Journal of Scientific Computing (2023) 94:7
https://doi.org/10.1007/s10915-022-02056-7

# Machine Learning Moment Closure Models for the Radiative Transfer Equation III: Enforcing Hyperbolicity and Physical Characteristic Speeds

**Juntao Huang**[1] · **Yingda Cheng**[2] · **Andrew J. Christlieb**[3] · **Luke F. Roberts**[4,5]

Received: 9 September 2021 / Revised: 23 September 2022 / Accepted: 7 November 2022 /
Published online: 23 November 2022


**Abstract**
This is the third paper in a series in which we develop machine learning (ML) moment closure models for the radiative transfer equation. In our previous work (Huang et al. in J Comput Phys 453:110941, 2022), we proposed an approach to learn the gradient of the unclosed high order moment, which performs much better than learning the moment itself and the conventional $P_N$ closure. However, while the ML moment closure has better accuracy, it is not able to guarantee hyperbolicity and has issues with long time stability. In our second paper (Huang et al., in: Machine learning moment closure models for the radiative transfer equation II: enforcing global hyperbolicity in gradient based closures, 2021. arXiv:2105.14410), we identified a symmetrizer which leads to conditions that enforce that the gradient based ML closure is symmetrizable hyperbolic and stable over long time. The limitation of this approach

Yingda Cheng: Research is supported by NSF Grants DMS-2011838 and AST-2008004.
Andrew J. Christlieb: Research is supported by AFOSR Grants FA9550-19-1-0281 and FA9550-17-1-0394; NSF Grants DMS-1912183 and AST-2008004; and DoE Grant DE-SC0017955.
Luke F. Roberts: Research is supported by NSF Grant AST-2008004; and DoE Grant DE-SC0017955.

✉ Juntao Huang
huangj75@msu.edu

Yingda Cheng
ycheng@msu.edu

Andrew J. Christlieb
christli@msu.edu

Luke F. Roberts
lfroberts@lanl.gov

1 Department of Mathematics, Michigan State University, East Lansing, MI 48824, USA

2 Department of Mathematics, Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI 48824, USA

3 Department of Computational Mathematics, Science and Engineering, Michigan State University, East Lansing, MI 48824, USA

4 National Superconducting Cyclotron Laboratory and Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA

5 CCS-2, Los Alamos National Laboratory, Los Alamos, NM 87545, USA

is that in practice the highest moment can only be related to four, or fewer, lower moments. In this paper, we propose a new method to enforce the hyperbolicity of the ML closure model. Motivated by the observation that the coefficient matrix of the closure system is a lower Hessenberg matrix, we relate its eigenvalues to the roots of an associated polynomial. We design two new neural network architectures based on this relation. The ML closure model resulting from the first neural network is weakly hyperbolic and guarantees the physical characteristic speeds, i.e., the eigenvalues are bounded by the speed of light. The second model is strictly hyperbolic and does not guarantee the boundedness of the eigenvalues. Several benchmark tests including the Gaussian source problem and the two-material problem show the good accuracy, stability and generalizability of our hyperbolic ML closure model.

**Keywords** Radiative transfer equation · Moment closure · Machine learning · Neural network · Hyperbolicity

## 1 Introduction

In this paper, we introduce an extension to our previous works on ML closures for radiative transfer modeling [23, 24]. The new approach enforces the hyperbolicity (and physical characteristic speeds) by ensuring mathematical consistency between the closure and the macroscopic model. Further, the numerical results demonstrate the plausibility of capturing kinetic effects in a moment system with a handful of moments and an appropriate closure model.

The study of radiative transfer is of vital importance in many fields of science and engineering including astrophysics [40], heat transfer [29], and optical imaging [28]. The kinetic description of radiative transfer is a integro-differential equation in six dimensions in spatial and angular spaces plus time. While there exist many numerical methods to solve this equation, from Monte Carlo methods to deterministic mesh based schemes, the fundamental fact remains that radiative transfer equation (RTE) is computationally demanding for many problems.

An alternative approach is to directly model the observables of the kinetic equations: density, momentum, energy etc. by taking moments of the kinetic equation. However, the resulting system of equations is not closed, since the equation for the $p$th moment depends on knowledge of the $(p+1)$th moment. This is known as the moment closure problem. To obtain a closed system of equations, typically a relationship has to be introduced to eliminate the dependency of the equations on the $(p+1)$th moment. This may be as simple as setting the $(p+1)$th moment to zero, or involve some other relations relating the $(p+1)$th moment to lower order moments. Many moment closure models have been developed, including the $P_N$ model [10]; the variable Eddington factor models [32, 38]; the entropy-based $M_N$ models [1, 2, 21]; the positive $P_N$ models [20]; the filtered $P_N$ models [30, 37]; the $B_2$ models [3]; and the $MP_N$ model [14, 15, 34].

In moment closure problems, hyperbolicity is a critical issue, which is essential for a system of first-order partial differential equations (PDEs) to be well-posed [45]. The pioneering work on the moment closure for the Boltzmann equation, in the context of gas kinetic theory, was introduced by Grad in [17] and is the most basic one among the moment models. Recent analysis for Grad's 13-moment model showed that the equilibrium of the model is on the boundary of the region of hyperbolicity in 3D [9]. This instability issue has led to a range

of efforts to develop closures that lead to globally hyperbolic moment systems [7, 8, 14, 15, 34].

The traditional trade off in introducing a closure relation and solving a moment model instead of a kinetic equation is generic accuracy verses practical computability. However, thanks to the rapid development of machine learning (ML) and data-driven modeling [6, 18, 42], a new approach to solve the moment closure problem has emerged based on ML [5, 19, 23–25, 35, 36, 41, 43, 44, 48]. This approach offers a path for multi-scale problems that is relatively unique, promising to capture kinetic effects in a moment model with only a handful of moments. For more detailed literature review, we refer readers to [23]. We remark that most of the works mentioned above are not able to guarantee hyperbolicity or long time stability, except the works in [24, 25, 41, 43]. In [25], based on the conservation-dissipation formalism [50] of irreversible thermodynamics, the authors proposed a stable ML closure model with hyperbolicity and Galilean invariance for the Boltzmann BGK equation. Nevertheless, the model is limited to only one extra non-equilibrium variable and it is still not clear how to generalize to an arbitrary number of moments. In [41, 43], the authors constructed ML surrogate models for the maximum entropy closure [33] of the moment system of the RTE and the Boltzmann equation. By approximating the entropy using convex splines and input convex neural networks [4], the ML model preserves the structural properties of the original system and reduces the computational cost of the associated ill-conditioned constrained optimization problem significantly, which needed to be solved at each time step in the original formulation of the maximum entropy closure.

This paper is a continuation of our previous work in [23], where we proposed to directly learn a closure that relates the gradient of the highest order moment to the gradients of the lower order moments. This gradient based closure is consistent with the exact closure for the free streaming limit and also provides a natural output normalization. A variety of numerical tests show that the ML closure model in [23] has better accuracy than an ML closure based on learning a relation between the moments, as opposed to a relation between the gradients, and the conventional $P_N$ closure. Further, the method was able to accurately model both the optically thin and optically thick regime in a single domain with only six moments and was in good agreement with moments computed from the kinetic solution. However, it is not able to guarantee hyperbolicity and long time simulations are not always satisfactory.

In our follow-up work [24], we proposed a method to enforce the global hyperbolicity of the ML closure model. The main idea is to seek a symmetrizer (a symmetric positive definite matrix) for the closure system, and derive constraints such that the system is globally symmetrizable hyperbolic. It was also shown that the hyperbolic ML closure system inherits the dissipativeness of the RTE and preserves the correct diffusion limit as the Knunsden number goes to zero. In the numerical tests, the method preformed as well as our original gradient based ML closure for short time simulations and also has the additional benefit of long time stability. A limitation of our approach in [24] is that in practice it is limited to relating the gradient of the highest moment to the gradient of the next 4 lower moments. However, our analysis in [23] indicated that in the free streaming limit, the gradient of the highest moment should be related to a range of gradients which include the lowest moments.

In this paper, to overcome this limitation, we take a different approach to enforce the hyperbolicity of our gradient based ML closure model. The approach is to design a structure preserving neural network that ensures that the desired hyperbolicity is preserved in our ML gradient based closure. The main idea is motivated by the observation that the coefficient matrix of the gradient based closure system [23] is an unreduced lower Hessenberg matrix, see Definition 2.1. Due to this particular mathematical structure, we relate its eigenvalues to the roots of some polynomials associated with the coefficient matrix. Therefore, the hyperbolicity

of the closure model is equivalent to the condition that the associated polynomial only has simple and real roots, see Theorems 2.4 and 3.1. Then, we derive the relation between the eigenvalues and the weights in the gradient based closure using the Vieta's formula and a linear transformation between monomial basis functions and Legendre polynomials. Based on this relation, we design two new neural network architectures both starting with a fully connected neural network which takes the input as the lower order moments. The first neural network architechture is then followed by a component-wise hyperbolic tangent function to enforce the boundedness of the eigenvalues, while the second one has some postprocessing layers to enforce that the eigenvalues are distinct. Lastly, two sublayers representing the Vieta's formula and a linear transformation are applied to produce the weights in the gradient based closure as the final output, see Figs. 1 and 2 in Sect. 4. The resulting ML closure model from the first neural network is weakly hyperbolic and guarantees the physical characteristic speeds, i.e. the eigenvalues lie in the range of the interval $[-1, 1]$, see Theorem 4.1, while the symmetrizer approach in [24] usually violates the physical characteristic speeds. The second model is strictly hyperbolic and does not guarantee the boundedness of the eigenvalues, see Theorem 4.2. Nevertheless, in practice, we find the characteristic speeds stay close to the physical bound. Maintaining physical characteristic speeds saves substantial computational efforts by allowing for a larger time step size, as compared to [24] when solving the closure system. We numerically tested that the hyperbolic ML closure model has good accuracy in a variety of numerical examples and, just as with our previous work, can capture accurate solutions to problems which have regions in both the optically thin and optically thick regime with only 6 moments. Further, we numerically demonstrate that as we increase the number of moments in the new approach, the ML closure converges rapidly to the solution of the kinetic equation.

Nevertheless, there exists some numerical instability for the current model when a small number of moments are used. For the first neural network, we observe numerically that the eigenvalues get too close, which behaves as if the system is weakly hyperbolic instead of strongly hyperbolic. For the second neural network, we check the linear stability of the system numerically and find that the loss of linear stability probably results in the blow up of the numerical solutions. How to stabilize the closure system, while maintaining the accuracy, is a topic to be investigated in the future.

The remainder of this paper is organized as follows. In Sect. 2, we present some preliminary results about Hessenberg matrixes. In Sect. 3, we introduce the hyperbolic ML moment closure model. In Sect. 4, we present the details in the architectures and the training of the neural networks. The effectiveness of our ML closure model is demonstrated through extensive numerical results in Sect. 5. Some concluding remarks are given in Sect. 6.

## 2 Preliminary Results About Hessenberg Matrix

In this section, we review important properties of the Hessenberg matrix. These properties facilitate directly relating the eigenvalues of a Hessenberg matrix to the roots of some associated polynomial and derive some equivalent conditions for a Hessenberg matrix to be real diagonalizable. As the matrix being real diagonalizable is equivalent to enforcing that the first-order system is hyperbolic, this is a critical aspect in the design of our structure-preserving neural network in Sects. 3 and 4.

We start with the definitions of the (unreduced) lower Hessenberg matrix and the associated polynomial sequence [13]:

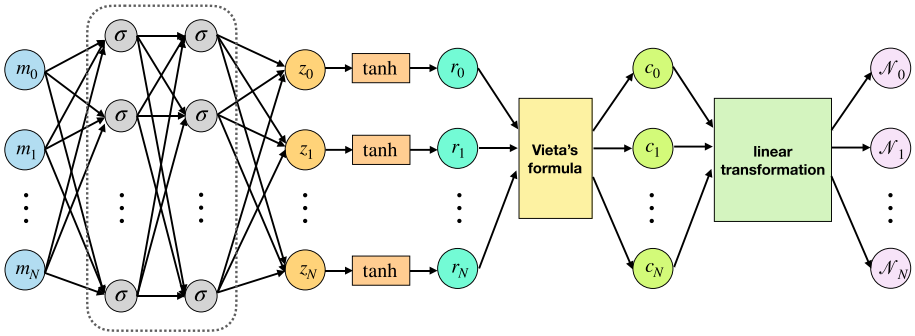



**Fig. 1** Schematic of the neural network with bounded eigenvalues. Input: the moments $(m_0, m_1, \ldots, m_N)$, output: the weights $(\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_N)$ in the gradient based closure in (3.4). The Vieta's formula is given in (3.18). The linear transformation from $(c_0, c_1, \ldots, c_N)$ to $(\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_N)$ is given in (3.27)

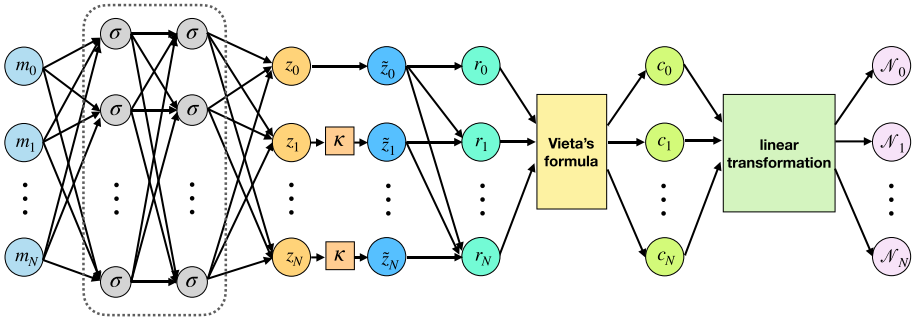



**Fig. 2** Schematic of the neural network with distinct eigenvalues. Input: the moments $(m_0, m_1, \ldots, m_N)$, output: the weights $(\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_N)$ in the gradient based closure in (3.4). Here, $\kappa = \kappa(x) > 0$ is a positive function. The layer connecting $(\tilde{z}_0, \tilde{z}_1, \ldots, \tilde{z}_N)$ and $(r_0, r_1, \ldots, r_N)$ is given in (4.3). The Vieta's formula is given in (3.18). The linear transformation from $(c_0, c_1, \ldots, c_N)$ to $(\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_N)$ is given in (3.27)

**Definition 2.1** (*lower Hessenberg matrix*) The matrix $H = (h_{ij})_{n\times n}$ is called lower Hessenberg matrix if $h_{ij} = 0$ for $j > i + 1$. It is called unreduced lower Hessenberg matrix if further $h_{i,i+1} \neq 0$ for $i = 1, 2, \ldots, n - 1$.

**Definition 2.2** (*associated polynomial sequence* [13]) Let $H = (h_{ij})_{n\times n}$ be an unreduced lower Hessenberg matrix. The associated polynomial sequence $\{q_i\}_{0\le i\le n}$ with $H$ is defined as: $q_0 = 1$, and

$$q_i(x) = \frac{1}{h_{i,i+1}}\left(xq_{i-1}(x) - \sum_{j=1}^{i} h_{ij}q_{j-1}(x)\right), \quad 1 \le i \le n, \tag{2.1}$$

with $h_{n,n+1} := 1$.

Notice that the recurrence relation in (2.1) can be written as a matrix–vector form:

$$H\boldsymbol{q}_{n-1}(x) = x\boldsymbol{q}_{n-1}(x) - q_n(x)\boldsymbol{e}_n, \tag{2.2}$$

where $\boldsymbol{q}_{n-1}(x) = (q_0(x), q_1(x), \ldots, q_{n-1}(x))^T$ and $\boldsymbol{e}_n = (0, 0, \ldots, 0, 1)^T \in \mathbb{R}^n$. From this relation, one can immediately relate the roots of $q_n$ to the eigenvalues of $H$ [13]:

**Theorem 2.3** [13] *Let $H = (h_{ij})_{n\times n}$ be an unreduced lower Hessenberg matrix and $\{q_i\}_{0\le i\le n}$ is the associated polynomial sequence with $H$. The following conclusion holds true:*

1. *If $\lambda$ is a root of $q_n$, then $\lambda$ is an eigenvalue of the matrix $H$ and a corresponding eigenvector is $(q_0(\lambda), q_1(\lambda), \ldots, q_{n-1}(\lambda))^T$;*
2. *If all the roots of $q_n$ are simple, then the characteristic polynomial of $H$ is precisely $\rho q_n$ with $\rho = \Pi_{i=1}^{n-1} h_{i,i+1}$, i.e.,*

$$\det(x I_n - H) = \rho q_n(x), \tag{2.3}$$

*where $I_n$ denotes the identity matrix of order $n$.*

By analyzing the eigenspace of the unreduced lower Hessenberg matrix, we have the following equivalent conditions for an unreduced lower Hessenberg matrix to be real diagonalizable. The proof is included in the "Appendix".

**Theorem 2.4** *Let $H = (h_{ij})_{n\times n}$ be an unreduced lower Hessenberg matrix and $\{q_i\}_{0\le i\le n}$ is the associated polynomial sequence with $H$. The following conditions are equivalent:*

1. *$H$ is real diagonalizable;*
2. *all the eigenvalues of $H$ are distinct and real;*
3. *all the roots of $q_n$ are simple and real.*

## 3 Moment Closure for Radiative Transfer Equation

In this section, we first review the gradient based ML moment closure method for the RTE in slab geometry proposed in [23]. Then, we present our approach to enforce the hyperbolicity of the ML moment closure model. Our method for enforcing hyperbolicity comes from a direct relation we derive in Sect. 3.2 between the coefficients of the neural network in the gradient based model and the eigenvalues of coefficient matrix. Given this relation, in Sect. 4 we propose two neural network architectures where we directly learn the eigenvalues of the coefficient matrix $A$ such that the eigenvalues are real. The resulting setup produces distinct eigenvalues and there by guarantees that the learned gradient based closure is hyperbolic.

### 3.1 Gradient Based ML Moment Closure

We consider the time-dependent RTE for a gray medium in slab geometry:

$$\partial_t f + v\partial_x f = \sigma_s \left( \frac{1}{2}\int_{-1}^{1} f dv - f \right) - \sigma_a f, \quad -1 \le v \le 1 \tag{3.1}$$

Here, $f = f(x, v, t)$ is the specific intensity of radiation. The variable $v \in [-1, 1]$ is the cosine of the angle between the photon velocity and the $x$-axis. $\sigma_s = \sigma_s(x) \ge 0$ and $\sigma_a = \sigma_a(x) \ge 0$ are the scattering and absorption coefficients.

Denote the $k$-th order Legendre polynomial by $P_k = P_k(x)$. Define the $k$-th order moment by

$$m_k(x, t) = \frac{1}{2}\int_{-1}^{1} f(x, v, t) P_k(v) dv. \tag{3.2}$$

Multiplying by $P_k(v)$ on both sides of (3.1) and integrating over $v \in [-1, 1]$, we derive the moment equations:

$$\begin{aligned}
&\partial_t m_0 + \partial_x m_1 = -\sigma_a m_0 \\
&\partial_t m_1 + \frac{1}{3}\partial_x m_0 + \frac{2}{3}\partial_x m_2 = -(\sigma_s + \sigma_a) m_1 \\
&\cdots \\
&\partial_t m_{N-1} + \frac{N-1}{2N-1}\partial_x m_{N-2} + \frac{N}{2N-1}\partial_x m_N = -(\sigma_s + \sigma_a) m_{N-1} \\
&\partial_t m_N + \frac{N}{2N+1}\partial_x m_{N-1} + \frac{N+1}{2N+1}\partial_x m_{N+1} = -(\sigma_s + \sigma_a) m_N
\end{aligned} \tag{3.3}$$

The above system is clearly not closed due to the existence of $\partial_x m_{N+1}$ in the last equation. The learning gradient approach proposed in [23] is to find a relation between $\partial_x m_{N+1}$ and the gradients on lower order moments:

$$\partial_x m_{N+1} = \sum_{i=0}^{N} \mathcal{N}_i(m_0, m_1, \ldots, m_N)\partial_x m_i \tag{3.4}$$

with $\mathcal{N} = (\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_N) : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$ approximated by a neural network and learned from data. Plugging (3.4) into the closure system, we derive the moment closure model:

$$\begin{aligned}
&\partial_t m_0 + \partial_x m_1 = -\sigma_a m_0 \\
&\partial_t m_1 + \frac{1}{3}\partial_x m_0 + \frac{2}{3}\partial_x m_2 = -(\sigma_s + \sigma_a) m_1 \\
&\cdots \\
&\partial_t m_{N-1} + \frac{N-1}{2N-1}\partial_x m_{N-2} + \frac{N}{2N-1}\partial_x m_N = -(\sigma_s + \sigma_a) m_{N-1} \\
&\partial_t m_N + \frac{N}{2N+1}\partial_x m_{N-1} + \frac{N+1}{2N+1}\left(\sum_{k=0}^{N} \mathcal{N}_k(m_0, m_1, \ldots, m_N)\partial_x m_k\right) = -(\sigma_s + \sigma_a) m_N.
\end{aligned} \tag{3.5}$$

In the numerical tests, this approach is shown to be accurate in the optically thick regime, intermediate regime and the optically thin regime. Moreover, the accuracy of this gradient-based model is much better than the approach based on creating a ML closure directly trained to match the moments, as well as the conventional $P_N$ closure. However, this model exhibits numerical instability due to the loss of hyperbolicity [23]. This severely restricts the application of this model, especially for long time simulations.

### 3.2 Hyperbolic ML Moment Closure

In this work, our main idea to enforce the hyperbolicity is motivated by the observation that the coefficient matrix of the closure system is a lower Hessenberg matrix. We write the closure model (3.5) into an equivalent form:

$$\partial_t \boldsymbol{m} + A\partial_x \boldsymbol{m} = S\boldsymbol{m} \tag{3.6}$$

with $\boldsymbol{m} = (m_0, m_1, \ldots, m_N)^T$ and the coefficient matrix $A \in \mathbb{R}^{(N+1)\times(N+1)}$:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ \frac{1}{3} & 0 & \frac{2}{3} & 0 & \ldots & 0 \\ 0 & \frac{2}{5} & 0 & \frac{3}{5} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & \frac{N-1}{2N-1} & 0 & \frac{N}{2N-1} \\ a_0 & a_1 & \ldots & a_{N-2} & a_{N-1} & a_N \end{pmatrix} \tag{3.7}$$

with

$$a_j = \begin{cases} \dfrac{N+1}{2N+1}\mathcal{N}_j, & j \neq N-1, \\ \dfrac{N}{2N+1} + \dfrac{N+1}{2N+1}\mathcal{N}_j, & j = N-1. \end{cases} \tag{3.8}$$

and the source term

$$S = \mathrm{diag}(-\sigma_a, -(\sigma_s + \sigma_a), \ldots, -(\sigma_s + \sigma_a)). \tag{3.9}$$

In what follows, we will use the properties of the Hessenberg matrix in Sect. 2 to analyze the real diagonalizability of the coefficient matrix $A$ in (3.7).

We first write down the associated polynomial sequence of $A$ using the definition (2.1):

$$q_0(x) = 1, \tag{3.10a}$$

$$\frac{i}{2i-1}q_i(x) = xq_{i-1}(x) - \frac{i-1}{2i-1}q_{i-2}(x), \quad i = 1, \ldots, N \tag{3.10b}$$

$$q_{N+1}(x) = xq_N(x) - \sum_{k=0}^{N} a_k q_k(x). \tag{3.10c}$$

Notice that (3.10b) is exactly the same as the recurrence relation for the Legendre polynomial. Thus, we have

$$q_i(x) = P_i(x), \quad i = 0, 1, \ldots, N. \tag{3.11}$$

Then from (3.10c), we derive

$$q_{N+1}(x) = \frac{N+1}{2N+1}P_{N+1}(x) + \frac{N}{2N+1}P_{N-1}(x) - \sum_{k=0}^{N} a_k P_k(x), \tag{3.12}$$

where we used the recurrence relation for the Legendre polynomial:

$$\frac{N+1}{2N+1}P_{N+1}(x) = xP_N(x) - \frac{N}{2N+1}P_{N-1}(x). \tag{3.13}$$

By Theorem 2.3, it is easy to derive the following theorem:

**Theorem 3.1** *For the coefficient matrix $A$ in (3.7), the associated polynomial sequence satisfies:*

$$q_i(x) = P_i(x), \quad i = 0, 1, \ldots, N, \tag{3.14a}$$

$$q_{N+1}(x) = \frac{N+1}{2N+1}P_{N+1}(x) + \frac{N}{2N+1}P_{N-1}(x) - \sum_{k=0}^{N} a_k P_k(x), \tag{3.14b}$$

*where $P_n(x)$ denotes the Legendre polynomial of degree n. If all the roots of $q_{N+1}(x)$ are simple, then the characteristic polynomial of A is:*

$$\det(xI_{N+1} - A) = \rho q_{N+1}(x) = \rho\left(\frac{N+1}{2N+1}P_{N+1}(x) + \frac{N}{2N+1}P_{N-1}(x) - \sum_{k=0}^{N} a_k P_k(x)\right) \tag{3.15}$$

*with $\rho = \frac{N!}{(2N-1)!!}$. If further assuming all the roots of $q_{N+1}(x)$ are simple and real, then all the eigenvalues of A are distinct and real. In this case, the moment closure system is strictly hyperbolic. If further assuming all the roots of $q_{N+1}(x)$ are simple, real and lie in the interval $[-1, 1]$, then the moment closure system is strictly hyperbolic with physical characteristic speeds.*

***Remark 3.2*** From Theorem 2.4, the condition that all the roots of $q_{N+1}(x)$ are simple and real, is also necessary for the moment closure system to be hyperbolic.

Next, we will derive the relation between the eigenvalues of $A$ (or the roots of $q_{N+1}(x)$) and the weights of the gradients in (3.4). In particular, we will represent $\{\mathcal{N}_k\}_{0\le k\le N}$ in (3.4) using the eigenvalues of $A$.

We denote the distinct real eigenvalues of $A$ by $\{r_k\}_{0\le k\le N}$. Then, by Theorem 3.1, we have

$$(x - r_0)(x - r_1)\cdots(x - r_N) = \rho\left(\frac{N+1}{2N+1}P_{N+1}(x) + \frac{N}{2N+1}P_{N-1}(x) - \sum_{k=0}^{N} a_k P_k(x)\right). \tag{3.16}$$

First, we expand the characteristic polynomial using a set of monomial basis:

$$\det(xI_{N+1} - A) = c_0 + c_1 x + \cdots + c_N x^N + x^{N+1}. \tag{3.17}$$

Using Vieta's formulas, we relate the coefficients $\{c_k\}_{0\le k\le N}$ to the sums and products of its roots $\{r_k\}_{0\le k\le N}$:

$$\begin{aligned}
&r_0 + r_1 + \cdots + r_{N-1} + r_N = -c_N,\\
&(r_0r_1 + r_0r_2 + \cdots + r_0r_N) + (r_1r_2 + r_1r_3 + \cdots + r_1r_N) + \cdots + r_{N-1}r_N = c_{N-1},\\
&\quad\vdots\\
&r_0r_1\cdots r_{N-1}r_N = (-1)^{N+1}c_0,
\end{aligned} \tag{3.18}$$

or equivalently written as a compact formulation

$$\sum_{0\le i_1<i_2<\cdots<i_k\le N}\left(\Pi_{j=1}^{k} r_{i_j}\right) = (-1)^k c_{N+1-k}, \quad k = 1, 2, \ldots, N+1. \tag{3.19}$$

Here the indices $i_k$ are sorted in strictly increasing order to ensure each product of $k$ roots is used exactly once.

Then, we establish the relationship between $\{c_k\}_{0\le k\le N}$ to $\{a_k\}_{0\le k\le N}$. Using the generating function of Legendre polynomials, one can express the monomial in terms of a summation of Legendre polynomials [47]. We present the conclusion in the following lemma and include the proof in the "Appendix".

**Lemma 3.3** *For any integer $m \geq 0$, there holds the following equality:*

$$x^m = \sum_{k=0}^{\lfloor m/2 \rfloor} F(m,k) P_{m-2k}(x), \tag{3.20}$$

*with $F(m,k) = \frac{m!(2m-4k+1)}{2^k k!(2m-2k+1)!!}$. Here $P_n(x)$ is the n-th order Legendre polynomial, and $\lfloor \cdot \rfloor$ is the floor function which takes a real number x as input, and gives the greatest integer less than or equal to x as output.*

We rewrite (3.20) into an equivalent formulation:

$$x^m = \sum_{k=0}^{m} b_{mk} P_k(x), \quad m \geq 0, \tag{3.21}$$

with

$$b_{mk} = \begin{cases} F\left(m, \frac{1}{2}(m-k)\right), & \text{if } m \equiv k \pmod 2, \\ 0, & \text{otherwise.} \end{cases} \tag{3.22}$$

From this formula, we can expand any polynomial $\sum_{i=0}^{n} c_i x^i$ in terms of Legendre polynomials:

$$\sum_{i=0}^{n} c_i x^i = \sum_{i=0}^{n} c_i \left( \sum_{k=0}^{i} b_{ik} P_k(x) \right) = \sum_{k=0}^{n} \left( \sum_{i=k}^{n} c_i b_{ik} \right) P_k(x) = \sum_{k=0}^{n} \alpha_k P_k(x), \tag{3.23}$$

with

$$\alpha_k = \sum_{i=k}^{n} c_i b_{ik}. \tag{3.24}$$

We apply the above relation to derive the relationship between $\{c_k\}_{0 \leq k \leq N}$ to $\{a_k\}_{0 \leq k \leq N}$:

$$c_0 + c_1 x + \cdots + c_N x^N + x^{N+1} = \rho \left( \frac{N+1}{2N+1} P_{N+1}(x) + \frac{N}{2N+1} P_{N-1}(x) - \sum_{k=0}^{N} a_k P_k(x) \right), \tag{3.25}$$

and obtain

$$-\rho a_k = \sum_{i=k}^{N+1} c_i b_{ik}, \quad k = 0, 1, \ldots, N-3, N-2, N, \tag{3.26a}$$

$$\rho \left( \frac{N}{2N+1} - a_{N-1} \right) = \sum_{i=N-1}^{N+1} c_i b_{i,N-1}, \tag{3.26b}$$

$$\rho \frac{N+1}{2N+1} = \sum_{i=N+1}^{N+1} c_i b_{i,N+1}, \tag{3.26c}$$

with $c_{N+1} := 1$. The last one (3.26c) is automatically satisfied since $b_{N+1,N+1} = F(N+1,0) = \frac{(N+1)!(2N+3)}{(2N+3)!!} = \frac{(N+1)!}{(2N+1)!!}$.

Lastly, we rewrite (3.26a)–(3.26b) in terms of $\{\mathcal{N}_k\}_{0\le k\le N}$ using the relation (3.8):

$$\mathcal{N}_k = -\frac{2N+1}{\rho(N+1)}\sum_{i=k}^{N+1} c_i b_{ik}, \quad k = 0, 1, \ldots, N, \tag{3.27}$$

with $c_{N+1} := 1$.

Now, together with (3.18) and (3.27), we have expressed $\{\mathcal{N}_k\}_{0\le k\le N}$ using the eigenvalues $\{r_k\}_{0\le k\le N}$.

## 4 Architectures and Training of the Neural Network

In this section, we provide the architectures and training of the proposed neural networks that enforces the hyperbolicity of the closure system.

### 4.1 Architectures of the Neural Network

We start with the first neural network architecture. As shown in Fig. 1, this neural network begins with a fully connected neural network denoted by $\mathcal{M}_\theta : \mathbb{R}^{N+1} \to \mathbb{R}^{N+1}$ with the input being the lower order moments $(m_0, m_1, \ldots, m_N)$ and the output denoted by $(z_0, z_1, \ldots, z_N)$. Here $\theta$ denotes the collection of all the parameters to be trained in the neural network. It is then followed by a component-wise hyperbolic tangent function to enforce the boundness of the eigenvalues, i.e. $r_i = \tanh(z_i)$ for $i = 0, 1, \ldots, N$. Lastly, two sublayers representing the Vieta's formula (3.18) and a linear transformation (3.27) are applied to produce the weights $(\mathcal{N}_0, \mathcal{N}_1, \ldots, \mathcal{N}_N)$ in the gradient based closure in (3.4) as the final output. For the ML moment closure model resulted by this neural network in Fig. 1, we have the following conclusion:

**Theorem 4.1** *The ML moment closure model* (3.5) *resulting from the neural network with bounded eigenvalues shown in* Fig. 1 *is weakly hyperbolic. Moreover, it guarantees the physical characteristic speeds, i.e., the eigenvalues lie in the interval* $[-1, 1]$.

There is a small gap between the implementation of the neural network in Fig. 1 and the theory presented in previous sections. From Theorem 2.4, all the eigenvalues being distinct and real is a necessary and sufficient condition for the moment closure system to be hyperbolic. However, we do not force all the eigenvalues to be distinct in the current neural network architecture. Thus, this only guarantees that the resulting system is theoretically weakly hyperbolic instead of the system being hyperbolic, and might cause an instability issue.

In the numerical tests in Sect. 5, we observe that the ML closure model is numerically stable for $N \ge 6$. Meanwhile, for $N \le 5$, the model has some stability issues, and these stability issues are associated with the case when two of the eigenvalues are within $10^{-3}$ of each other on a range of grid points, see the detailed discussion in Fig. 7 in Sect. 5.

To fix this problem, we tried several approaches by enforcing that the eigenvalues are distinct (or well separated). The first approach is to divide the interval $[-1, 1]$ into $(N+1)$ uniform subintervals with some threshold gap between two neighbouring subintervals: $I_k = [-1 + \frac{2k}{N+1} + \gamma, -1 + \frac{2(k+1)}{N+1} - \gamma]$ for $k = 0, \ldots, N$. Then, we put exactly one eigenvalue into each subinterval, enforced by a scaled hyperbolic tangent function. Here $\gamma \ge 0$ is a small number to guarantee a minimum distance of any two eigenvalues. We take $\gamma = 0$ and $10^{-3}$ in the implementation. This approach is motivated by the fact that, in the $P_N$

closure, each subinterval contains exactly one eigenvalue. However, we find that, the neural network results in large training errors in the training process, which are generally larger than 14% with $N = 3, 4, \ldots, 10$. In these tests, we fix the number of nodes to be 64 and the number of layers to be 6. This indicates that the assumption of the uniform distribution of the eigenvalues is too restrictive, so that the approximation power of the neural network is not enough to produce an accurate closure. We will not focus on this neural network in the afterwards.

The other approach is to replace the hyperbolic tangent layer in Fig. 1 with some other postprocessing layers. As illustrated in Fig. 2, we first applied some positive function, $\kappa$, to the outputs of the fully-connected neural network except for the first component:

$$\tilde{z}_0 = z_0, \qquad \tilde{z}_i = \kappa(z_i), \quad i = 1, \ldots, N, \tag{4.1}$$

Here $\kappa = \kappa(x) \geq \gamma > 0$ is a strictly positive function taken as

$$\kappa(x) = \ln(1 + e^x) + \gamma \tag{4.2}$$

with $\gamma = 0.1$. Then, it is followed by a linear transformation:

$$r_i = \sum_{k=0}^{i} \tilde{z}_k, \quad i = 0, \ldots, N, \tag{4.3}$$

which produces the eigenvalues of the closure system. Next, the Vieta's formula and the linear transformation are imposed as in Fig. 1. This approach can guarantee that the eigenvalues are distinct, i.e. $r_0 < r_1 < \cdots < r_N$, but may lose the boundness property of the eigenvalues. Nevertheless, in the numerical simulations, we observe that the model has the physical characteristic speeds for most of the time although this constrain is not enforced explicitly, see the discussion in Fig. 6 in Sect. 5. For the ML moment closure model resulted from this neural network in Fig. 2, we have the following conclusion:

**Theorem 4.2** *The ML moment closure model* (3.5) *resulted from the neural network with distinct eigenvalues shown in* Fig. 2 *is strictly hyperbolic.*

We remark that the the current neural network architectures could not guarantee (strongly) hyperbolicity and the physical characteristic speeds simultaneously. We also tried another approach by adding the hyperbolic tangent function after the roots $r_i$ for $i = 0, \ldots, N$ before feeding into the Vieta's formula in Fig. 2. The eigenvalues are thus bounded and also distinct since tanh is a monotone increasing function with the range between -1 and 1. Theoretically, this could make the eigenvalues bounded and also distinct. However, we observed numerically that the resulted roots will not be well separated when they are away from zero (e.g. $|r_i| > 2$) and fall in the flat region of the hyperbolic tangent function. This makes the system weakly hyperbolic and result in the instability.

We note that, in the neural network architecture shown in Fig. 1, the hyperbolic tangent function could be also removed. In this case, the boundness of the eigenvalues could be enforced by adding penalty in the loss function:

$$\mathcal{L} = \frac{1}{N_{\text{data}}} \sum_{j,n} \left| \partial_x m_{N+1}^{\text{true}}(x_j, t_n) - \partial_x m_{N+1}^{\text{appx}}(x_j, t_n) \right|^2 + \frac{\lambda}{N_{\text{data}}} \sum_{j,n} \sum_{i=0}^{N} \text{ReLU}(r_i^2(x_j, t_n) - 1). \tag{4.4}$$

Here $r_i(x_j, t_n)$ for $i = 0, 1, \ldots, N$ denotes the eigenvalues at $(x_j, t_n)$, and $\lambda > 0$ is the penalty parameter. For this approach, we did not find a set of hyperparameters to reach the reasonable training error. Nevertheless, we will more fully explore this direction in our future work.

### 4.2 Training of the Neural Network

For the training of the neural network, we take 1000 total epochs (the number of iterations in the optimization process). We investigated two activation functions including the hyperbolic tangent (tanh) function and Rectified Linear Unit (ReLU) function. The learning rate is set to be $10^{-3}$ in the initial epoch and decays by 0.5 every 100 epochs. The $L^2$ regularization is applied with weight $10^{-7}$. The batch size is taken to be 1024. The training is implemented within the PyTorch framework [39]. We use the same hyperparameters for the two neural networks.

Following [23], in the training process, the loss function is taken to be:

$$\mathcal{L} = \frac{1}{N_{\text{data}}} \sum_{j,n} \left| \partial_x m_{N+1}^{\text{true}}(x_j, t_n) - \partial_x m_{N+1}^{\text{appx}}(x_j, t_n) \right|^2. \tag{4.5}$$

Here, $\partial_x m_{N+1}^{\text{true}}(x_j, t_n)$ denotes the spatial derivative of $(N+1)$-th order moment at $x = x_j$ and $t = t_n$ computed from the kinetic solver and $\partial_x m_{N+1}^{\text{appx}}(x_j, t_n)$ comes from the evaluation of the neural network using (3.4).

Following [23], the training data comes from numerically solving the RTE (3.1) using the space-time discontinuous Galerkin (DG) method [11, 12] with a range of initial conditions in the form of truncated Fourier series and different scattering and absorption coefficients which are constants over the computational domain. We consider the unit interval [0, 1] in the physical domain with periodic boundary conditions. The initial condition is taken to be an isotropic distribution in the form of a truncated Fourier series:

$$f_0(x, v) = a_0 + \sum_{k=1}^{k_{\max}} a_k \sin(2k\pi x + \phi_k). \tag{4.6}$$

Here, we take $k_{\max} = 10$ in our dataset. For $k \geq 1$, $a_k$ and $\phi_k$ are random variables sampled from the uniform distributions on $[-\frac{1}{k}, \frac{1}{k}]$ and $[0, 2\pi]$, respectively. We take $a_0 = c + \sum_{k=1}^{k_{\max}} \frac{1}{k}$ with $c$ a random variable sampled from the uniform distributions on [0, 1]. This guarantees the positivity of the distribution function. Both $\sigma_s$ and $\sigma_a$ are constants over the domain for each run. The scattering coefficient $\sigma_s$ is sampled from a log-uniform distribution on [0.1, 100]. The absorption coefficient $\sigma_a$ are randomly sampled from [0, 10]. In the current work, we train with 100 different initial data. For each initial data set, we run the numerical solver up to $t = 1$. The other parameters are the same as in [23].

To evaluate the accuracy in the training process, we define the relative $L^2$ error for the gradient to be

$$E_2 = \sqrt{\frac{\sum_{j,n} (\partial_x m_{N+1}^{\text{true}}(x_j, t_n) - \partial_x m_{N+1}^{\text{appx}}(x_j, t_n))^2}{\sum_{j,n} (\partial_x m_{N+1}^{\text{true}}(x_j, t_n))^2}}. \tag{4.7}$$

The depth and width of neural networks (i.e., the number of hidden layers and the number of nodes in the hidden layers) are crucial hyperparameters in a neural network. Here, we perform a grid search to find the optimal hyperparameters of the neural network including the number of layers and the number of nodes in the first fully-connected neural network $\mathcal{M}_\theta$. In particular, we take the number of layers to be $\{2, 3, \ldots, 10\}$ and the number of nodes to be $\{16, 32, \ldots, 256\}$. For the first neural network in Fig. 1, the relative $L^2$ errors in the training data with different depths and widths, and tanh and ReLU activation functions are shown in Fig. 3. Here, we only show the cases with the number of moments to be $N = 5, 7, 9$, the

cases with $N = 6, 8$ are similar. With the ReLU activation function, the error decreases when we increase the number of layers and nodes in hidden layers until it saturates, see Fig. 3b, d. However, we observe a different phenomenon with the hyperbolic tangent activation function. When we increase the depth and width, the error decreases only when the network stay relatively small widths 16, 32 and 64 in Fig. 3a and widths 16 and 32 in Fig. 3c. When the neural networks get deeper, the error increases with width. This numerical observation is similar to the well-known vanishing gradient problem. In our current setup, the problem is probably caused by the strong nonlinearity of the Vieta's formula after the fully connected neural network, which stops the neural network from further training. The hyperbolic tangent function, as the activation function, has gradients in the range of $(0, 1)$, which makes it easy for the neural network to become stuck in a local minimum due to the vanishing gradient problem. ReLU suffers less from the vanishing gradient problem than the hyperbolic tangent function, because it only saturates in one direction, the one with negative inputs. Other solutions to the vanishing gradient problem, such as residual neural networks (ResNet) [22] and batch normalization [26], may also be applied here to achieve better performance. We will explore this direction in our future work. Moreover, these tests indicate that taking number of layers to be 6 and number of nodes to be 64 and ReLU activation function are good hyperparameters for our neural network. As such these are the values used in all the numerical tests in Sect. 5 unless otherwise stated.

## 5 Numerical Tests

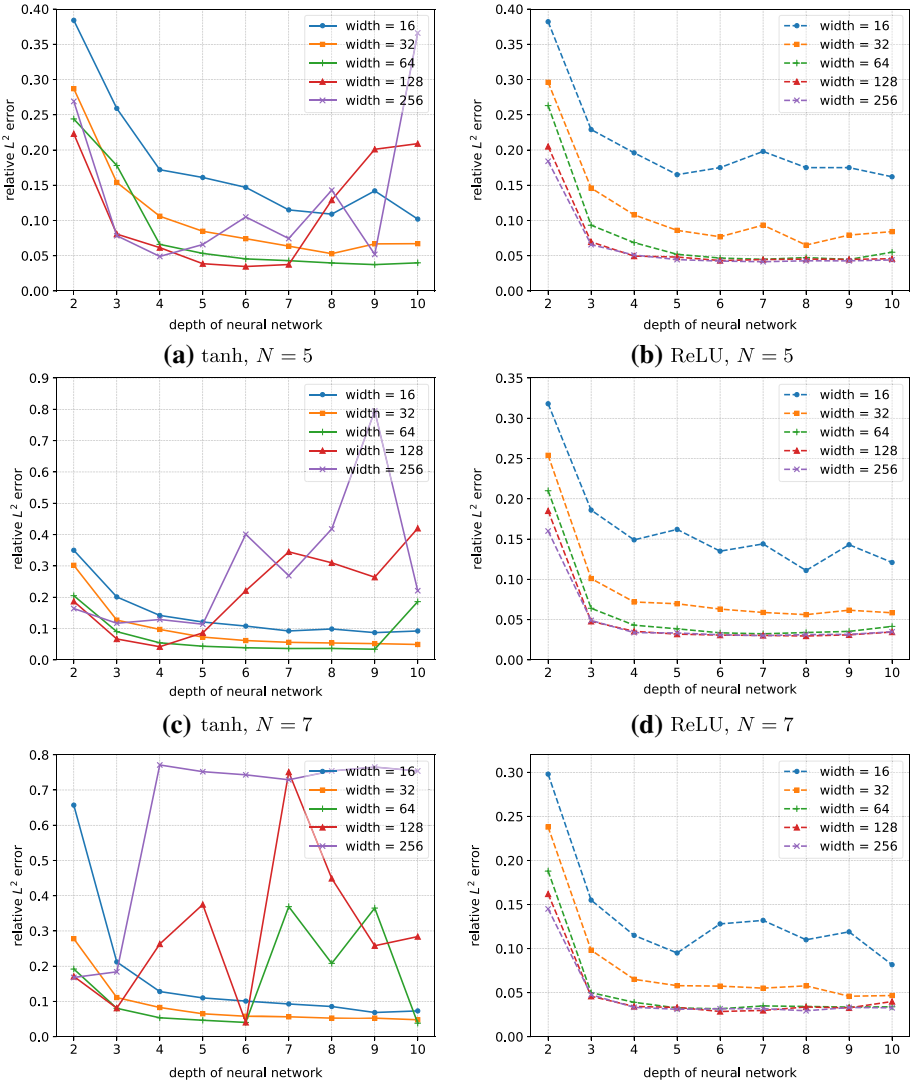
In this section, we show the performance of our ML closure model on a variety of benchmark tests, including problems with constant scattering and absorption coefficients, Gaussian source problems and two-material problems. The main focus of the tests is on the comparison of four moment closure models: (i) the symmetrizer based hyperbolic ML closure [24] (termed as "hyperbolic (symmetrizer)"); (ii) the hyperbolic ML closure with bounded eigenvalues (termed as "hyperbolic (bound)"), see the neural network architechture in Fig. 1; (iii) the hyperbolic ML closure with distinct eigenvalues (termed as "hyperbolic (distinct)"), see the neural network architechture in Fig. 2; (iv) the classical $P_N$ closure [10].

In all the numerical examples, we take the physical domain to be the unit interval $[0, 1]$ and periodic boundary conditions are imposed. To numerically solve the moment closure system, we apply the fifth-order finite difference WENO scheme [27] with a Lax–Friedrichs flux splitting for the spatial discretization, and the third-order strong-stability-preserving Runge-Kutta (RK) scheme [46] for the time discretization. We take the grid number in space to be $N_x = 256$. The CFL condition is taken to be $\Delta t = 0.8\Delta x/c$ with $c$ being the maximum eigenvalues in all the grid points.

***Example 5.1*** (*constant scattering and absorption coefficients*) The setup of this example is the same as the data preparation. The scattering and absorption coefficients are taken to be constants over the domain. The initial condition is taken to be a truncated Fourier series, see the details in [23].

In Fig. 4, we show the numerical solutions of $m_0$ and $m_1$ with seven moments in the closure system ($N = 6$) in the optically thin regime ($\sigma_s = \sigma_a = 1$). It is observed that, at $t = 0.5$ and $t = 1$, all the hyperbolic ML moment closures agree well the RTE. As a comparison, the $P_N$ closure has large deviations from the exact solution at both $t = 0.5$ and $t = 1$.

In Fig. 5, we display the log-log scatter plots of the relative $L^2$ error versus the scattering coefficient for $N = 6$ at $t = 1$. We observe that, all the hyperbolic ML closures have better




**(a)** tanh, $N = 5$ **(b)** ReLU, $N = 5$

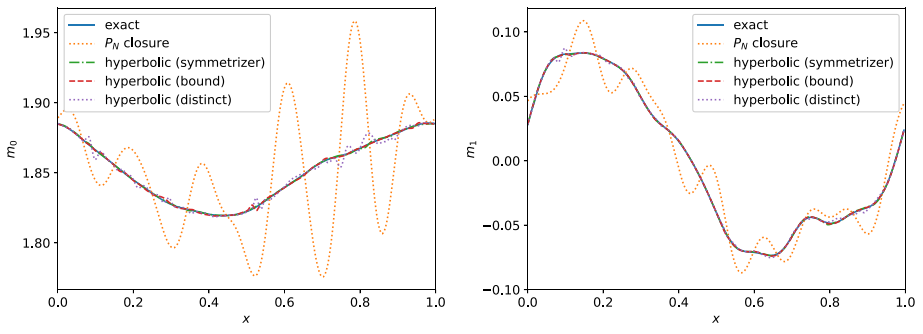
**(c)** tanh, $N = 7$ **(d)** ReLU, $N = 7$

**(e)** tanh, $N = 9$ **(f)** ReLU, $N = 9$

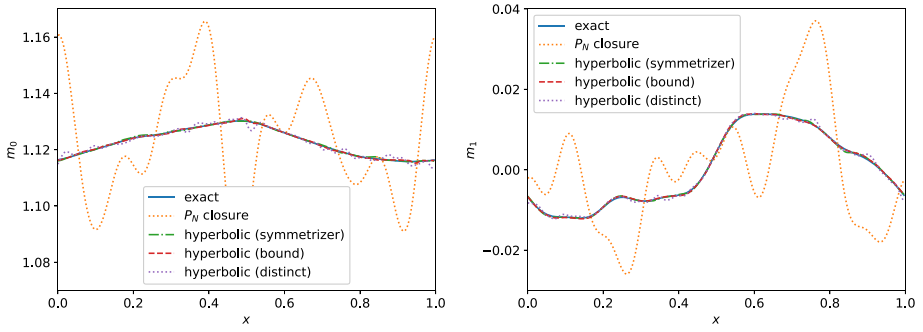
**Fig. 3** Relative $L^2$ error in the training data with different depths and widths of the neural networks. Here, we use the first neural network architecture in Fig. 1. The number of layers: 2, 3, . . . , 10; the number of nodes in the hidden layers: 16, 32, . . . , 256. Left: hyperbolic tangent activation function; right: ReLU activation function. The number of moments $N = 5, 7, 9$

accuracy than the $P_N$ closure. Moreover, in the optically thick regime, all the closures perform well. It is also observed that the ML hyperbolic closure model with bounded eigenvalues generally has better accuracy than the other two ML closures.

In Fig. 6a, we present the $L^2$ errors as a function of time for the solutions of the three hyperbolic ML moment closure systems and the solution generated by the RTE in the optically thin regime ($\sigma_s = \sigma_a = 1$). We observe that the three hyperbolic closures generate good predictions in the long time simulation up to $t = 10$. Moreover, the eigenvalue based ML

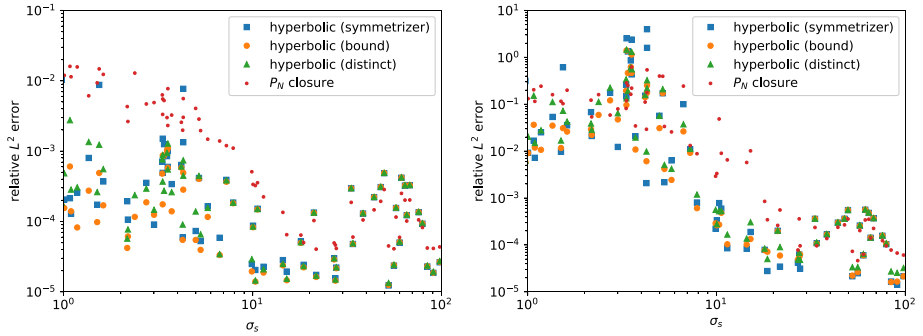



**(a)** $m_0$ at $t = 0.5$ **(b)** $m_1$ at $t = 0.5$




**(c)** $m_0$ at $t = 1$ **(d)** $m_1$ at $t = 1$

**Fig. 4** Example 5.1: constant scattering and absorption coefficients, optically thin regime ($\sigma_s = \sigma_a = 1$), $N = 6$, $t = 0.5$ and $t = 1$

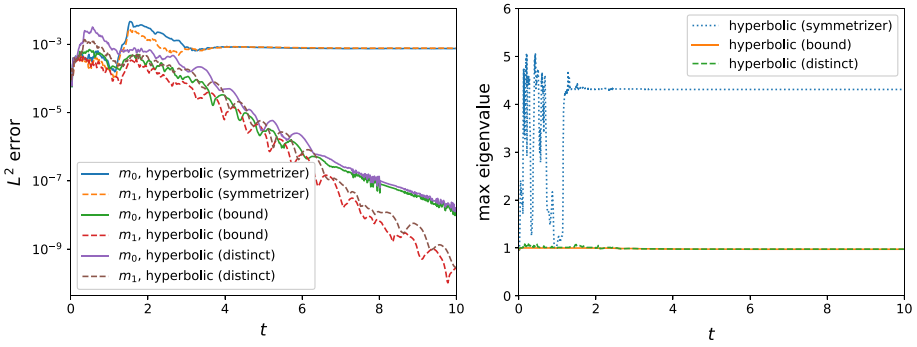



**(a)** $m_0$ at $t = 1$ **(b)** $m_1$ at $t = 1$

**Fig. 5** Example 5.1: constant scattering and absorption coefficients, $N = 6$ and $t = 1$

hyperbolic closure models are more accurate than the symmetrizer based model in [24]. This is probably due to the fact that there is only 4 degrees of freedom in [24]. In contrast, the current eigenvalue based approach makes full use of all the degrees of freedom, which results in better approximation results.

We also display the maximum eigenvalues of the three hyperbolic ML closure models at all the grid points during the time evolution in Fig. 6b. It is observed that the eigenvalues are



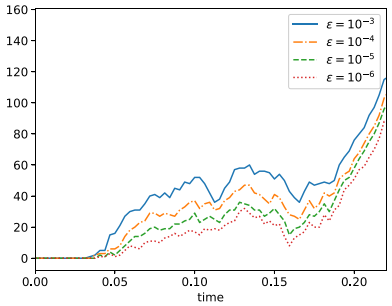

**(a)** $L^2$ errors of $m_0$ and $m_1$ **(b)** maximum eigenvalues

**Fig. 6** Example 5.1: constant scattering and absorption coefficients, $N = 6$

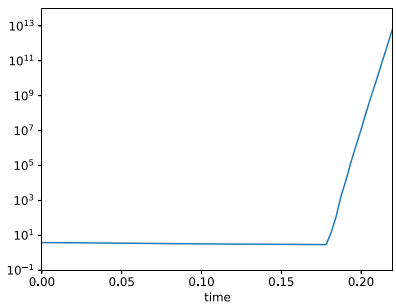
always real numbers, which validates the hyperbolicity feature of the closure models. Moreover, the ML closure with bounded eigenvalues always has physical characteristic speeds bounded by 1. For the closure with distinct eigenvalues, it is interesting to see that the model has the physical characteristic speeds for most of the time although this constrain is not enforced explicitly. The largest eigenvalues of this model during the time evolution is 1.12, which is slightly larger than 1. As a comparision, the symmetrizer based closure in [24] usually violates the physical characteristic speeds, which can be as large as 5.05. The physical characteristic speed of the current ML closure model results in larger time step size in the numerical simulations and thus less computational cost. Moreover, to determine the time step size in the symmetrized based ML model [24] during the time evolution based on the CFL condition, it is required to first compute the coefficient matrix for the closure models and then compute the maximum eigenvalues, which results in additional computational cost. Therefore, the current two ML closure models are better than the symmetrizer based model in [24] in terms of the efficiency.

Next, we discuss the instability issue of the hyperbolic ML closure with bounded eigenvalues. The two eigenvalues get too close for small numbers of moments ($N = 3, 4, 5$), which behaves as if the system is weakly hyperbolic. We simulate the ML closure model with bounded eigenvalues with $N = 3$ and $N = 5$ in the optically thin regime ($\sigma_s = \sigma_a = 1$). The numerical solutions blow up at $t = 0.18$ for $N = 3$ and $t = 1.25$ for $N = 5$, see Fig. 7b, d for the $L^\infty$ norm of the numerical solutions during the time evolution. As a comparison, the solution stays stable for $N = 7$, see Fig. 7f. To investigate this phenomenon in detail, in each time step, we compute the eigenvalues at each grid point, and compute the number of grid points with two eigenvalues which are closer than a given thresholds $\varepsilon$. The number of grid points with close eigenvalues with different thresholds in the time evolution are presented in Fig. 7a, c. From the figure, we observe that there are no grid points with close eigenvalues in the beginning. As time evolves, more grid points with non-distinct eigenvalues appear for $N = 3$ and $N = 5$. For $N = 7$, there only exists a couple of grid points with the thresholds $10^{-3}$ and $10^{-4}$ and no grid points with the thresholds $10^{-5}$ and $10^{-6}$. This does not affect the numerical stability of the simulation.

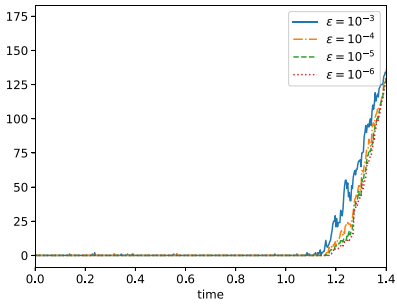
We also observe numerical instability in the hyperbolic ML closure model with distinct eigenvalues for some parameters. The model is numerically stable for $N \geq 6$ but numerically unstable for $N = 3, 4$, and 5 in the optically thin regime. We show the distributions of the training data and the numerical solution during the time evolution of the ML closure model



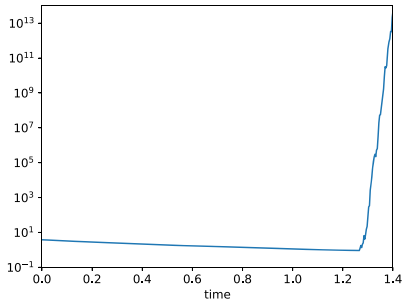

**(a)** number of grid points with close eigenvalues, $N = 3$



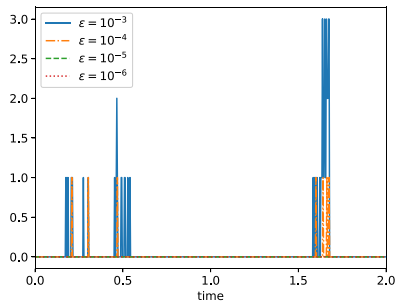

**(b)** $L^\infty$ norm of numerical solution, $N = 3$



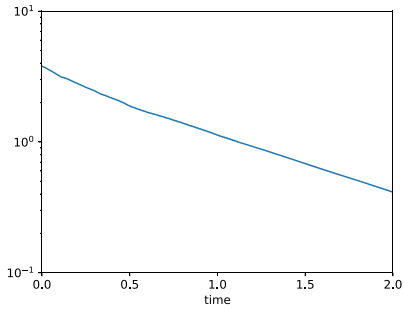

**(c)** number of grid points with close eigenvalues, $N = 5$




**(d)** $L^\infty$ norm of numerical solution, $N = 5$




**(e)** number of grid points with close eigenvalues, $N = 7$

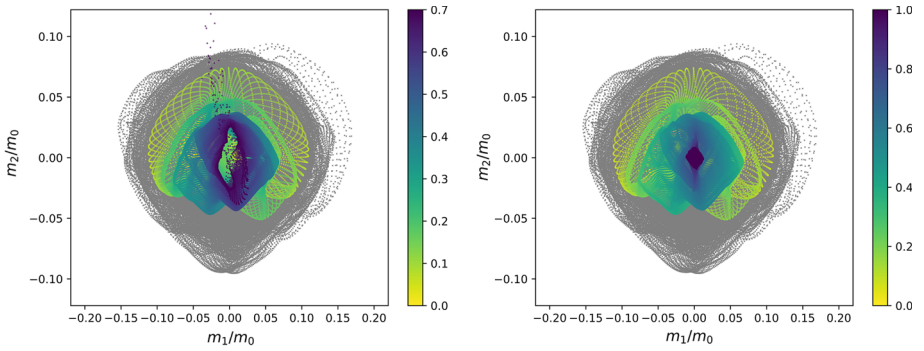



**(f)** $L^\infty$ norm of numerical solution, $N = 7$

**Fig. 7** Example 5.1: constant scattering and absorption coefficients. Here, we use the first neural network architecture in Fig. 1. The number of grid points with imaginary eigenvalues and $L^\infty$ norm of numerical solutions during the time evolution in the optically thin regime ($\sigma_s = \sigma_a = 1$) with $N = 3, 5, 7$

with $N = 3$ and $N = 6$ in Fig. 8. At each time step, there is a curve composed of 256 points and the plots represent the evolution of the closed curve where the color denotes the evolution time. It can be seen for the $N = 3$ case, that as the numerical solutions is approaching the steady state, it suddenly undergoes a dramatic change in the dynamics of the solution and then proceeds to run out side of the range of the training data. This in contrast to the case $N = 6$ which is plotted in Fig. 8b, which clearly shows relaxation to the steady state. In the plots, the color bar represents the time of the solution.

To investigate the instability of the ML closure model with distinct eigenvalues further, we check the linear stability of the system numerically. We denote the source term of the




**(a)** $m_2/m_0$ vs $m_1/m_0$ with $N = 3$ **(b)** $m_2/m_0$ vs $m_1/m_0$ with $N = 6$

**Fig. 8** Example 5.1: constant scattering and absorption coefficients, $m_2/m_0$ vs $m_1/m_0$ with $N = 3$ and $N = 6$. Here, we use the second neural network architecture in Fig. 2. At each time step, there is a curve composed of 256 points and the plots represent the evolution of the closed curve where the color denotes the evolution time. For $N = 3$, as the numerical solutions is approaching the steady state, it suddenly has a dramatic change in dynamics of the solution and then proceeds to run outside of the range of the training data. This in contrast to the case $N = 6$ which is plotted, which clearly shows relaxation to the steady state. The grey points denote the training data and the colorful points denote the numerical solutions solving from the ML moment closure system

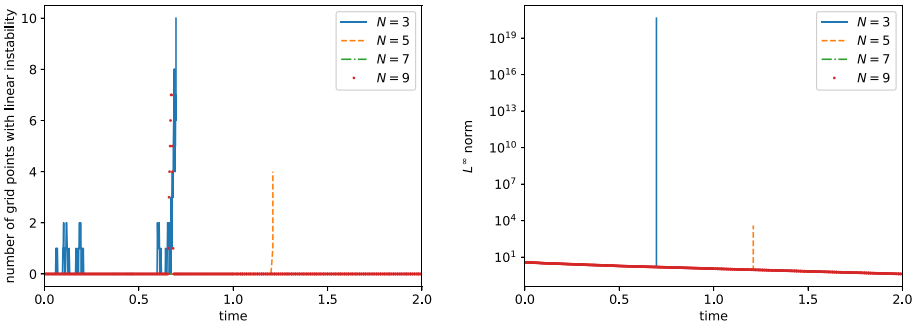
closure model in (3.5) by $S = (-\sigma_a m_0, -(\sigma_s + \sigma_a)m_1, \ldots, -(\sigma_s + \sigma_a)m_N)$. Then, the Jacobian matrix of the source term is $S_U = \text{diag}(-\sigma_a, -(\sigma_s + \sigma_a), \ldots, -(\sigma_s + \sigma_a))$. The model is called linearly stable if all the eigenvalues of $(i\xi A + S_U)$ have non-positive real part for any $\xi \in \mathbb{R}$. Here, $A$ is the coefficient matrix of the closure system given in (3.7) and $i$ is the imaginary unit. Linear stability is essential for the closure system to generate stable results in long time simulations [49]. The symmetrizer based hyperbolic ML moment closure model in [24] satisfies this stability condition. We test for linear stability numerically, by taking $\xi = -100, -99, \ldots, 99, 100$, and computing the eigenvalues of $(i\xi A + S_U)$ at all grid points. The number of grid points with eigenvalues with positive real part and the $L^\infty$-norm of $m_0$ during the time evolution is shown in Fig. 9. It is observed that for $N = 3$ and 5, the solution blows up when the grid points with linear instability appear. This indicates that the loss of linear stability probably results in the blow up of the numerical solutions. It is also interesting to see that for $N = 9$, the model generates stable solution; however, there also exists several grid points with linear instability when the time is around 0.7 and the model returns to stability in the time afterwards. How to stabilize the closure system, while maintaining training accuracy, is a topic to be investigated in the future.

***Example 5.2*** (*Gaussian source problem*) In this example, we investigate the RTE with the initial condition to be a Gaussian distribution in the physical domain:

$$f_0(x, v) = \frac{c_1}{(2\pi\theta)^{1/2}} \exp\left(-\frac{(x - x_0)^2}{2\theta}\right) + c_2. \tag{5.1}$$

In this test, we take $c_1 = 0.5$, $c_2 = 2.5$, $x_0 = 0.5$ and $\theta = 0.01$. We note that this problem is named the Gaussian source problem in the literature [14, 16].

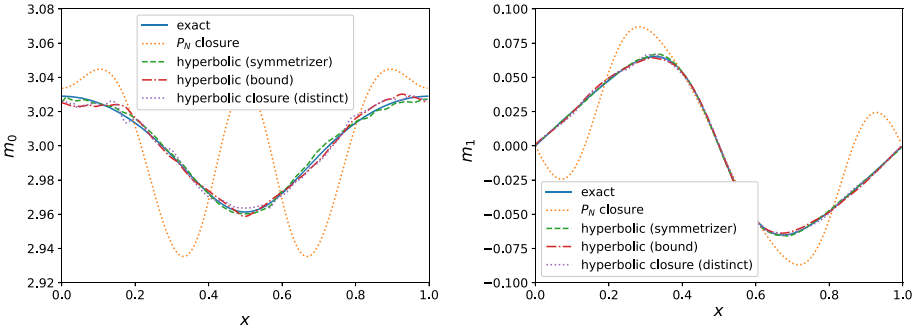
In Fig. 10, we present the results obtained using various closure models. Here, we take $\sigma_s = 1$ and $\sigma_a = 0$. We observe good agreement between the three ML closure models and the kinetic model, while the $P_N$ model has large deviations from the kinetic model.




**(a)** number of grid points with linear instability in the time evolution

**(b)** $L^\infty$-norm of $m_0$ in the time evolution

**Fig. 9** Example 5.1: constant scattering and absorption coefficients




**(a)** $m_0$ at $t = 1$

**(b)** $m_1$ at $t = 1$

**Fig. 10** Example 5.2: Gaussian source problem, $N = 6$ and $t = 1$

These results show the good generalizability of our ML closure models. Moreover, the three hyperbolic ML models have the same level of accuracy in this test.

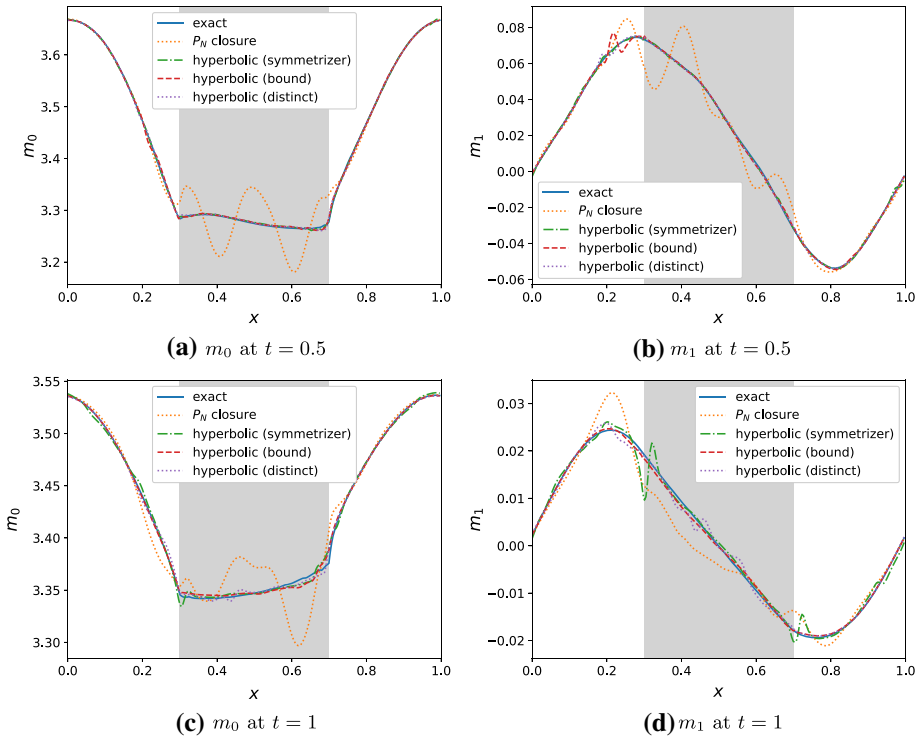
***Example 5.3*** (*two-material problem*) The two-material problem models a domain with a discontinuous material cross section [31]. In our problem setup, there exist two discontinuities $0 < x_1 < x_2 < 1$ in the domain, and $\sigma_s$ and $\sigma_a$ are piecewise constant functions:

$$\sigma_s(x) = \begin{cases} \sigma_{s1}, & x_1 < x < x_2, \\ \sigma_{s2}, & 0 \le x < x_1 \text{ or } x_2 \le x < 1. \end{cases}$$

and

$$\sigma_a(x) = \begin{cases} \sigma_{a1}, & x_1 < x < x_2, \\ \sigma_{a2}, & 0 \le x < x_1 \text{ or } x_2 \le x < 1. \end{cases}$$

Specifically, we take $x_1 = 0.3$, $x_2 = 0.7$, $\sigma_{s1} = 1$, $\sigma_{s2} = 10$ and $\sigma_{a1} = \sigma_{a2} = 0$. The numerical results are shown in Fig. 11. The gray part is in the optically thin regime and the other part is in the intermediate regime. We observe that our current closure model agrees well with the kinetic solution over the whole domain at both $t = 0.5$ and $t = 1$. We note that this is in contrast to the $P_N$ closure, which has large deviations from the kinetic solution



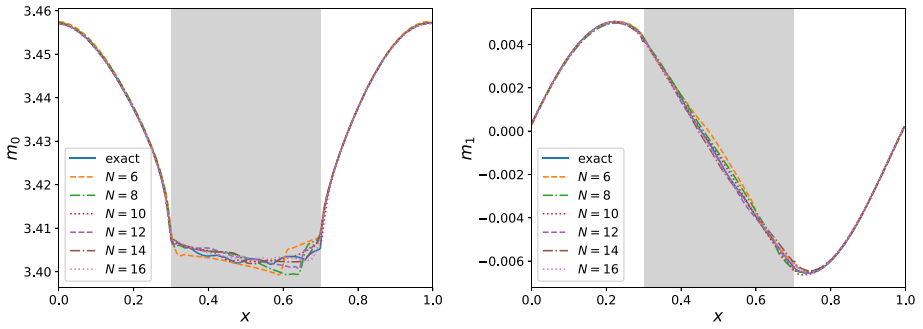

**(a)** $m_0$ at $t = 0.5$ **(b)** $m_1$ at $t = 0.5$

**(c)** $m_0$ at $t = 1$ **(d)** $m_1$ at $t = 1$

**Fig. 11** Example 5.3: two-material problem. Numerical solutions of $m_0$ and $m_1$ at $t = 0.5$ and $t = 1$ with $N = 6$. The gray part in the middle is in the optically thin regime and the other part is in the intermediate regime

in the optically thin portion of the domain, see Fig. 11. Moreover, the two eigenvalue based hyperbolic closures perform better than the closure in [24] which has some overshoot near the discontinuities, see Fig. 11d.

In Fig. 12, we numerically investigate the convergence of the ML closure model with bounded eigenvalues to the kinetic model as the number of moments increases. We take the number of moment to be $N = 6, 8, 10, 12, 14, 16$. In Table 1, we present the relative $L^2$ errors of $m_0$ and $m_1$ for the same numerical example. We observe that the error between the solution to the ML closure model and the solution to the kinetic equation becomes smaller with an increasing number of moments. This numerically indicates that the ML closure model converges to the kinetic model as the number of moments increases. It is worth noting that the saturation in convergence seen in Table 1 is of the same order as the training error in the ML closure model.




**(a)** $m_0$ at $t = 2$ **(b)** $m_1$ at $t = 2$

**Fig. 12** Example 5.3: two-material problem, convergence with respect to number of moments, the ML closure model with bounded eigenvalues. Numerical solutions of $m_0$ at $t = 2$ with $N = 6, 8, 10, 12, 14, 16$. The gray part in the middle is in the optically thin regime and the other part is in the intermediate regime

**Table 1** Example 5.3: two-material problem, convergence with respect to number of moments, the ML closure model with bounded eigenvalues

| $N$ | Relative $L^2$ error of $m_0$ | Relative $L^2$ error of $m_1$ |
|---|---|---|
| 6 | 5.79e−4 | 7.84e−2 |
| 8 | 3.78e−4 | 5.69e−2 |
| 10 | 3.67e−4 | 2.93e−2 |
| 12 | 3.66e−4 | 3.64e−2 |
| 14 | 1.98e−4 | 2.94e−2 |
| 16 | 1.66e−4 | 2.21e−2 |

The relative $L^2$ errors of the numerical solutions of $m_0$ and $m_1$ at $t = 2$ with $N = 6, 8, 10, 12, 14, 16$

## 6 Conclusion

In this paper, we propose a new method to enforce the hyperbolicity of a ML closure model. Motivated by the observation that the coefficient matrix of the closure system is a lower Hessenberg matrix, we relate its eigenvalues to the roots of an associated polynomial. We design two new neural network architectures based on this relation. The ML closure model resulting from the first neural network is weakly hyperbolic and guarantees the physical characteristic speeds, i.e. the eigenvalues lie in the range of the interval $[-1, 1]$. The second model is strictly hyperbolic, but does not guarantee the boundedness of the eigenvalues, although in practice the eigenvalues lie nearly within the physical range. Having the physical characteristic speeds saves substantial computational expenses when numerically solving the closure system by allowing for a larger time step size compared to [24]. Several benchmark tests including the Gaussian source problem and the two-material problem show the good accuracy and generalizability of our hyperbolic ML closure model. Nevertheless, there exists some numerical instability for the current model when a small number of moments are used. We will try to fix this problem in the future work.

**Acknowledgements** We thank Michael M. Crockatt in Sandia National Laboratories for providing numerical solver for the radiative transfer equation. We acknowledge the High Performance Computing Center (HPCC) at Michigan State University for providing computational resources that have contributed to the research results

reported within this paper. JH would like to thank Professor Wen-An Yong in Tsinghua University for many fruitful discussions. This work has been assigned a document release number LA-UR-21-28626.

**Data Availability** Enquiries about data availability should be directed to the authors.

## Declarations

**Competing interests** The authors have not disclosed any competing interests.

## Appendix A: Collections of Proofs

In this appendix, we collect some lemma and proofs. We start with a lemma which characterize the eigenspace of unreduced lower Hessenberg matrix.

**Lemma A.1** *For an unreduced lower Hessenberg matrix $H = (h_{ij})_{n\times n}$, the geometric multiplicity of any eigenvalue $\lambda$ is 1 and the corresponding eigenvector is $(q_0(\lambda), q_1(\lambda), \ldots, q_{n-1}(\lambda))^T$. Here $\{q_i\}_{0\le i\le n-1}$ is the associated polynomial sequence defined in (2.1).*

***Proof*** By Definition 2.1, we have that $h_{ij} = 0$ for $j > i+1$ and $h_{i,i+1} \neq 0$ for $i = 1, \ldots, n-1$. Let $r = (r_1, r_2, \ldots, r_n)$ be an eigenvector associated with $\lambda$. We write $Ar = \lambda r$ as an equivalent component-wise formulation:

$$\sum_{j=1}^{i} h_{ij} r_j + h_{i,i+1} r_{i+1} = \lambda r_i, \quad i = 1, \ldots, n-1, \tag{A.1}$$

and

$$\sum_{j=1}^{n} h_{nj} r_j = \lambda r_n. \tag{A.2}$$

Here we used the fact that $h_{ij} = 0$ for $j > i+1$. Since $h_{i,i+1} \neq 0$ for $i = 1, \ldots, n-1$, (A.1) is equivalent to

$$r_{i+1} = \frac{1}{h_{i,i+1}} \left( \lambda r_i - \sum_{j=1}^{i} h_{ij} r_j \right), \quad i = 1, \ldots, n-1 \tag{A.3}$$

From (A.3), we deduce that $r_1 \neq 0$, otherwise $r_2 = \cdots = r_n = 0$. Moreover, $r_i$ for $i = 2, \ldots, n$ are uniquely determined by $r_1$. Therefore, the geometric multiplicity of $\lambda$ is 1. Moreover, without loss of generality, we take $r_1 = 1$. In this case, $r$ is exactly the same with $(q_0(\lambda), q_1(\lambda), \ldots, q_{n-1}(\lambda))^T$. Here $\{q_i\}_{0\le i\le n-1}$ is the associated polynomial sequence defined in (2.1). □

**Lemma A.2** *Let $H = (h_{ij})_{n\times n}$ be an unreduced lower Hessenberg matrix and $\{q_i\}_{0\le i\le n}$ is the associated polynomial sequence with $H$. If $\lambda$ is an eigenvalue of $H$, then $\lambda$ is a root of $q_n$.*

***Proof*** From Lemma A.1, we have the geometric multiplicity of $\lambda$ is 1 and the corresponding eigenvector $\boldsymbol{q}_{n-1}(\lambda) = (q_0(\lambda), q_1(\lambda), \ldots, q_{n-1}(\lambda))^T$, i.e. $H\boldsymbol{q}_{n-1}(\lambda) = \lambda \boldsymbol{q}_{n-1}(\lambda)$. Plugging $\lambda$ into (2.2), we immediately have $q_n(\lambda) = 0$, i.e., $\lambda$ is a root of $q_n$. □

### A.1: Proof of Theorem 2.4

***Proof*** We start by proving that condition 1 and condition 2 are equivalent. First, it is easy to see that condition 2 implies condition 1. We only need to prove that condition 1 implies condition 2. Since $A$ is real diagonalizable, all the eigenvalues of $A$ are real. Moreover, for any eigenvalue of $A$, the geometric multiplicity is equal to its algebraic multiplicity. By Lemma A.1, the geometric multiplicity of any eigenvalue of an unreduced lower Hessenberg matrix is 1. Therefore, any eigenvalue of $A$ has algebraic multiplicity of 1, i.e. all the eigenvalues of $A$ are distinct.

Next, we prove that the equivalence of condition 2 and condition 3. It is easy to see that, condition 3 implies condition 2 from Theorem 2.3, and condition 2 implies condition 3 from Lemma A.2. This completes the proof. □

### A.2: Proof of Lemma 3.3

***Proof*** We start from the definition of Legendre polynomials by the generating function:

$$\frac{1}{\sqrt{1-2tx+t^2}} = \sum_{n=0}^{\infty} P_n(x)t^n. \tag{A.4}$$

Introduce the variable $s$ such that

$$1 - ts = \sqrt{1-2tx+t^2}, \tag{A.5}$$

which is equivalent to

$$x = \frac{1+t^2-(1-ts)^2}{2t} = s + \frac{t}{2}(1-s^2). \tag{A.6}$$

Therefore, we have

$$\begin{aligned}\sum_{n=0}^{\infty} t^n \int_{-1}^{1} x^m P_n(x)dx &\overset{(A.4)}{=} \int_{-1}^{1} \frac{x^m dx}{\sqrt{1-2tx+t^2}} \\ &\overset{(A.6)}{=} \int_{-1}^{1} \frac{x^m(1-ts)ds}{\sqrt{1-2tx+t^2}} \overset{(A.5)\text{-}(A.6)}{=} \int_{-1}^{1} \left(s + \frac{t}{2}(1-s^2)\right)^m ds.\end{aligned} \tag{A.7}$$

Define

$$a_{m,n} := \int_{-1}^{1} x^m P_n(x)dx. \tag{A.8}$$

By comparing the coefficients of $t^n$ on both sides of (A.7), we find that $a_{m,n} = 0$ if $n > m$ or $m, n$ has different parity. For $n = m - 2k$ for some integer $k \geq 0$, we have

$$a_{m,m-2k} = 2^{2k-m}\binom{m}{2k}\int_{-1}^{1} s^{2k}(1-s^2)^{m-2k}ds = 2^{2k-m}\binom{m}{2k}\int_{0}^{1} 2s^{2k}(1-s^2)^{m-2k}ds \tag{A.9}$$

By introducing the variable $\tau = s^2$ or equivalently $s = \tau^{\frac{1}{2}}$, we have

$$\begin{aligned} a_{m,m-2k} &= 2^{2k-m}\binom{m}{2k}\int_0^1 2s^{2k}(1-s^2)^{m-2k}ds \\ &= 2^{2k-m}\binom{m}{2k}\int_0^1 2\tau^k(1-\tau)^{m-2k}\frac{1}{2}\tau^{-\frac{1}{2}}d\tau \\ &= 2^{2k-m}\binom{m}{2k}\int_0^1 \tau^{k-\frac{1}{2}}(1-\tau)^{m-2k}d\tau \\ &= 2^{2k-m}\binom{m}{2k}\frac{\Gamma(k+\frac{1}{2})\Gamma(m-2k+1)}{\Gamma(m-k+\frac{3}{2})} \\ &= \frac{m!}{2^{k-1}k!(2m-2k+1)!!} \end{aligned} \tag{A.10}$$

where in the fourth equality we used the relation between the gamma function and the beta function:

$$B(x,y) := \int_0^1 t^{x-1}(1-t)^{y-1}dt = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}, \tag{A.11}$$

and in the last equality we used the properties of the gamma function: for any integer $n \geq 0$

$$\Gamma(n) = (n-1)!, \quad \Gamma(n+\frac{1}{2}) = \frac{(2n-1)!!}{2^n}\sqrt{\pi}. \tag{A.12}$$

Lastly, using the orthogonality relation $\int_{-1}^1 P_m(x)P_n(x) = \frac{2}{2m+1}\delta_{m,n}$, we have for any integer $m \geq 0$,

$$x^m = \sum_{k=0}^{\lfloor m/2 \rfloor}\left(\frac{2m-4k+1}{2}\right)a_{m,m-2k}P_{m-2k}(x) = \sum_{k=0}^{\lfloor m/2 \rfloor}\frac{m!(2m-4k+1)}{2^k k!(2m-2k+1)!!}P_{m-2k}(x) \tag{A.13}$$

This completes the proof. □

## References


1. Alldredge, G.W., Hauck, C.D., OLeary, D.P., Tits, A.L.: Adaptive change of basis in entropy-based moment closures for linear kinetic equations. J. Comput. Phys. **258**, 489–508 (2014)
2. Alldredge, G.W., Hauck, C.D., Tits, A.L.: High-order entropy-based closures for linear transport in slab geometry ii: a computational study of the optimization problem. SIAM J. Sci. Comput. **34**(4), B361–B391 (2012)
3. Alldredge, G.W., Li, R., Li, W.: Approximating the $M_2$ method by the extended quadrature method of moments for radiative transfer in slab geometry. Kinetic Relat. Models **9**(2), 237 (2016)
4. Amos, B., Xu, L., Kolter, J.Z.: Input convex neural networks. In: International Conference on Machine Learning, pp. 146–155. PMLR (2017)
5. Bois, L., Franck, E., Navoret, L., Vigon, V.: A neural network closure for the Euler-Poisson system based on kinetic simulations. Preprint arXiv:2011.06242 (2020)
6. Brunton, S.L., Proctor, J.L., Kutz, J.N.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **113**(15), 3932–3937 (2016)
7. Cai, Z., Fan, Y., Li, R.: Globally hyperbolic regularization of Grad's moment system in one-dimensional space. Commun. Math. Sci. **11**(2), 547–571 (2013)
8. Cai, Z., Fan, Y., Li, R.: Globally hyperbolic regularization of Grad's moment system. Commun. Pure Appl. Math. **67**(3), 464–518 (2014)

9. Cai, Z., Fan, Y., Li, R.: On hyperbolicity of 13-moment system. Kinetic Relat. Models **7**(3), 415 (2014)
10. Chandrasekhar, S.: On the radiative equilibrium of a stellar atmosphere. Astrophys. J. **99**, 180 (1944)
11. Crockatt, M.M., Christlieb, A.J., Garrett, C.K., Hauck, C.D.: An arbitrary-order, fully implicit, hybrid kinetic solver for linear radiative transport using integral deferred correction. J. Comput. Phys. **346**, 212–241 (2017)
12. Crockatt, M.M., Christlieb, A.J., Garrett, C.K., Hauck, C.D.: Hybrid methods for radiation transport using diagonally implicit runge-kutta and space-time discontinuous galerkin time integration. J. Comput. Phys. **376**, 455–477 (2019)
13. Elouafi, M., Hadj, A.D.A.: A recursion formula for the characteristic polynomial of hessenberg matrices. Appl. Math. Comput. **208**(1), 177–179 (2009)
14. Fan, Y., Li, R., Zheng, L.: A nonlinear hyperbolic model for radiative transfer equation in slab geometry. SIAM J. Appl. Math. **80**(6), 2388–2419 (2020)
15. Fan, Y., Li, R., Zheng, L.: A nonlinear moment model for radiative transfer equation in slab geometry. J. Comput. Phys. **404**, 109128 (2020)
16. Frank, M., Hauck, C.D., Olbrant, E.: Perturbed, entropy-based closure for radiative transfer. Preprint arXiv:1208.0772 (2012)
17. Grad, H.: On the kinetic theory of rarefied gases. Commun. Pure Appl. Math. **2**(4), 331–407 (1949)
18. Han, J., Jentzen, A., Weinan, E.: Solving high-dimensional partial differential equations using deep learning. Proc. Natl. Acad. Sci. **115**(34), 8505–8510 (2018)
19. Han, J., Ma, C., Ma, Z., Weinan, E.: Uniformly accurate machine learning-based hydrodynamic models for kinetic equations. Proc. Natl. Acad. Sci. **116**(44), 21983–21991 (2019)
20. Hauck, C., McClarren, R.: Positive $P_N$ closures. SIAM J. Sci. Comput. **32**(5), 2603–2626 (2010)
21. Hauck, C.D.: High-order entropy-based closures for linear transport in slab geometry. Commun. Math. Sci. **9**(1), 187–205 (2011)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
23. Huang, J., Cheng, Y., Christlieb, A.J., Roberts, L.F.: Machine learning moment closure models for the radiative transfer equation I: directly learning a gradient based closure. J. Comput. Phys. **453**, 110941 (2022)
24. Huang, J., Cheng, Y., Christlieb, A.J., Roberts, L.F., Yong, W.-A.: Machine learning moment closure models for the radiative transfer equation II: enforcing global hyperbolicity in gradient based closures. Preprint arXiv:2105.14410 (2021)
25. Huang, J., Ma, Z., Zhou, Y., Yong, W.-A.: Learning thermodynamically stable and Galilean invariant partial differential equations for non-equilibrium flows. J. Non-Equilibrium Thermodyn. **46**, 355–70 (2021)
26. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456. PMLR (2015)
27. Jiang, G.-S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**(1), 202–228 (1996)
28. Klose, A.D., Netz, U., Beuthan, J., Hielscher, A.H.: Optical tomography using the time-independent equation of radiative transfer-part 1: forward model. J. Quant. Spectrosc. Radiat. Transf. **72**(5), 691–713 (2002)
29. Koch, R., Becker, R.: Evaluation of quadrature schemes for the discrete ordinates method. J. Quant. Spectrosc. Radiat. Transf. **84**(4), 423–435 (2004)
30. Laboure, V.M., McClarren, R.G., Hauck, C.D.: Implicit filtered $P_N$ for high-energy density thermal radiation transport using discontinuous galerkin finite elements. J. Comput. Phys. **321**, 624–643 (2016)
31. Larsen, E., Morel, J.: Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes ii. J. Comput. Phys. **83**(1) (1989)
32. Levermore, C.: Relating eddington factors to flux limiters. J. Quant. Spectrosc. Radiat. Transf. **31**(2), 149–160 (1984)
33. Levermore, C.D.: Moment closure hierarchies for kinetic theories. J. Stat. Phys. **83**(5), 1021–1065 (1996)
34. Li, R., Li, W., Zheng, L.: Direct flux gradient approximation to close moment model for kinetic equations. Preprint arXiv:2102.07641 (2021)
35. Ma, C., Zhu, B., Xu, X.-Q., Wang, W.: Machine learning surrogate models for Landau fluid closure. Phys. Plasmas **27**(4), 042502 (2020)
36. Maulik, R., Garland, N.A., Burby, J.W., Tang, X.-Z., Balaprakash, P.: Neural network representability of fully ionized plasma fluid model closures. Phys. Plasmas **27**(7), 072106 (2020)
37. McClarren, R.G., Hauck, C.D.: Robust and accurate filtered spherical harmonics expansions for radiative transfer. J. Comput. Phys. **229**(16), 5597–5614 (2010)

38. Murchikova, E., Abdikamalov, E., Urbatsch, T.: Analytic closures for M1 neutrino transport. Mon. Not. R. Astron. Soc. **469**(2), 1725–1737 (2017)
39. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. Preprint arXiv:1912.01703 (2019)
40. Pomraning, G.C.: The Equations of Radiation Hydrodynamics. Pergamon Press, Oxford (1973)
41. Porteous, W.A., Laiu, M.P., Hauck, C.D.: Data-driven, structure-preserving approximations to entropy-based moment closures for kinetic equations. Preprint arXiv:2106.08973 (2021)
42. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 686–707 (2019)
43. Schotthöfer, S., Xiao, T., Frank, M., Hauck, C.D.: A structure-preserving surrogate model for the closure of the moment system of the Boltzmann equation using convex deep neural networks. Preprint arXiv:2106.09445 (2021)
44. Scoggins, J.B., Han, J., Massot, M.: Machine learning moment closures for accurate and efficient simulation of polydisperse evaporating sprays. In: AIAA Scitech 2021 Forum, p. 1786 (2021)
45. Serre, D.: Systems of Conservation Laws 1: Hyperbolicity, Entropies, Shock Waves. Cambridge University Press (1999)
46. Shu, C.-W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. **77**(2), 439–471 (1988)
47. Szeg, G.: Orthogonal Polynomials, vol. 23. American Mathematical Soc. (1939)
48. Wang, L., Xu, X., Zhu, B., Ma, C., Lei, Y.-A.: Deep learning surrogate model for kinetic Landau-fluid closure with collision. AIP Adv. **10**(7), 075108 (2020)
49. Yong, W.-A.: Basic aspects of hyperbolic relaxation systems. In: Advances in the Theory of Shock Waves, pp. 259–305. Springer (2001)
50. Zhu, Y., Hong, L., Yang, Z., Yong, W.-A.: Conservation-dissipation formalism of irreversible thermodynamics. J. Non-Equilib. Thermodyn. **40**(2), 67–74 (2015)