A Mixture of Experts Approach to Solving the Euler Equations

Vedula, Keshav
kvedula@seas.upenn.edu

Sieg, Phil
philsieg@seas.upenn.edu

May 10, 2022

Department of Mechanical Engineering and Applied Mechanics,
University of Pennsylvania,
Philadelphia, PA 19104, USA

Abstract

In this work, we aim to solve the Euler equations using physics-informed neural networks (PINNs) with multiple model architectures and training procedures. Specifically, we implement a Mixture of Experts (MoE) version of the PINNs approach to solve the steady two-dimensional Euler equations for the oblique shock wave problem. Compared to the traditional PINNs approach, it is found that the MoE architecture offers a tremendous improvement in relative accuracy (reduced relative $\mathbb{L}_2$-error from 17.6% to 0.6%) for the same number of model parameters and fewer training iterations. Further marginal improvement is achieved by implementing a causal training procedure in the direction of the body surface. In general, we observe that the MoE architecture reduces the model's sensitivity to hyperparameter tuning including the relative weighting of the various loss terms. The authors interpret this result as an improvement in the robustness of the PINNs approach for solving hyperbolic PDEs when coupled with the MoE architecture.

1 Introduction

1.1 Motivation

The Euler equations are fundamental to the study of high-speed aerodynamics, as they form the simplest model of fluid flow that allows for distinctly compressible phenomena including shock waves. Indeed, they are often considered a numerical testing ground and are key to the development of numerical methods for compressible flows. The accurate prediction of shock formation, intensity, and interactions is critical for estimating the wall fluxes (heat flux and shear stress) experienced by supersonic vehicles, in part because flow properties can change by an order of magnitude across a shock. Importantly, modeling shock behavior involves a prediction of shock location, shape, and strength.

The resolution of true discontinuities in the flow domain presents unique challenges for traditional numerical methods related to flux reconstruction at cell interfaces. It also imposes strict stability constraints which increase the computational cost of such methods. The present work aims to follows more recent data-driven approaches to solve the Euler equations, and leverages a Mixture of Experts architecture to resolve discontinuities. While computational savings are not expected for foundational test cases, it is hoped that the generality of this approach will inspire more flexible and efficient data-driven methods for solving compressible flow problems of engineering interest.

1.2 Existing Methods

The existing approaches for solving the Euler equations in any number of spatial dimensions can be partitioned into traditional numerical methods and more recent data-driven approaches described below.

1.2.1 Numerical Methods

The dominant approach to numerically solving the Euler equations is the finite volume (FV) method originally proposed by Patankar in 1980 [1]. Research since then has focused on tailoring this approach to compressible flows where the flux reconstruction schemes are especially important [2, 3]. This challenge arises as the FV framework assumes constant flow properties within each discrete cell, and thus the flux function at cell interfaces is not uniquely defined. This naturally gives rise to a Riemann problem at each cell interface, for which many exact and approximate solvers have been proposed [4]. Flux reconstruction methods strongly depend on the quality of the solution Riemann problem underpinning them, and thus can be difficult to accurately extend to problems with diffusion and reaction processes.

A more recent line of work is based on the discontinuous Galerkin (DG) framework, an adaptation of the finite element method [4, 5]. This class of methods is able to achieve higher orders of accuracy compared to their finite volume counterparts by virtue of allowing for intra-cell variation of flow properties. Nonetheless, the central concern with both FV and DG methods is the development of robust, high-order, non-dispersive flux reconstruction schemes. It is known that there is a natural trade-off between order of accuracy and dispersion error particularly for linear numerical methods, and thus the resolution of shock waves requires tailored non-linear schemes in practice [4]. A key limitation in the numerical solution of compressible flow problems is computational expense when trying to resolve shock phenomena, as explicit schemes have strict stability requirements and implicit schemes require several matrix inversions.

1.2.2 Data Driven Methods

A recent line of research has focused on using physics informed neural networks (PINNs) to solve PDEs [6, 7, 8, 9, 10]. The traditional PINNs approach seeks to approximate the solution to the PDE with a deep neural network, and deploys a composite loss function consisting of a term that enforces the boundary conditions at collocation points along the boundaries, and a term that forces the residual of the PDE to zero at a series of collocation points in the domain. This method leverages the universal approximation property of neural networks with non-linear activation functions.

Mao et al. employed this traditional approach to solve the one-dimensional unsteady and two-dimensional steady Euler equations [7]. For the latter case, they considered the forward oblique shock wave problem, and demonstrated the ability of the neural network to accurately resolve the shock. Notably though, this result was dependent on an additional term in the loss function which penalizes gradients of the primitive flow variables along the boundaries of the domain. The authors also needed a precise relative weighting of the three components in the loss function and a two-part training procedure to achieve this reasonable degree of accuracy.

Related work by Diab et al. explored Riemann type problems containing shock waves which are generally difficult to solve with numerical methods [8]. Using PINNs with residual-based adaptive refinement (RAR), they show that they can solve problems with simultaneously very low diffusion and dispersion error.

For these traditional PINNs approaches, it has been observed in the literature that PINNs struggle heavily with multi-scale chaotic or turbulent behaviour. Wang et al. argue that this is due to the fact that traditional PINNs formulations do not respect the spatio-temporal causality found in many physical systems [9]. To address this, the authors implement an adaptively weighted loss function which requires that early time behavior is accurately resolved before later time behavior. They find strong performance improvements for the solution of chaotic systems including the Lorenz equations.

A novel line of inquiry has attempted to merge elements of the FV framework with that of PINNs into an approach known as Control Volume PINNs (CV-PINNs). As implemented by Patel et al., this method first involves discretizing the domain into a finite number of non-overlapping elements (as in the FV method), and then assigns the loss function to be the residual of the integral form of the conservation law over the entire domain [10]. In this setting, a neural network is still used to approximate the solution to the PDE. But by specifying the flux function appropriately along domain boundaries, the need for tuning the relative weightings of the various terms in a composite loss function is eliminated. This integral approach also permits a way to directly enforce the entropy inequality and to penalize total variation of the solution, two key properties that are common in traditional numerical schemes. The authors implement a number of regularizers to introduce thermodynamic inductive biases into the model. They conclude that the CV-PINNs approach is particularly well-suited for hyperbolic conservation laws and can be thought of as the data-driven counterpart to the FV method.

An approach that has yet to be merged with the PINNs framework in the solution of the Euler equations is known as the Mixture of Local Experts (MoE). Introduced by Jacobs et al. in 1991, this method seeks to adaptively partition the domain such that each subset is accurately learned by a single "expert" network [11]. Then, a gating network decides which expert to trust given a location in the domain. This architecture has been applied to regression problems and VAE problems, but to the authors' best knowledge has yet to be applied to the data-driven solution to hyperbolic conservation laws such as the Euler equations.

1.3 Objectives

The purpose of the present work is to merge the traditional PINNs framework with the MoE architecture in the solution of the Euler equations for the oblique shock wave problem. Strong performance would demonstrate how the MoE procedure can be used to augment PINNs to handle discontinuities in the solution of hyperbolic PDEs and overcome some of the challenges highlighted above.

The first objective is to reproduce the approach outlined by Mao et al. for the oblique shock problem using just the traditional PINNs approach, and to note the implementation challenges with this method. Following this, we implement a MoE version of the PINNs approach for the same problem in two stages: one with a non-causal training procedure and one with a spatially causal training procedure. The distinction in the training process will suggest whether or not spatially hyperbolic problems also benefit from causality enforced along the hyperbolic direction.

2 Methodology

2.1 Problem Definition

The forward problem of solving the steady Euler equations in two dimensions on a domain Ω can be formulated as follows. The flow state vector $\mathbf{U} = (\rho, \rho u_1, \rho u_2, \rho E)$ satisfying the following PDE

$$\nabla \cdot \mathbf{F}(\mathbf{U}) = 0 \qquad (x, y) \in \Omega \tag{1}$$

is sought for flux function

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 & \mathbf{F}_2 \end{bmatrix} \qquad \mathbf{F}_k = \begin{bmatrix} \rho u_k & \delta_{k1} p + \rho u_1 u_k & \delta_{k2} p + \rho u_2 u_k & p u_k + \rho u_k E \end{bmatrix}^\top \tag{2}$$

where we impose suitable Dirichlet boundary conditions corresponding to the oblique shock wave case: $\mathbf{U}(x, y) = \mathbf{U}_b(x, y)$ for $(x, y) \in \partial\Omega$. Note that ρ is the flow density, u_i denotes the flow velocity in the i^{th} direction, and E denotes the total energy per unit mass. For the present work, we take the flow domain to be

$$\Omega = [0,1] \times [0,1] \tag{3}$$

where the inlet is taken to be the $x = 0$ boundary, the outlet is the $x = 1$ boundary, and the body surface is the $y = 0$ boundary. Thus, for flow deflection angle δ, the boundary conditions are given by

$$x = 0, \text{ or } y = 1: \qquad \rho = \rho^{(1)},\ u_1 = u_1^{(1)},\ u_2 = u_2^{(1)}, E = E^{(1)} \tag{4}$$

$$y = 0: \qquad \rho = \rho^{(2)},\ u_1 = u_1^{(2)},\ u_2 = u_2^{(2)}, E = E^{(2)} \tag{5}$$

where the post-shock conditions denoted by the superscript $^{(2)}$ are determined from the Rankine-Hugoniot jump conditions. Notice that no boundary conditions are imposed on the outlet as the x direction is the spatially hyperbolic direction in the post-shock regime.

2.2 PINNs Approach

The initial method of solution for the steady oblique shock problem was a traditional PINNs architecture. Following the findings from Mao et al., the network was specified to be an MLP consisting 4 hidden layers of 40 neurons each, all using hyperbolic tangent activation functions [7]. The output layer had four nodes, one for each component of the flow state vector. The loss function was formulated as follows:

$$\mathcal{L}(\theta) = \text{MSE}_{bd} + \lambda_1 \cdot \text{MSE}_{res} + \lambda_2 \cdot \text{MSE}_{grad} \tag{6}$$

The first term is a data fit term for the Dirichlet boundary conditions of the PDE.

$$\text{MSE}_{bd} = \frac{1}{N_b} \sum_{i=1}^{N_b} \sum_{i=1}^{4} |U_j^{NN}(x_i^b, y_i^b) - U_j^b(x_i^b, y_i^b)|^2 \tag{7}$$

The second term enforces the structure of the PDE and is based on the residual evaluated at various collocation points within the domain.

$$\text{MSE}_{res} = \frac{1}{N_f} \sum_{i=1}^{N_f} \sum_{i=1}^{4} |f_j(x_i^f, y_i^f)|^2 \tag{8}$$

$$\mathbf{f} = \nabla \cdot \mathbf{F}(\mathbf{U}^{NN}) \tag{9}$$

The final term penalizes the gradient magnitude of all flow variables on the boundaries of the domain.

$$\text{MSE}_{grad} = \frac{1}{N_{bg}} \sum_{i=1}^{N_{bg}} \sum_{i=1}^{4} ||\nabla U_j^{NN}(x_i^{bg}, y_i^{bg})||_2^2 \tag{10}$$

Parameters λ_1 and λ_2 are used to weight the individual loss terms during training as they differ significantly in magnitude. Adjustment of these weights can lead to qualitatively different final solutions based on whether the PDE or the boundary conditions are overemphasized. The model performs best when the PDE residual is given a very small weight $\lambda_1 \approx \frac{1}{100}$, and the boundary gradient term has neutral weight $\lambda_2 \approx 1$.

With the loss function specified, the parameters of the MLP, θ, are then determined via Adam gradient descent, resulting in an approximate solution $\mathbf{U}_{NN}$ for $\mathbf{U}$. The present implementation uses $N_b = N_{bg} = 400$ equi-spaced boundary collocation points, and $N_f = 10,000$ randomly selected collocation in the interior of the domain.

2.3 Mixture of Experts Approach

The second model architecture considered was a MoE approach. This is a supervised learning procedure that utilizes many separate networks. All but one of these networks are referred to as *expert networks.* As the name suggests, the role of these experts is to make accurate predictions over a localized subset of the input space. These experts are local in the sense that their weights are (either approximately or fully) uncoupled from all other experts, and are only allocated to a small region of the input space. Additionally, there is a *gating network* whose role is to take as input a point in the domain and delegate which expert (or combination of experts) will generate the model's final prediction.

Work by Jacobs et al. [11] highlights that prior attempts at a similar model structure lacked success due to issues with the loss formulation. When the gating network is purely cooperative – meaning that the model output is a weighted average over the experts – the loss function give rise to strong coupling between experts. Experts would learn to overcompensate for the poor predictions of their peers, which meant that they could not produce standalone predictions. The authors argue however that a properly trained MoE approach should be able to receive any input and assign a single expert to make a prediction. They introduce a formulation which encourages competition over cooperation, to reduce the aforementioned coupling. In this version, the gating network assigns each expert a probability determining how likely it perceives the expert to provide the most accurate prediction. Next, a selector randomly chooses a single experts prediction weighted by the probabilities from the gating network. Through this, they are able to ensure that the prediction of one expert is independent of those made by other experts for the same input. Figure 1 shows a sample architecture for a model with three experts. It is important to note that the structure of the experts is problem dependant. In our case, we use MLPs with a PINNs inspired loss. Depending on the nature of the problem however, the experts themselves could be of any form that is suitable.

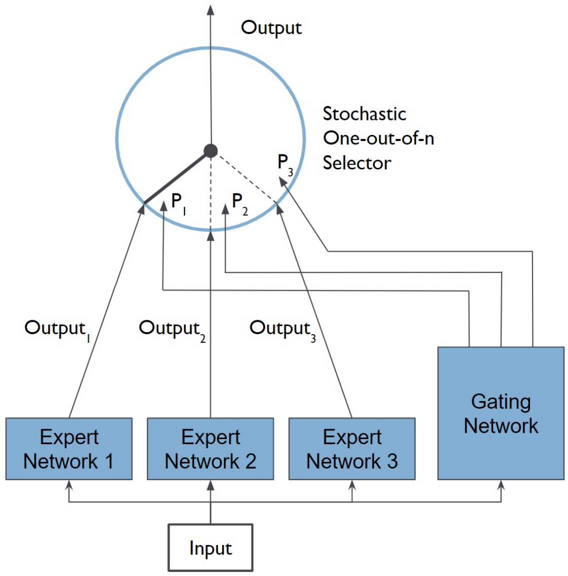


Figure 1: Sample Mixture of Experts Architecture

The two-dimensional steady, oblique shock wave problem naturally lends itself towards a two expert scheme due to the splitting of the domain at the shock interface. This is represented in figure 2 by the dashed line. The role of each expert would be to accurately model the flow properties in either the pre- or post-shock flow. It is then the gating network's duty to learn where the shock occurs, and when to apply each expert. In practice during training, we employ a competitive loss formulation for the boundary conditions. This forces each expert to accurately learn flow properties on its respective side of the shock, as it effectively uncouples the learning of the pre- and post-shock boundary conditions. However, the residual loss inside the domain is computed using a cooperative approach to promote a non-trivial solution in the domain, and to prevent an exact discontinuity from forming wherein automatic differentiation would be undefined. This cooperative approach forces the learned shock to be nearly discontinuous, but formally smooth. By transforming the outputs of the gating network with a temperature hyperparameter to artificially increase the gating network's confidence, the resolution of the shock can approach (but never reach) a true discontinuity. These resolution properties are advantageous because unlike FV methods they do not require arbitrarily fine grid resolution to achieve convergence. Rather, one can reduce the artificial diffusion of the scheme at zero additional cost, and without trading diffusion for dispersion errors.

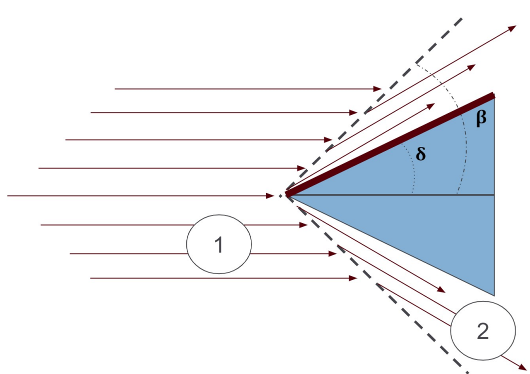


Figure 2: Mixture of Experts Applied to the Oblique Shock Wave Problem

For the present study, it was found that a gating network with 2 hidden layers of 64 neurons each, and 2 expert networks each with 2 hidden layers of 32 neurons were sufficient to reliably predict the shock pattern. Note that in the MoE implementation, the boundary gradient loss term was removed ($\lambda_2 = 0$), as it was found to be not necessary given the performance without it. Interestingly, the PDE residual loss term was much smaller than the boundary condition loss term, and thus setting λ_1 to be in the range $10^2 \leq \lambda_2 \leq 10^6$ proved to give reliable results. It was found that $\lambda_1 = 10^2$ was optimal for the test case described in the following section, but that for stronger shocks a higher weight on the PDE loss term was necessary to enforce the conservation laws more strictly.

2.4 Causal Training

As previously mentioned, the MoE version of PINNs was implemented with and without causal training, a concept which stems from work done by Wang et al. [9]. It has been observed in the literature that PINNs struggle heavily with multi-scale chaotic or turbulent behaviour. Wang et al. argue that this is due to the fact that traditional PINN formulations do not respect spatio-temporal causality found in many physical systems. Not only do PINNs not have this causality embedded in their structure, but the authors show that they are in fact biased towards approximating solutions first at late times. To do this, they attempt to solve the Allen-Cahn equation using a conventional PINN. Following some of their prior work, they are able to solve for the temporal convergence rate ($C(t)$) of the residual loss by examining the Neural Tangent Kernel (NTK). Figure 3 below shows the temporal convergence rate at different iterations of the training.

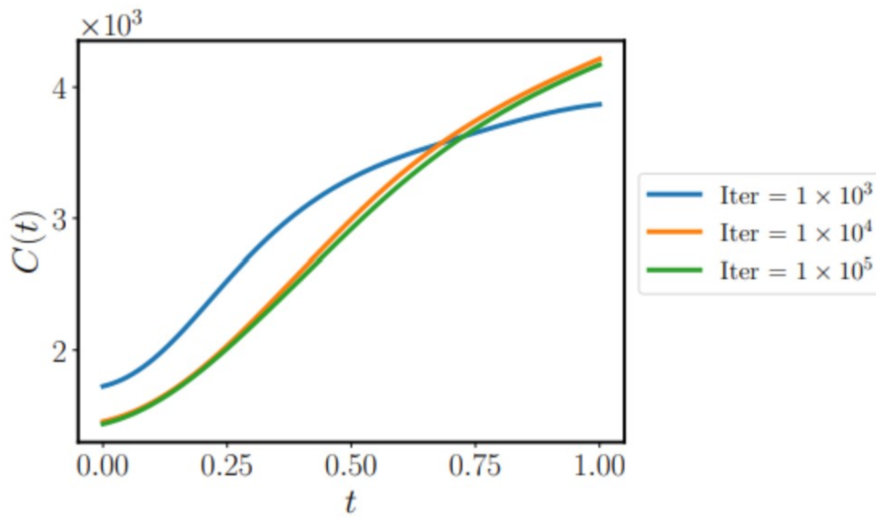


Figure 3: Temporal Convergence Rate of Residual Loss

The convergence rate is much higher at the later time steps, revealing the undesirable implicit bias towards minimizing the residual at larger times. In practice, this means that the model will be prone to first fitting the parameters to minimize the residual at the end of the time domain before working backwards towards the starting time. This is opposed to the direction of causality in physical systems. To tackle this problem, the authors present a simple but effective strategy of embedding causality into the model which involves a reformulation of the loss. The residual loss now is of the following form

$$\mathcal{L}_{res}(\theta) = \frac{1}{N_t} \sum_{i=1}^{N_t} w_i \mathcal{L}_{res}(t_i, \theta) \tag{11}$$

where

$$w_i = \exp\left(-\epsilon \sum_{j=1}^{i-1} \mathcal{L}_{res}(t_j, \theta) \right), \text{ for } i = 2, 3, \ldots, N_t \tag{12}$$

for the case where time t is the hyperbolic direction. Notice that these new weights, w_i will be large only if the residuals prior to t_i are minimized. If they are not properly minimized, w_i will be small and not contribute to the overall residual loss. By not contributing to the loss, the residual at t_i will not affect the model parameters during gradient descent. The parameter ϵ is referred to as a causality parameter and controls the steepness of the weights w_i. In practice, all of the weights are initialized to be zero except at t_1. As the model begins to approximate the initial residuals, the envelope expands and the w_i's begin to grow until eventually, the entire domain is contributing to the loss. Through this simple reformulation, the authors are able to embed causality into the structure of PINNs. They show that with this modified loss function, they are able to solve the Lorentz system, Kuramoto–Sivashinsky equation, and Navier-Stokes equation, all problems that were previously unsolvable with traditional PINNs.

Although the present work deals with a steady flow, since the PDE is hyperbolic both upstream and downstream of the shock, the effective direction of causality is the direction of the flow in both regions. For ease implementation (and generality) however, we treat x – the direction tangent to the body surface – to be the effective direction of causality which becomes an increasingly better approximation for small deflection angles. This leads instead to a residual loss function of the form

$$\mathcal{L}_{res}(\theta) = \frac{1}{N_x} \sum_{i=1}^{N_x} w_i \mathcal{L}_{res}(x_i, y_i; \theta) \tag{13}$$

where

$$w_i = \exp\left(-\epsilon \sum_{j=1}^{i-1} \mathcal{L}_{res}(x_j, y_j; \theta) \right), \text{ for } i = 2, 3, \ldots, N_x \tag{14}$$

This formulation of causal training is implemented for the MoE architecture and the results are compared with that of the non-causal procedure in the following sections.

3 Results & Discussion

3.1 Test Case

The primary test case in this work was that of the steady oblique shock wave from the reference study by Mao et al. [7]. Specifically, we consider a $M^{(1)} = 2.0$ flow with deflection angle $\delta = 10°$. The corresponding boundary conditions in primitive variables are given by

$$M^{(1)} = 2.0,\ \rho^{(1)} = 1.0 \text{ kg/m}^3,\ T^{(1)} = 300 \text{ K},\ \delta = 10° \tag{15}$$

$$M^{(2)} = 1.64,\ \rho^{(2)} = 1.46 \text{ kg/m}^3,\ T^{(2)} = 351.05 \text{ K},\ \beta = 39.31°, \tag{16}$$

These can be readily transformed to boundary conditions on the conserved variables using the ideal thermodynamic equation of state and assuming a compressibility factor of $\gamma = 1.4$. Note that since the bottom boundary of the domain is taken to be the body surface, the perceived angle of the shock wave will be $\beta - \delta = 29.31°$ relative to the $y = 0$ boundary.

3.2 PINNs

For the test case described above, the first approach taken was the traditional PINNs model with 4 hidden layers and 40 neurons per layer. The converged Mach, density, and temperature fields are shown in the top row of figure 4 after 10,000 iterations. Compared to the exact solution in the bottom row of the same figure, notice that the pre-shock and post-shock quantities are approximately correct, but that the shock wave angle decreases and the shock thickness widens as one proceeds downstream. This behavior was not observed in the reference study by Mao et al. and is thought to be a reflection of the sensitivity of the solution to the training procedure and the weights on the component loss terms [7].

Interestingly, one would expect that the shock angle decreases downstream, approaching the Mach angle, for a body of finite length. Thus, the observed results could be explained as an approximate satisfaction of the conservation laws for which the model is unaware that the body surface is semi-infinite. The loss term in eq. (10) is thought to be an ad hoc method of enforcing the semi-infinite constraint in the finite computational domain, but the present work does not find this approach to be effective. The model performance for this test case is summarized in table 1.

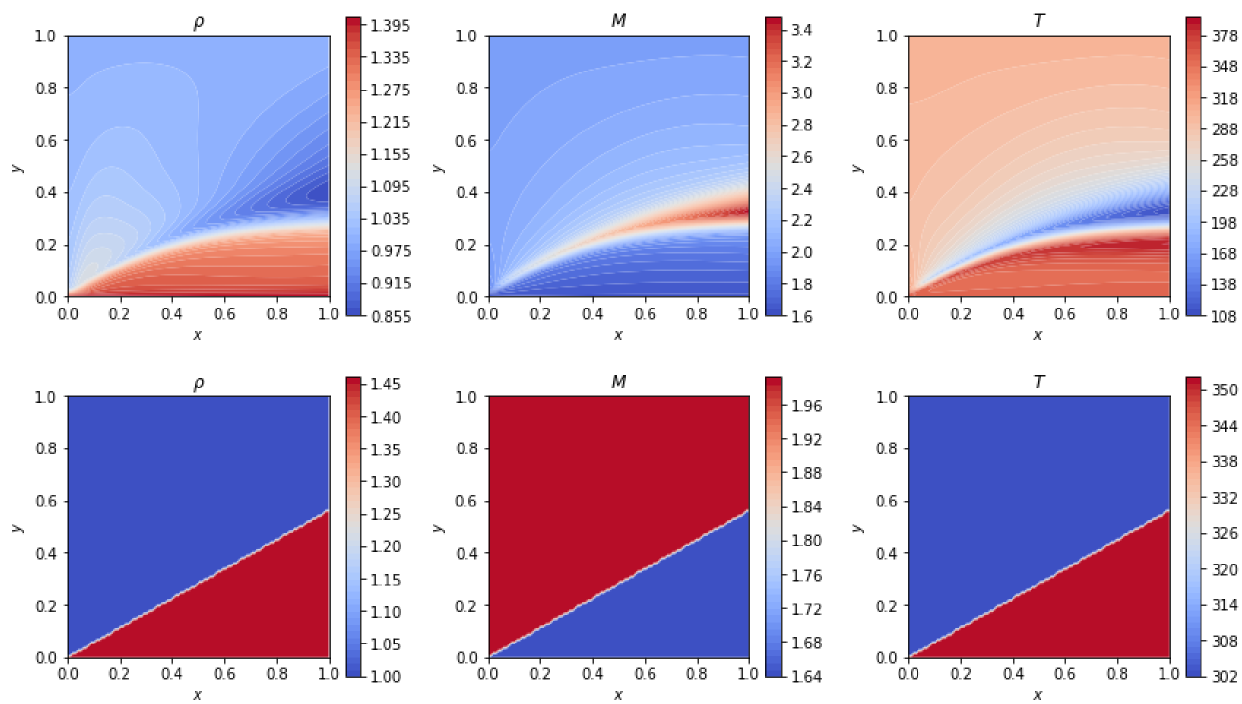


Figure 4: Density, Mach, and Temperature Distributions (Top row: traditional PINNs, bottom row: exact solution)

3.3 Mixture of Experts

The second approach taken was the MoE version of the PINNs model with non-causal training. Here, the two expert networks each have 2 hidden layers of 32 neurons each, and the gating network has 2 hidden layers of 64 neurons each. The corresponding converged Mach, density, and temperature fields are shown in the top row of figure 5 after 400 iterations. Notice the very close agreement between the predicted solution and the exact solution in the pre- and post-shock regimes. The shock angle is also correctly predicted with less than 1° of error.

Figure 6b highlights that the gating network is correctly learning the position of the shock, with the first expert network learning the pre-shock flow and the second expert network learning the post-shock flow.[1] This result was achieved without the need for the loss term in eq. (10), and thus suggests that all the necessary information to solve the problem using the PINNs framework is embedded in the PDE and Dirichlet boundary conditions.

Figure 6a depicts the loss convergence and the relative $\mathbb{L}_2$-error during training. Notice that the relative $\mathbb{L}_2$-error is actually minimized early in training between 200 and 400 iterations, after which it slowly increases. This suggests the possibility of over-training, and hence an early stopping procedure was implemented in practice to resolve this issue. It is worth noting that after 500 iterations, the predicted shock angle begins to decrease slightly while the pre- and post-shock flow Dirichlet conditions are increasingly better satisfied. This indicates that the boundary condition loss term is weighted too heavily, and thus late in training the model attempts to satisfy the boundary conditions by compromising the satisfaction of the conservation laws. It was found that this issue could be resolved for this test case by increasing the weight on the PDE residual term to $\lambda_1 = 10^6$, however this dramatically slowed the rate of loss convergence such that at least 2,000 iterations were necessary. The model performance for this test case is again summarized in table 1.

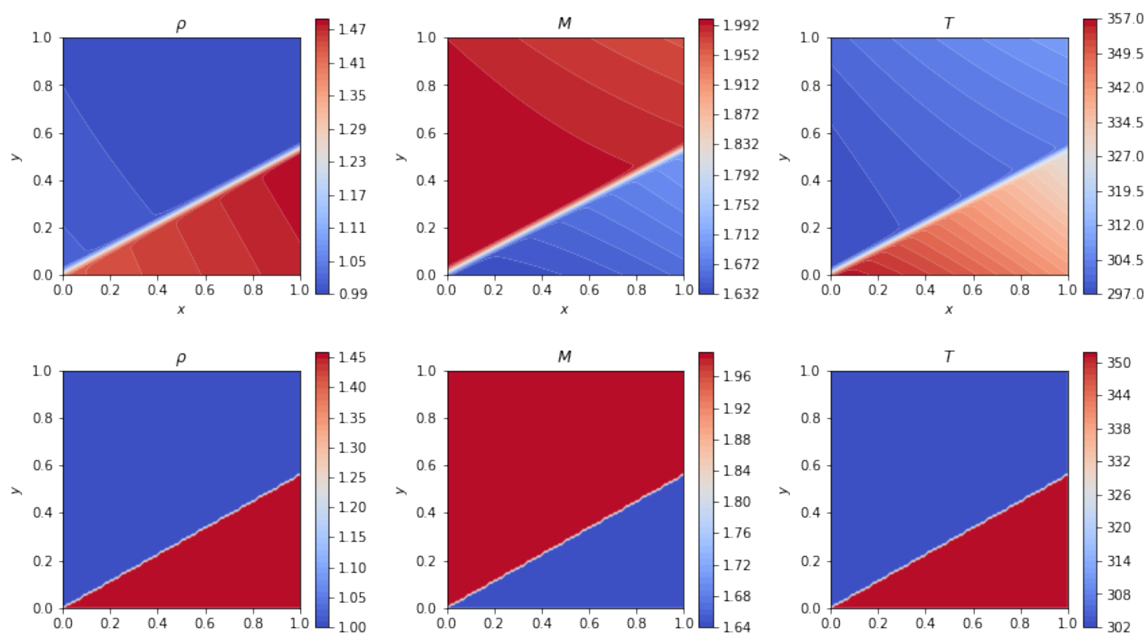
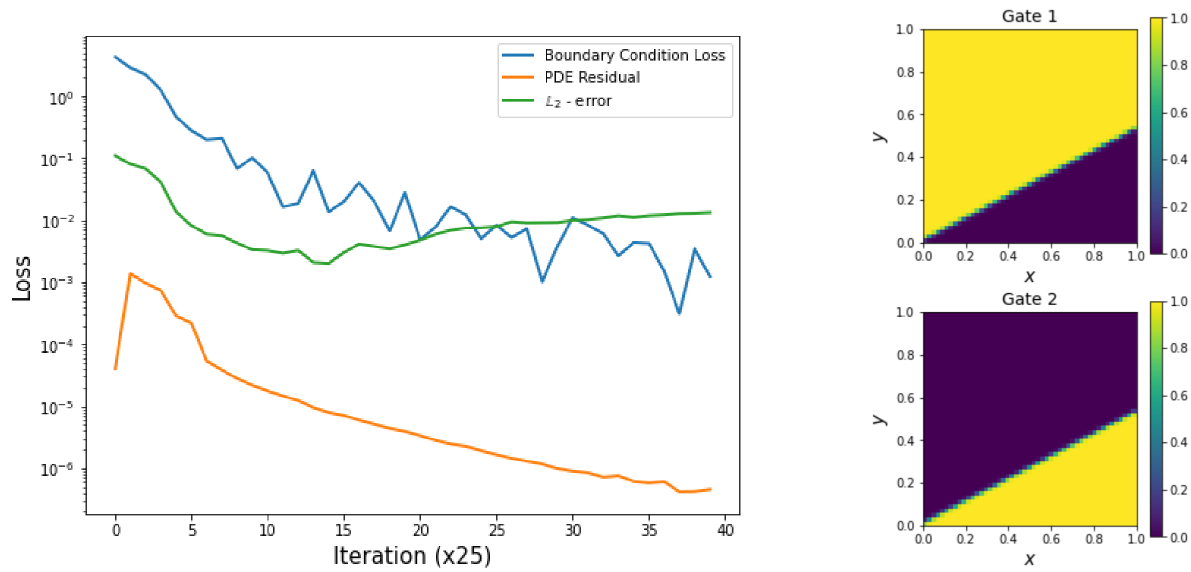


Figure 5: Density, Mach, and Temperature Distributions (Top row: MoE PINNs, bottom row: exact solution)

[1] An animation of the gating network learning the shock can be found at the following link.



(a) Loss Convergence

(b) Visualization of Gating Networks

Figure 6: Mixture of Experts Solution Characteristics

3.4 Causal Training

The final approach modified the previous MoE implementation to include causal training with causality implemented in the x direction. The resulting the converged flow field is visually identical to that in figure 5. However the convergence behavior of the relative $\mathbb{L}_2$-error is notably different. Figure 7 demonstrates that while the early convergence behavior is similar between the two training methods, the causal training implementation reduces the model's susceptibility to over-training, and leads to a reduced relative error even when early-stopping is implemented for both versions. The best relative error achieved by each model is summarized in table 1.

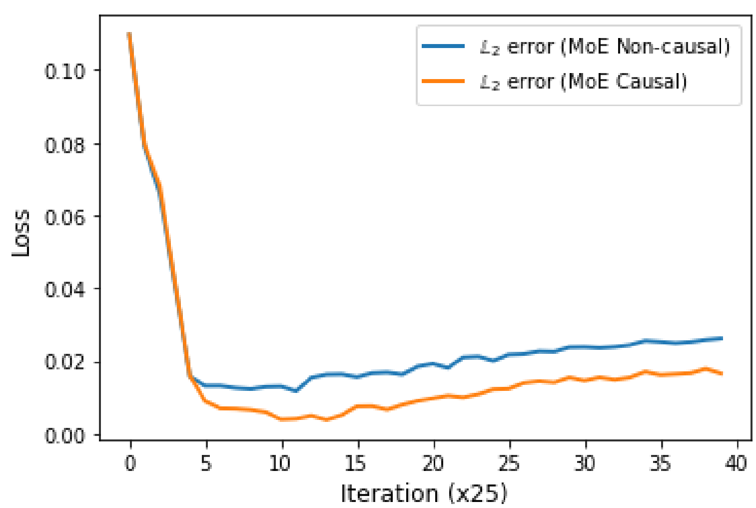


Figure 7: Relative $\mathbb{L}_2$-error Convergence: Causal vs. Non-causal Training

Method	Architecture	Iterations to Convergence	Relative $\mathbb{L}_2$-error
Traditional PINN	4 layers, 40 neurons	10,000	17.6%
MoE Non-causal Training	GN: 2 layers, 64 neurons EN: 2 layers, 32 neurons	400	0.6%
MoE Causal Training	GN: 2 layers, 64 neurons EN: 2 layers, 32 neurons	400	0.3%

Table 1: Summary of Model Performance

4 Conclusions

The results of the three primary models applied to the test case are detailed in the previous section. The dramatic reduction in the relative error of the PINNs model when coupled with the MoE architecture suggests that the MoE approach is particularly well-suited for modeling hyperbolic PDEs with discontinuities internal to the domain. The cooperative formulation for the PDE residual loss term does not seem to compromise shock resolution, and importantly does not suffer from artificial diffusion as observed with the traditional PINNs approach in figure 4. This in tandem with the competitive formulation on the boundaries allowed the model to closely satisfy the boundary conditions while reliably learning the shock position.

A second key finding in this work has been that a spatial formulation of causal training improves relative prediction accuracy in much the same way observed for temporal causal training by Wang et al. [9]. This improvement is marginal for the present test case potentially due to the non-causal formulation already being very accurate, but it suggests that implementing causal training along hyperbolic (or almost hyperbolic) directions can force the model to explore among a subset of physically faithful flow states.

This study has demonstrated the strong performance of the MoE architecture in the context of the PINNs approach for the oblique shock wave problem. Additional marginal improvement was achieved with causal training along the direction of the body surface. This is thought to be a first step in the process to better understanding how the MoE framework and other adaptions of the PINNs approach can be tailored to solving high-speed compressible flow problems.

5 Future Work

Future extensions of this work would look to extend the results for the MoE implementation to the high Mach regime ($M_1 > 4.0$), and to large flow deflection angles, ($\delta > 20°$), where stronger shocks are expected to form. A more general line of inquiry would investigate whether the MoE version of PINNs can adequately model bow shock waves, where the post-shock flow is no longer constant and the wave geometry is not linear.

Additional work may focus on merging the MoE architecture with the CV-PINNs framework, to better understand whether both or just one of the two is necessary when simulating compressible flows with data-driven methods.

References

[1] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow.* Computational Methods in Mechanics and Thermal Sciences. Hemisphere, Washington, 1980.

[2] Kartikey Asthana and Antony Jameson. High-order flux reconstruction schemes with minimal dispersion and dissipation. *Journal of Scientific Computing*, 62(3):913–944, 2015.

[3] F.D. Witherden, P.E. Vincent, and A. Jameson. Chapter 10 - high-order flux reconstruction schemes. In Rémi Abgrall and Chi-Wang Shu, editors, *Handbook of Numerical Methods for Hyperbolic Problems*, volume 17 of *Handbook of Numerical Analysis*, pages 227–263. Elsevier, 2016.

[4] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction.* Springer Berlin Heidelberg, 2009.

[5] J. Romero, F. D. Witherden, and A. Jameson. A direct flux reconstruction scheme for advection—diffusion problems on triangular grids. *Journal of Scientific Computing*, 73(2–3):1115–1144, 2017.

[6] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[7] Zhiping Mao, Ameya Jagtap, and George Karniadakis. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 03 2020.

[8] Waleed Diab and Mohammed Al Kobaisi. Pinns for the solution of the hyperbolic buckley-leverett problem with a non-convex flux function, 2021.

[9] Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint arXiv:2203.07404*, 2022.

[10] Ravi G. Patel, Indu Manickam, Nathaniel A. Trask, Mitchell A. Wood, Myoungkyu Lee, Ignacio Tomas, and Eric C. Cyr. Thermodynamically consistent physics-informed neural networks for hyperbolic systems. *Journal of Computational Physics*, 449:110754, 2022.

[11] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[12] Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks. 2022.