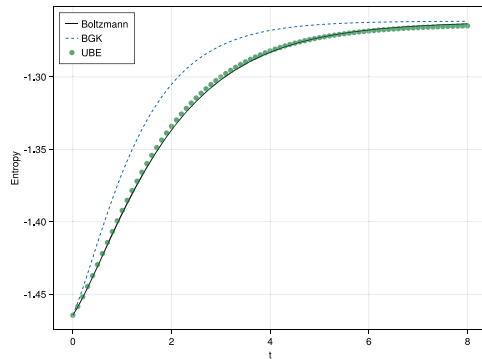


Fig. 19. Evolution of entropy with time in the homogeneous relaxation problem (full Boltzmann model as reference solution, Section 6.2.3).

distribution is slight, and thus the correction of the BGK solution is mainly achieved by the velocity-dependent relaxation frequency. Different from predefined frequencies in the ν -BGK model, e.g., in Eq. (72), RelaxNet provides a variable relaxation frequency that depends on the solution of local particle distribution function, and thus provides an approximation that better fits the real collision frequency in Eq. (2). Fig. 19 shows the evolution of entropy density defined as $\eta = \langle f \log(f) \rangle$, from which the principle of increase of entropy and the H-theorem demonstrated in Section 2 and Theorem 4 are validated.

Table 8 presents the computational costs of the three methods to simulate the time-series solution of the homogeneous relaxation problem. Since a large number of convolution operations for solving the Boltzmann collision integral are replaced by tensor multiplication, the computational cost of RelaxNet is significantly reduced compared. Under the current numerical setup, the simulation time is reduced to 1/50 of the fast spectral method for the full Boltzmann equation, and the memory usage is reduced to 1/17. This numerical experiment verifies the ability of the RelaxNet-based UBE to solve the Boltzmann equation accurately and efficiently.

Table 8
Computational cost of computing time-series solutions in the homogeneous relaxation problem.

	Time	Allocations	Memory
Boltzmann	2.167 s	90322	6.43 GB
BGK	1.076 ms	2939	3.38 MB
UBE	43.472 ms	14344	375.66 MB

Table 9
Computational setup in the wave propagation problem.

D	\mathbf{v}	N_u	N_v	N_w	N_i
3	$[-8, 8]^3$	80	28	28	10000
ρ_s	U_s	V_s	W_s	T_s	γ_4
$\mathcal{U}(0.5, 5)$	$\mathcal{U}(-1, 1)$	0	0	$\mathcal{U}(0.5, 2)$	$\mathcal{N}(0, 0.005)$
γ_s	Kn_s	N_h	N_m	Reference	Optimizer
$\mathcal{T}(0, 0.005, -\infty, 0)$	$\mathcal{U}(0.0001, 1)$	2	4	Boltzmann	Adam
t	x	N_x	Kn	ϖ	Integrator
(0, 1]	[0, 1]	[5, 40]	[0.001, 0.1]	5/3	Tsit5
Boundary	Quadrature	CFL			
Periodic	Rectangular	0.1			

Table 10
Errors and convergences of the solution algorithm in the wave propagation problem at $\text{Kn} = 0.001$.

Δx	L^1 error	Order	L^2 error	Order
0.2	5.13390E-2		1.57017E-2	
0.1	1.30266E-2	1.98	2.86593E-3	2.45
0.05	3.21382E-3	2.02	5.17958E-4	2.47
0.025	8.06801E-4	2.00	9.32237E-5	2.47

Table 11
Errors and convergences of the solution algorithm in the wave propagation problem at $\text{Kn} = 0.01$.

Δx	L^1 error	Order	L^2 error	Order
0.2	4.34367E-2		1.29830E-2	
0.1	1.10154E-2	1.98	2.31591E-3	2.49
0.05	2.65201E-3	2.06	4.07591E-4	2.50
0.025	6.38268E-4	2.05	7.19061E-5	2.50

6.3. Wave propagation

We then consider the one-dimensional wave propagation problem to validate the order of convergence of the RelaxNet-based universal Boltzmann equation and its solution algorithm. Similar to the first case in Section 6.1, the initial flow field is set as

$$\begin{pmatrix} \rho \\ U \\ p \end{pmatrix}_{t=0} = \begin{pmatrix} 1 + 0.1 \sin(2\pi x) \\ 1 \\ 1 \end{pmatrix},$$

and the particle distribution function is set to Maxwellian everywhere in correspondence with the macroscopic flow variables. The detailed computational setup is presented in Table 9. The grid-convergent solutions with 1000 cells are used as reference.

As shown in Section 5, the first-order spatial derivatives are used to reconstruct the particle distribution functions, resulting in second order of accuracy by design. Table 10, 11, and 12 present the numerical errors and the corresponding convergence orders at different Knudsen numbers. It is clear that the current method preserves the desired accuracy in all flow regimes, which corroborates Theorem 6.

Table 12
Errors and convergences of the solution algorithm in the wave propagation problem at Kn = 0.1.

Δx	L^1 error	Order	L^2 error	Order
0.2	1.36199E-2		6.83337E-3	
0.1	3.32411E-3	2.03	1.18189E-3	2.53
0.05	8.23288E-4	2.01	2.09832E-4	2.49
0.025	2.04163E-4	2.01	3.69725E-5	2.50

Table 13
Computational setup for the normal shock structure problem.

D	\mathbf{v}	N_u	N_v	N_w	N_i
3	$[-10, 10]^3$	80	28	28	10000
ρ_s	U_s	V_s	W_s	T_s	γ_4
$\mathcal{U}(0.5, 5)$	$\mathcal{U}(-3, 3)$	0	0	$\mathcal{U}(0.5, 5)$	$\mathcal{N}(0, 0.005)$
γ_5	Kn_s	N_h	N_m	Reference	Optimizer
$\mathcal{T}(0, 0.005, -\infty, 0)$	$\mathcal{U}(0.01, 5)$	2	4	Boltzmann	Adam
t	x	N_x	Kn	Ma	ϖ
$(0, 250]$	$[-35, 35]$	100	1	$[2, 3]$	$5/3$
Integrator	Boundary	Quadrature	CFL		
Tsit5	Dirichlet	Rectangular	0.5		

6.4. Normal shock structure

We then study flow transport in inhomogeneous gases. Due to the existence of non-Maxwellian particle distribution functions [50], the normal shock wave problem is an ideal case to validate the RelaxNet-based UBE in solving highly non-equilibrium flows. The initial flow field is initialized according to the Rankine-Hugoniot jump conditions, i.e.,

$$\begin{pmatrix} \rho \\ U \\ V \\ W \\ T \end{pmatrix}_{t=0, x_L} = \begin{pmatrix} \rho_- \\ U_- \\ 0 \\ 0 \\ T_- \end{pmatrix}, \quad \begin{pmatrix} \rho \\ U \\ V \\ W \\ T \end{pmatrix}_{t=0, x_R} = \begin{pmatrix} \rho_+ \\ U_+ \\ 0 \\ 0 \\ T_+ \end{pmatrix},$$

where the upstream and downstream density, velocity, and temperature are denoted by $\{\rho_-, U_-, T_-\}$ and $\{\rho_+, U_+, T_+\}$, respectively. The initial particle distribution functions are set to Maxwellian according to the local flow variables,

$$\begin{aligned} \frac{\rho_+}{\rho_-} &= \frac{(\varpi + 1)Ma^2}{(\varpi - 1)Ma^2 + 2}, \\ \frac{U_+}{U_-} &= \frac{(\varpi - 1)Ma^2 + 2}{(\varpi + 1)Ma^2}, \\ \frac{T_+}{T_-} &= \frac{((\varpi - 1)Ma^2 + 2)(2\varpi Ma^2 - \varpi + 1)}{(\varpi + 1)^2 Ma^2}, \end{aligned}$$

where Ma is the Mach number, and ϖ is the ratio of specific heat. The upstream flow variables are used as references to dimensionless the system. The detailed computational setup is provided in Table 13. Note that the dataset used to optimize the parameters of RelaxNet consists of two parts. In addition to the data generated by Algorithm 5 with predefined parameters, as the numerical simulation progresses, the case-specific data can be supplemented on the fly, as shown in Fig. 4.

Fig. 20 shows the profiles of density, velocity, temperature, viscous stress, heat flux, and entropy density in the shock structure at Ma = 2. The definition of viscous stress is the difference between the first element in the pressure tensor and the isotropic pressure, i.e.,

$$\Delta = \mathbf{P}_{xx} - p.$$

When the Mach number is small, the particle distribution functions inside the shock wave deviate moderately away from Maxwellian, and thus the Boltzmann, BGK, and UBE solutions are roughly equivalent. The discrepancy of the viscous stress and heat flux provided by the Boltzmann and BGK models can be observed due to the fact that high-order moments are more sensitive to the nuances of distribution function than density and bulk velocity. Fig. 21 presents the macroscopic flow

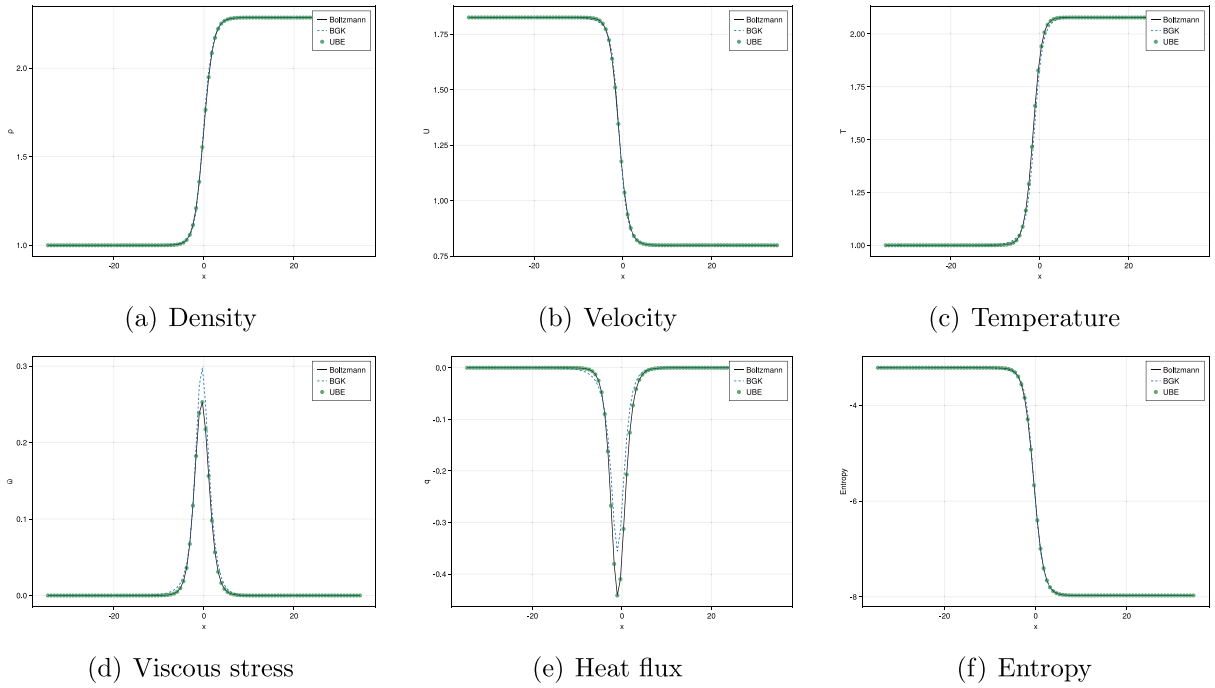


Fig. 20. Profiles of macroscopic variables at $Ma = 2$ in the normal shock structure (Section 6.4).

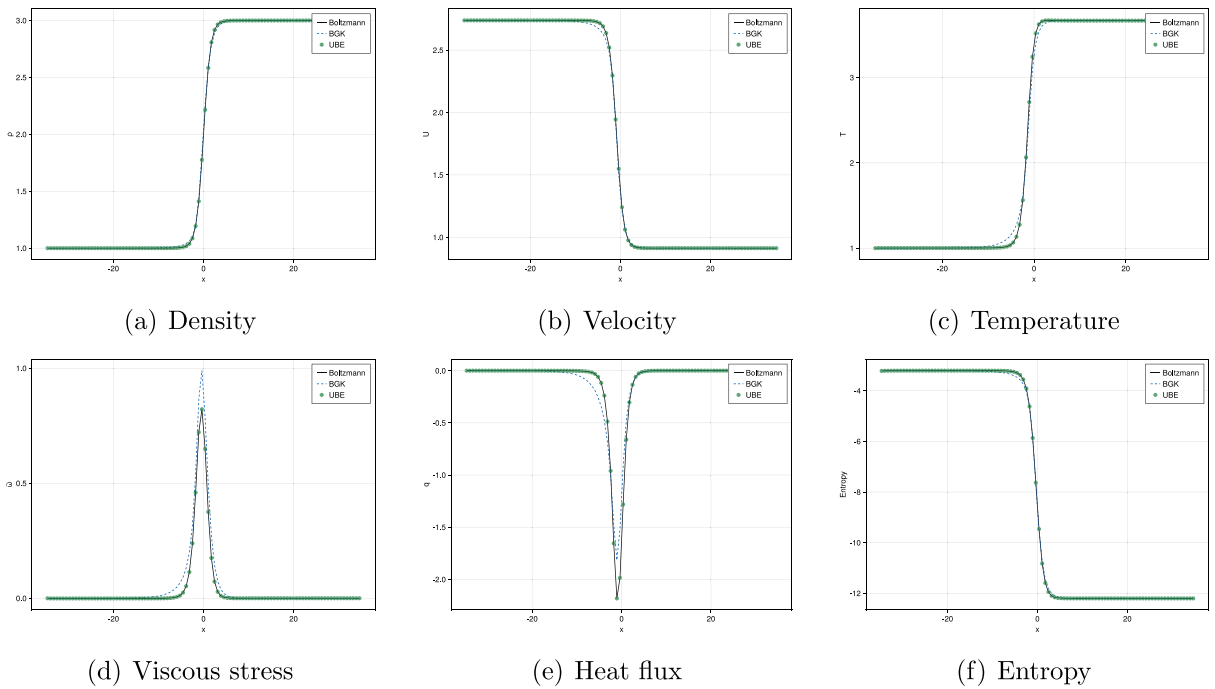


Fig. 21. Profiles of macroscopic variables at $Ma = 3$ in the normal shock structure (Section 6.4).

field at $Ma = 3$. It is obvious that the difference between the Boltzmann and BGK solutions become more significant with the increasing Mach number. The unit Prandtl number in the BGK model leads to an incorrect heat transfer rate and an early rise in the temperature profile. Thanks to the solution-dependent equilibrium and relaxation frequency, the current UBE is able to provide solutions equivalent to the reference Boltzmann results in all cases.

Fig. 22 and 23 provide the reduced distribution functions and collision terms in the $x - u$ phase space at $Ma = 2$ and $Ma = 3$. An one-to-one correspondence can be clearly observed between the evolution of macroscopic quantities and particle

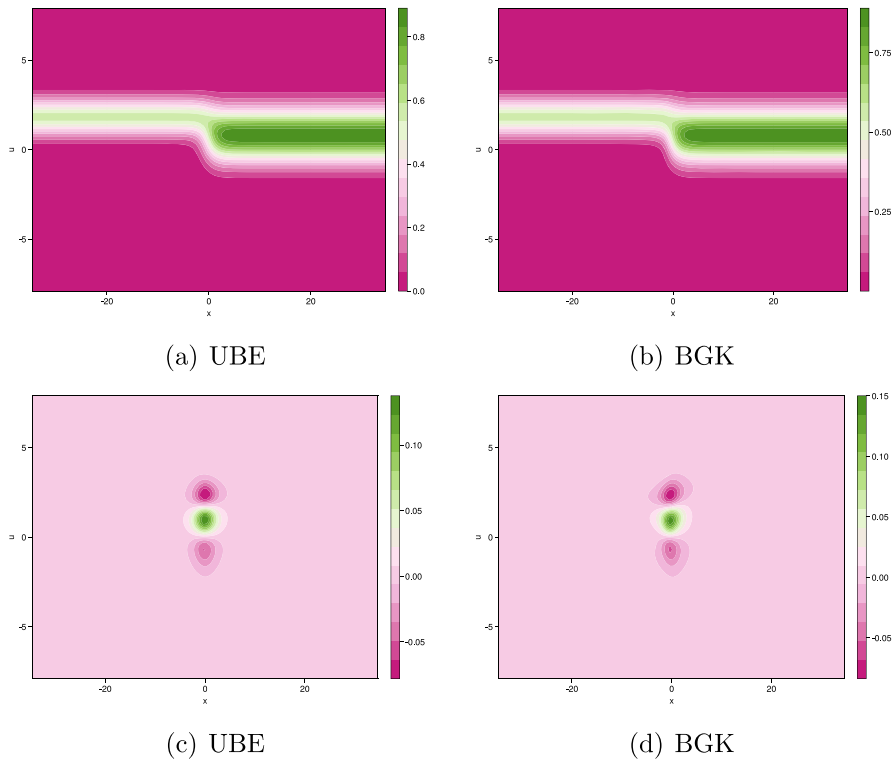


Fig. 22. Contours of particle distribution function (first row) and collision term (second row) at $Ma = 2$ in the normal shock structure (Section 6.4).

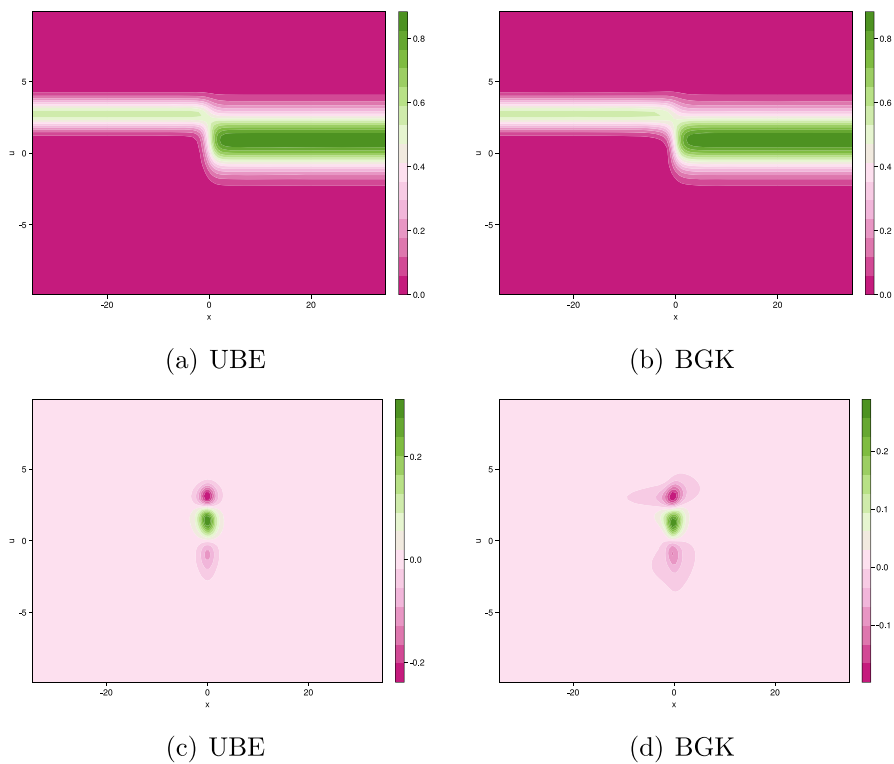


Fig. 23. Contours of particle distribution function (first row) and collision term (second row) at $Ma = 3$ in the normal shock structure (Section 6.4).

Table 14
Computational cost of computing the right-hand side of the kinetic model in the normal shock structure problem.

	Time	Allocations	Memory
Boltzmann	32.114 ms	1446	102.98 MB
BGK	5.893 μ s	36	12.92 KB
UBE	295.947 μ s	93	2.38 MB

Table 15
Computational setup for the lid-driven cavity problem.

D	\mathbf{v}	N_u	N_v	N_w	N_i
3	$[-5, 5]^3$	28	28	28	10000
ρ_s	U_s	V_s	W_s	T_s	γ_4
$\mathcal{U}(0.5, 5)$	$\mathcal{U}(-1, 1)$	$\mathcal{U}(-1, 1)$	0	$\mathcal{U}(0.5, 5)$	$\mathcal{N}(0, 0.005)$
γ_5	Kn_s	N_h	N_m	Reference	Optimizer
$\mathcal{T}(0, 0.005, -\infty, 0)$	$\mathcal{U}(0.01, 1)$	2	4	Boltzmann	Adam
\mathbf{x}	N_x	N_y	\mathbf{V}_w^u	$\mathbf{V}_w^{d,l,r}$	T_w
$[0, 1]^2$	45	45	$(0.15, 0, 0)^T$	$(0, 0, 0)^T$	1
ϖ	Kn	Integrator	Boundary	Quadrature	CFL
$5/3$	0.075	Tsit5	Maxwell	Rectangular	0.5

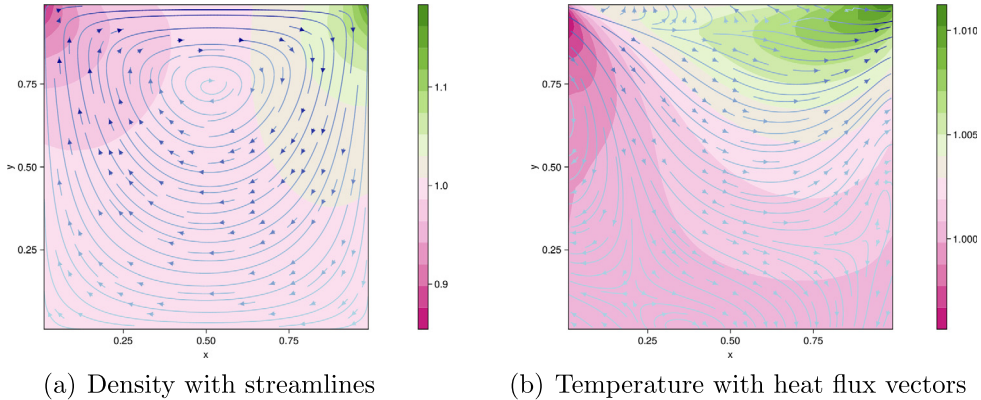


Fig. 24. Contours of density with streamlines and temperature with heat flux vectors inside the cavity (Section 6.5).

distribution functions. When $\text{Ma} = 3$, significant difference can be obtained between the BGK and UBE collision terms, resulting in different distributions of particle distribution functions and macroscopic flow variables. Table 14 shows the computational cost of computing the right-hand side operator once in the Boltzmann, BGK, and UBE models. The benchmark verifies that the RelaxNet-based UBE increases the computational efficiency by a factor of 108 and reduces the memory load to 1/43 compared to the fast spectral method of the Boltzmann equation.

6.5. Lid-driven cavity

In the last numerical experiment, we study the lid-driven cavity to validate the ability of current approach to solve non-equilibrium flows under multi-dimensional geometry. The initial flow field inside the square cavity is set as

$$\begin{pmatrix} \rho \\ U \\ V \\ W \\ T \end{pmatrix}_{t=0} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix},$$

and the particle distribution function is set to Maxwellian everywhere. The flow domain is enclosed by four isothermal solid walls. The upper wall moves in the tangent direction with $\mathbf{V}_w^u = (0.15, 0, 0)^T$, with the rest three walls kept still with $\mathbf{V}_w^{d,l,r} = (0, 0, 0)^T$. Maxwell's diffusive boundary is considered at all wall boundaries. The detailed computational setup is provided in Table 15.

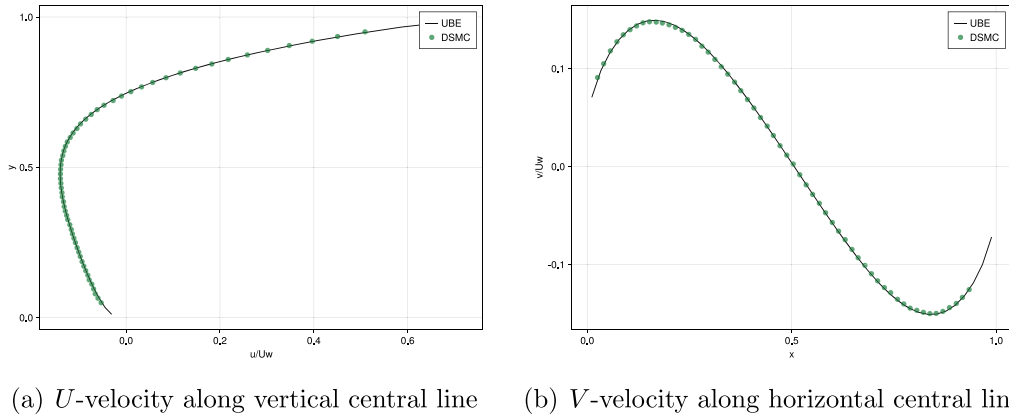


Fig. 25. Velocity profiles along vertical and horizontal central lines of the cavity. The results are normalized with the wall speed U_w (Section 6.5).

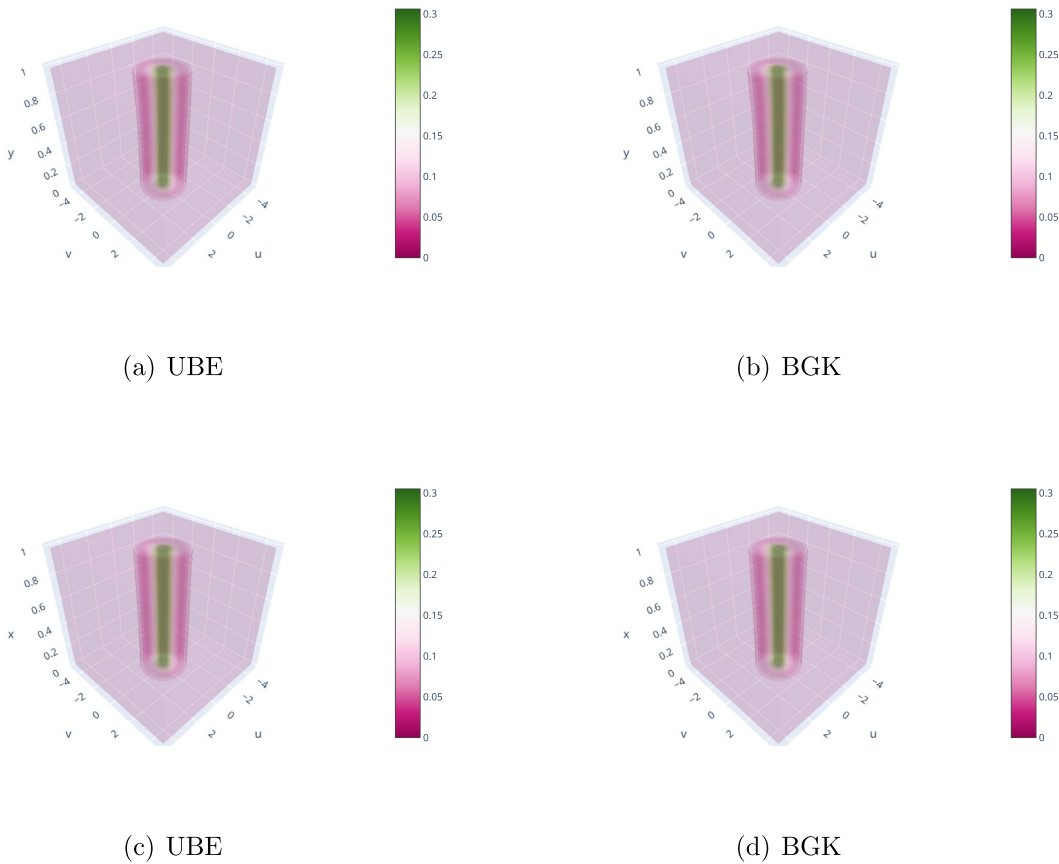


Fig. 26. Particle distribution functions along vertical (first row) and horizontal (second row) central lines of the cavity (Section 6.5).

Fig. 24 shows the contours of density with streamlines and temperature with heat flux vectors inside the cavity. Consistent with the report in [51], the inverse-Fourier heat flux driven by the viscous stress in the non-equilibrium regime is accurately identified. Fig. 25 presents the velocity profiles along the horizontal and vertical central lines of the cavity. The validity of UBE and the corresponding numerical method is fully validated by a quantitative comparison with the reference DSMC solution. Fig. 26 and 27 further provide the particle distribution functions and collision terms along the two central lines. For low-speed microflows, the non-equilibrium effect is weak and the difference between the Boltzmann and BGK solutions is mild. The correction provided by the RelaxNet-based UBE for the BGK model can be observed, which ensures the correct heat transfer rate and Prandtl number.

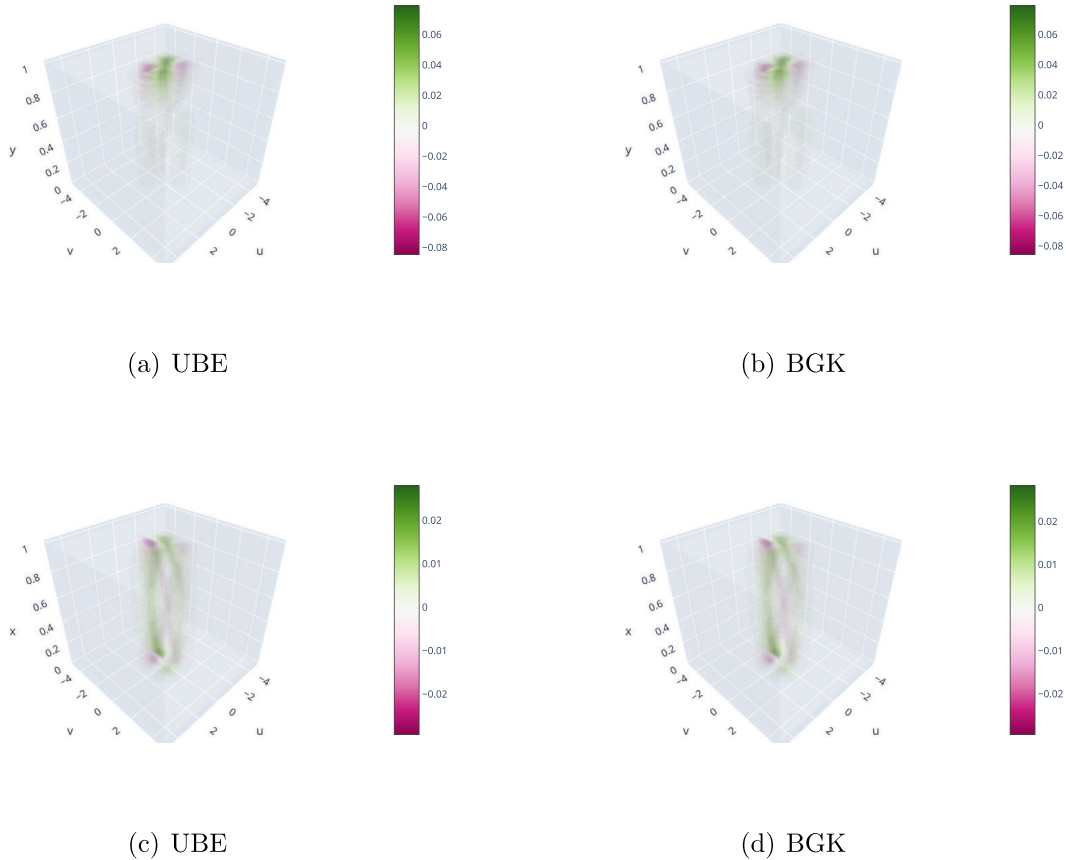


Fig. 27. Collision terms along vertical (first row) and horizontal (second row) central lines of the cavity (Section 6.5).

Conclusion

Scientific machine learning is increasingly showing its power and versatility in modeling and simulation in computational physics. In this paper, a novel relaxation neural network (RelaxNet) is proposed as a structure-preserving surrogate model of the collision operator in the Boltzmann equation. Based on the architecture of residual neural networks, RelaxNet builds a parameterized relaxation model with solution-dependent equilibrium state and frequency. The quantitative interpretability of RelaxNet makes it the first machine learning model that strictly preserves all the key structural properties of the Boltzmann equation, including invariance, conservation, H-theorem, and the correct continuum limit. Since the convolution in the original Boltzmann equation is simplified by the tensor multiplication in RelaxNet, the dimensionality and computational complexity of the collision term are greatly reduced. Based on RelaxNet, a universal Boltzmann equation (UBE) model is developed, which fuses mechanical and neural models into a single differentiable framework that is compatible with source-to-source automatic differentiation. Based on the entropy closure of the Boltzmann moment system, an algorithmic sampler is designed to generate datasets for model training and testing, and the superiority of this strategy over simulation-based samplers is demonstrated through numerical experiments. The solution algorithm for solving the RelaxNet-based UBE model is described in detail. Both spatially homogeneous and inhomogeneous test cases are presented to validate UBE and its solution algorithm. The current approach provides an accurate and efficient tool for the study of non-equilibrium flow physics. It is feasible to couple UBE with hydrodynamic solvers to build Like the network chain that contains multiple ResNet blocks, it is promising to extend RelaxNet to build multi-level relaxation models [52], which remains to be investigated in future work.

CRediT authorship contribution statement

Tianbai Xiao: Conceptualization, Formal analysis, Investigation, Methodology, Project administration, Software, Visualization, Writing – original draft, Writing – review & editing. **Martin Frank:** Conceptualization, Formal analysis, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

The current research is funded by the German Research Foundation in the frame of the priority program SPP 2298 "Theoretical Foundations of Deep Learning" with the project number 441826958.

References

- [1] Tianbai Xiao. Kinetic jl, A portable finite volume toolbox for scientific and neural computing, *J. Open Sour. Softw.* 6 (62) (2021) 3060.
- [2] Maziar Raissi, Paris Perdikaris, George E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [3] Lionel Raff, Ranga Komanduri, Martin Hagan, Satish Bukkapatnam, *Neural Networks in Chemical Reaction Dynamics*, OUP, USA, 2012.
- [4] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, E. Weinan, Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics, *Phys. Rev. Lett.* 120 (14) (2018) 143001.
- [5] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, Liu Yang, Physics-informed machine learning, *Nat. Rev. Phys.* 3 (6) (2021) 422–440.
- [6] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, Alan Edelman, Universal differential equations for scientific machine learning, preprint, arXiv:2001.04385, 2020.
- [7] Samuel Greydanus, Misko Dzamba, Jason Yosinski, Hamiltonian neural networks, *Adv. Neural Inf. Process. Syst.* 32 (2019).
- [8] Erik J. Bekkers, B-spline cnns on lie groups, preprint, arXiv:1909.12057, 2019.
- [9] Muhammed Shuaibi, Adeesh Kolluru, Abhishek Das, Aditya Grover, Anuroop Sriram, Zachary Ulissi, C. Lawrence Zitnick, Rotation invariant graph neural networks using spin convolutions, preprint, arXiv:2106.09575, 2021.
- [10] Steffen Schotthöfer, Tianbai Xiao, Martin Frank, Cory D. Hauck, Structure preserving neural networks: a case study in the entropy closure of the boltzmann equation, in: *Proceedings of the International Conference on Machine Learning*, PMLR, Baltimore, MD, USA, 2022, pp. 17–23.
- [11] E. Weinan, Jiequn Han, Linfeng Zhang, et al., Integrating machine learning with physics-based modeling, preprint, arXiv:2006.02619, 2020.
- [12] Keyulu Xu, Mozhi Zhang, Jingling Li, Simon S. Du, Ken-ichi Kawarabayashi, Stefanie Jegelka, How neural networks extrapolate: from feedforward to graph neural networks, preprint, arXiv:2009.11848, 2020.
- [13] Zheyang Shen, Jiashuo Liu, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, Peng Cui, Towards out-of-distribution generalization: a survey, preprint, arXiv:2108.13624, 2021.
- [14] Manuel Torrilhon, Modeling nonequilibrium gas flow based on moment equations, *Annu. Rev. Fluid Mech.* 48 (2016) 429–458.
- [15] Byung C. Eu, Kinetic theory and irreversible thermodynamics, *NASA STI/Recon Tech. Rep. A 93* (1992) 24498.
- [16] Prabhu Lal Bhatnagar, Eugene P. Gross, Max Krook, A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems, *Phys. Rev.* 94 (3) (1954) 511.
- [17] Lowell H. Holway Jr., New statistical models for kinetic theory: methods of construction, *Phys. Fluids* (1958–1988) 9 (9) (1966) 1658–1673.
- [18] E.M. Shakhov, Generalization of the krook kinetic relaxation equation, *Fluid Dyn.* 3 (5) (1968) 95–96.
- [19] Mieussens Luc, Henning Struchtrup, Numerical comparison of bhatnagar–gross–krook models with proper prandtl number, *Phys. Fluids* 16 (8) (2004) 2797–2813.
- [20] Jeff Haack, Cory Hauck, Christian Klingenberg, Marlies Pirner, Sandra Warnecke, A consistent bgk model with velocity-dependent collision frequency for gas mixtures, *J. Stat. Phys.* 184 (3) (2021) 1–17.
- [21] Ching Shen, *Rarefied Gas Dynamics: Fundamentals, Simulations and Micro Flows*, Springer Science & Business Media, 2006.
- [22] Jiequn Han, Arnulf Jentzen, E. Weinan, Solving high-dimensional partial differential equations using deep learning, *Proc. Natl. Acad. Sci.* 115 (34) (2018) 8505–8510.
- [23] Tianbai Xiao, Martin Frank, Using neural networks to accelerate the solution of the boltzmann equation, *J. Comput. Phys.* 443 (2021) 110521.
- [24] Alexander Alekseenko, Robert Martin, Aihua Wood, Fast evaluation of the boltzmann collision operator using data driven reduced order models, *J. Comput. Phys.* (2022) 111526.
- [25] Sean T. Miller, Nathan V. Roberts, Stephen D. Bond, Eric C. Cyr, Neural-network based collision operators for the boltzmann equation, *J. Comput. Phys.* (2022) 111541.
- [26] Sydney Chapman, Thomas George Cowling, *The Mathematical Theory of Non-uniform Gases: an Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases*, Cambridge University Press, 1970.
- [27] Graham W. Alldredge, Martin Frank, Cory D. Hauck, A regularized entropy-based moment method for kinetic equations, *SIAM J. Appl. Math.* 79 (5) (2019) 1627–1653.
- [28] François Bouchut, François Golse, Mario Pulvirenti, *Kinetic Equations and Asymptotic Theory*, Elsevier, 2000.
- [29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [30] Christopher M. Bishop, et al., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
- [31] Mohammad Amin Nabian, Hadi Meidani, Physics-informed regularization of deep neural networks, preprint, arXiv:1810.05547, 2018.
- [32] Diederik P. Kingma, Jimmy Ba Adam, A method for stochastic optimization, preprint, arXiv:1412.6980, 2014.
- [33] Mike Innes, Alan Edelman, Keno Fischer, Chris Rackauckas, Elliot Saba, Viral B. Shah, Will Tebbutt, A differentiable programming system to bridge machine learning and scientific computing, preprint, arXiv:1907.07587, 2019.
- [34] Hans Babovsky, et al., *Die Boltzmann-Gleichung: Modellbildung-Numerik-Anwendungen*, Springer, 1998.
- [35] C. David Levermore, Moment closure hierarchies for kinetic theories, *J. Stat. Phys.* 83 (5) (1996) 1021–1065.
- [36] Michael Junk, Andreas Unterreiter, Maximum entropy moment systems and galilean invariance, *Contin. Mech. Thermodyn.* 14 (2002) 563–576.
- [37] Raúl E. Curto, Recursiveness, positivity and truncated moment problems, *Houst. J. Math.* 17 (1991) 603–635.

- [38] Michael Junk, Maximum entropy for reduced moment problems, *Math. Models Methods Appl. Sci.* 10 (07) (2000) 1001–1025.
- [39] Tianbai Xiao, Steffen Schotthöfer, Martin Frank, Predicting continuum breakdown with deep neural networks, *J. Comput. Phys.* 489 (2023) 112278.
- [40] T. Carleman, L'intégrale de fourier et questions que s'y rattachent, publications scientifiques de l'institut Mittag-Leffler, vol. 1, Almqvist & Wiksells, Uppsala, 1944.
- [41] Clément Mouhot, Lorenzo Pareschi, Fast algorithms for computing the boltzmann collision operator, *Math. Comput.* 75 (256) (2006) 1833–1852.
- [42] Kun Xu, Juan-Chen Huang, An improved unified gas-kinetic scheme and the study of shock structures, *IMA J. Appl. Math.* 76 (5) (2011) 698–711.
- [43] Ruifeng Yuan, Lei Wu, Capturing the influence of intermolecular potential in rarefied gas flows by a kinetic model with velocity-dependent collision frequency, *J. Fluid Mech.* 942 (2022).
- [44] Tianbai Xiao, Martin Frank, A stochastic kinetic scheme for multi-scale flow transport with uncertainty quantification, *J. Comput. Phys.* 437 (2021) 110337.
- [45] Tianbai Xiao, A flux reconstruction kinetic scheme for the boltzmann equation, *J. Comput. Phys.* 447 (2021) 110689.
- [46] Ch Tsitouras, I. Th Famelis, T.E. Simos, On modified runge–kutta trees and methods, *Comput. Math. Appl.* 62 (4) (2011) 2101–2111.
- [47] Lawrence F. Shampine, Implementation of rosenbrock methods, *ACM Trans. Math. Softw.* 8 (2) (1982) 93–113.
- [48] Uri M. Ascher, Steven J. Ruuth, Brian TR Wetton, Implicit-explicit methods for time-dependent partial differential equations, *SIAM J. Numer. Anal.* 32 (3) (1995) 797–823.
- [49] Harold Abelson, Gerald Jay Sussman, *Structure and Interpretation of Computer Programs*, The MIT Press, 1996.
- [50] Harold Meade Mott-Smith, The solution of the boltzmann equation for a shock wave, *Phys. Rev.* 82 (6) (1951) 885.
- [51] Benzi John, Xiao-Jun Gu, David R. Emerson, Investigation of heat and mass transfer in a lid-driven cavity under nonequilibrium flow conditions, *Numer. Heat Transf., Part B, Fundam.* 58 (5) (2010) 287–303.
- [52] Dominique d'Humières, Multiple-relaxation-time lattice boltzmann models in three dimensions, *Philos. Trans. R. Soc. Lond. A, Math. Phys. Eng. Sci.* 360 (1792) (2002) 437–451.