Fourier Neural Operator (FNO) for the Sod Shock Tube

George Hollyer, Roy Wang
ghollyer@seas.upenn.edu, wangroy@sas.upenn.edu

12 May 2025

1 Introduction

In functional analysis, a linear operator is a linear mapping between infinite-dimensional function spaces. Solving a system of partial differential equations (PDEs) can be viewed as finding a solution operator that maps the input function (initial and boundary conditions) to the output function (solution). Following from this perspective, neural operators have emerged in deep learning as powerful surrogates to traditional and expensive PDE solvers. One flavor of neural operator, the Fourier Neural Operator (FNO), has shown itself to be adept for fluid mechanics systems such as the Burgers' equation, Darcy flow, and Navier-Stokes equation [1].

We motivate the framework for neural operators by first examining the structure of a standard MLP. For input vector u and output vector v, we can write a general MLP as

$$v = (W_L \circ \sigma_L \circ \cdots \circ \sigma_1 \circ W_0)u, \tag{1}$$

where W is an affine or convolutional layer, and σ is a (non-linear) activation function. A neural operator is defined in an analogous manner, with functions as input and output rather than vectors. To handle this, we replace W with T, where T is a bounded (continuous) linear operator. $T : u_i \mapsto u_{i+1}$ is often formulated as:

$$T(u)(x) = K(u)(x) + Wu(x) = \int_\Omega \kappa_\phi(x, y)u(y)dy + W_\phi u(x), \tag{2}$$

where $K(u)(x)$ is a global integral operator, κ_ϕ is a learnable kernel function parameterized by a neural network, W_ϕ is a learnable linear transformation, and Ω is our given domain. Hence the $(n+1)$th neural operator layer can be represented as

$$u_{n+1}(x) = \sigma\left(T(u_n)(x)\right) \tag{3}$$

Since it is possible that our input and output functions live in function spaces of different dimensions, we introduce a lifting layer Q on the input side of our network and a projection layer P on the output side of our network. This will allow us to lift the input features into a higher dimensional hidden space, pass them through the operator layers, and project them down to the output space. So, the neural operator $H_\phi : u \mapsto v$ that we wish to learn can be expressed as

$$H_\phi = P \circ u_L \circ \cdots \circ u_1 \circ Q \tag{4}$$

In Section 2, we discuss a specific neural operator architecture, the Fourier neural operator (FNO). In Section 3, we provide background on the Sod shock tube and the Rusanov finite difference scheme we use as a baseline. In Section 4, we present a numerical experiment applying the FNO to the 1D Sod shock tube. In Section 5, we discuss our results and make concluding remarks. In the Appendix, we provide additional selected results.

2 Fourier Neural Operator

The Fourier transform has found significant application in spectral methods for solving differential equations. The Fourier Neural Operator is motivated by these techniques, as theory from Fourier analysis allows nice properties while working in Fourier space. The most relevant fact for our purposes is the convolution theorem:

Theorem 1 (Convolution Theorem)**.** *For $f : \Omega \to \mathbb{R}^n$, let $\mathcal{F}[f](\xi) = \int_\Omega f(x)e^{-2\pi i\xi\cdot x}dx$ be the Fourier transform of f. Then for functions f and g, we have*

$$\mathcal{F}[f * g](\xi) = \mathcal{F}[f](\xi) \cdot \mathcal{F}[g](\xi) \tag{5}$$

Taking the inverse Fourier Transform of (5), we also obtain the the corollary

$$(f * g)(x) = \mathcal{F}^{-1}\left[\mathcal{F}[f](\xi) \cdot \mathcal{F}[g](\xi)\right]$$

To leverage this property in formulating neural operators, we adapt K from (2) to be a convolution, more specifically assuming our kernel $\kappa_\phi(x, y)$ can be expressed as $\kappa_\phi(x - y)$. Letting $R_\phi(\xi) = \mathcal{F}[\kappa_\phi](\xi)$ and applying the convolution theorem, we have

$$\begin{aligned} K(u)(x) &= \int_\Omega \kappa_\phi(x - y)u(y)dy = (\kappa_\phi * u)(x) \\ \Rightarrow K(u)(x) &= \mathcal{F}^{-1}[R_\phi \cdot \mathcal{F}(u)](x). \end{aligned}$$

We can rewrite our neural operator layers (3) as

$$u_{n+1}(x) = \sigma\Big(K(u)(x) + Wu(x)\Big) = \sigma\Big(\mathcal{F}^{-1}[R_\phi \cdot \mathcal{F}(u)](x) + Wu(x)\Big) \tag{6}$$

We call (6) a Fourier layer. In practice, we truncate the infinite Fourier series that (6) admits to a finite number of modes n_{modes}. The more modes we keep, the more that the FNO can learn more local behavior. Conversely, less retained modes means that the FNO will capture less local behavior and learn more global dynamics. Intrinsic guarantees we receive by using the FNO include discretization invariance, as our parameters are directly learned in Fourier space [1]. This means that we can train our neural operator on lower-resolution data, and make predictions on high-resolution data without losing accuracy.

3 Sod Shock Tube

The Sod shock tube is a specific one dimensional case of Euler's equations of gas dynamics [2] used to evaluate the accuracy of computational methods in fluid dynamics. The equations model the system in terms of the conserved properties of density ρ, flux (density times velocity) ρv, and energy E.

$$\frac{\delta}{\delta t}\begin{bmatrix}\rho \\ \rho v \\ E\end{bmatrix} + \frac{\delta}{\delta x}\begin{bmatrix}\rho v \\ \rho v^2 + p \\ v(E + p)\end{bmatrix} = 0 \tag{7}$$

Pressure p can be described using these quantities and the ratio of heat capacities, $\gamma = \frac{C_p}{C_v}$:

$$p = (\gamma - 1)(E - \frac{1}{2}\rho v^2). \tag{8}$$

For our experiments, we take $\gamma = 1.4$. The Sod shock tube evaluates the dynamics of a discontinuous initial condition. A shock front will propagate into the lower pressure side of the system. The standard form of the problem is on the domain of $[0, 1]$, with a barrier at $x = 0.5$ with the following initial conditions at time $t = 0$:

$$(p, \rho, v) = \begin{cases} (1, 1, 0) & x \in [0, 0.5] \\ (0.1, 0.125, 0) & x \in (0.5, 1] \end{cases} \tag{9}$$

Exact solutions of the problem are readily obtainable for arbitrary initial conditions using Riemann solvers. The details of how training datasets were generated are discussed in section 4.

3.1 Rusanov FDM

As a baseline, we implement the Rusanov finite-difference scheme[3] (also sometimes called local Lax-Friederichs) and use it to solve the standard Sod shock tube problem. The local state of the system, U_i, is defined as the conservative variables of the Euler equations. Updates to the state are computed from the Rusanov flux, $\hat{F}$, and dissipation, λ, between the state and its neighbors.

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\left(\hat{F}_{i+1/2} - \hat{F}_{i-1/2}\right) \tag{10}$$

$$\hat{F}_{i+1/2} = \frac{1}{2}\left[F(U_i^n) + F(U_{i+1}^n)\right] - \frac{1}{2}\lambda_{i+1/2}\left(U_{i+1}^n - U_i^n\right) \tag{11}$$

$$F(U) = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ v(E+p) \end{bmatrix} \tag{12}$$

$$\lambda_{i+1/2} = \max\left(|v_i| + c_i,\ |v_{i+1}| + c_{i+1}\right) \quad \text{where} \quad c = \sqrt{\gamma \frac{p}{\rho}} \tag{13}$$

This scheme conserves physical quantities, is numerically stable and exhibits the shock front behaviors of the Sod shock tube problem. We implemented the Rusanov solver on a uniformly discretized grid from 0 to 1 with $N = 500$ points and solved the system to time $t = 0.1$. A Courant-Friederich-Lewy coefficient of 0.75 was used to define a variable time step dt.

4 Numerical Experiments

For our experiment, we implement an FNO and evaluate it on the 1-D Sod shock tube described in Section 3. Our dataset is composed of initial condition-final solution pairs, where the initial conditions for the primitive variables are given as

$$(p, \rho, v) = \begin{cases} (p_l, 1, 0) & x \in [0, 0.5] \\ (0.1, 0.125, 0) & x \in (0.5, 1] \end{cases}$$

where $p_l \in [1.0, 5.0]$ and an analytical solver [4] is used to generate solutions at $t = 0.1$. We generate 5000 equispaced p_l values and randomly assign 3750 (75%) samples for training and 1250 (25%) for testing. Our initial conditions and final solutions live on the uniformly discretized grid $[0, 1]$ of $N = 500$ points. We train the FNO to learn an operator from the initial conditions space to the solution space at final time $t = 0.1$. We train in batches by optimizing the mean-squared error between the ground truth solution u and the predicted solution $\hat{u}$ over all samples in the batch. The MSE loss function is given by

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N_b}\frac{1}{N}\sum_{i=1}^{N_b}\sum_{j=1}^{N}(u_{i,j} - u_{pred,i,j})^2, \tag{14}$$

where N_b is the batch size. We initialize our FNO with hyperparamaters $n_{\text{modes}} = 50$, width $= 4$, $n_L = 4$, and $N_b = 15$, where width is the dimension of our hidden Fourier layers, and n_L is the number of Fourier layers used. We use the tanh activation function and Adam optimizer initialized with learning rate 0.001, training

over a total of 1000 epochs. Due to time constraints, we did not conduct a test sweep of hyperparameters, but did gain some insights about enhancing performance while training different iterations of the FNO. For example, to best capture the shock dynamics of our problem, we increased the number of kept Fourier modes to 50, which we had originally initialized as $n_{\text{modes}} = 12$. Moreover, an earlier dataset only contained 500 examples, and we found evidence of overfitting when comparing our training and validation loss. After expanding our dataset to 5000 examples and increasing the number of Fourier modes, both the training and validation loss were of order 10^{-4} (Figure 1). Our best performing model had 3,315 trainable parameters. This model is trainable in approximately 40 minutes using the CPU on the free tier of Google Colaboratory.

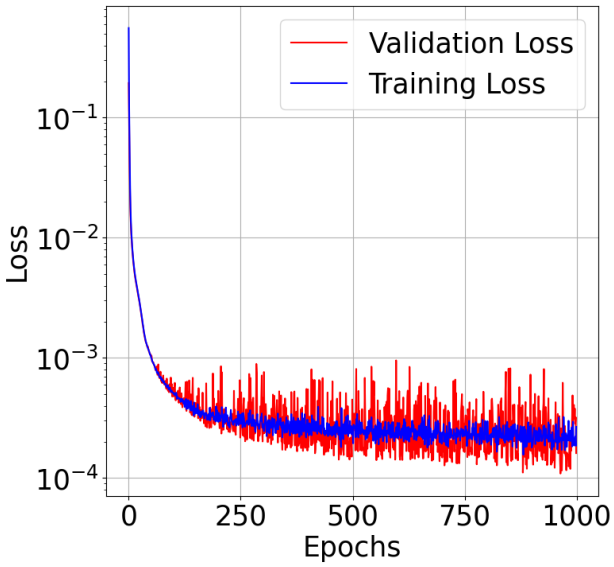


Figure 1: Training and Validation Loss for our best performing model

5 Discussion and Concluding Remarks

Figure 2 shows three randomly selected test cases with varying initial left pressures that demonstrates the FNO's ability to quantitatively and qualitatively match the ground truth. For every case, the even rows of Figure 2 show the point-wise absolute errors in each profile. In Figure 3, we compare our FNO prediction for the standard Sod shock problem (initial conditions given by (9)) to the Rusanov prediction.

	Pressure	Density	Velocity
FNO	0.2323	0.2336	0.5765
Rusanov	0.3379	0.3679	0.8400

Table 1: L^2 errors for the accompanying Figure 3.

The L^2 error of the FNO solution to the standard Sod shock tube problem is lower than that of the Rusanov solver in all three variables (Table 1). This is notable considering the low number of trainable parameters (3,315) and small training dataset (3,750 examples). However, the absolute error profiles shown in Figure 3 differ significantly between the two methods. The Rusanov solver shows extremely low error in steady-state areas of the profiles but large errors at the shock fronts due to smoothing. This is a known drawback of the Rusanov scheme [5]. By contrast, the FNO solution is closer to the ground truth at discontinuities but shows significant error in the steady-state areas of the profile. This appears to be an artifact introduced by the Fourier convolution. The solutions show similarity to approximate PDE solutions from a Fourier series with a small number of modes. Oscillations in the profile are present near sharp features. These features show the greatest L^2 error in the error profiles.

One notable downside of the FNO architecture is the lack of inbuilt physical guarantees. The FNO solutions show some undershoot in the low pressure and density areas of the profile and could produce unphysical features such as negative pressures or densities. On the other hand, the Rusanov scheme conserves physical quantities and does not carry the same risks as a result.

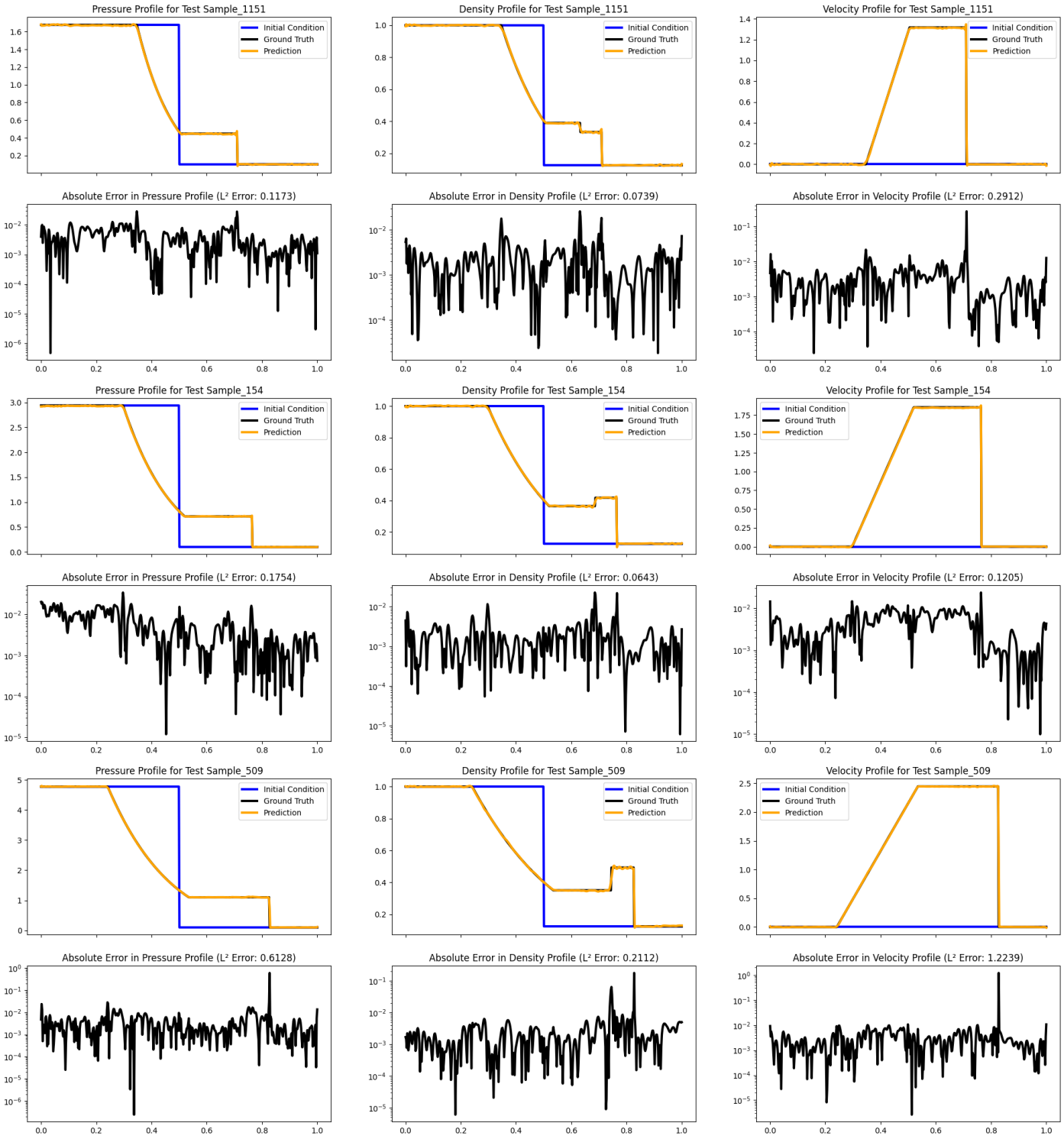


Figure 2: Test samples selected at random. Top (odd) rows: Initial conditions, ground truth solution, and FNO prediction for pressure, density, and velocity. Bottom (even) rows: Point-wise absolute error and L^2 error between the FNO prediction and ground truth for each profile.

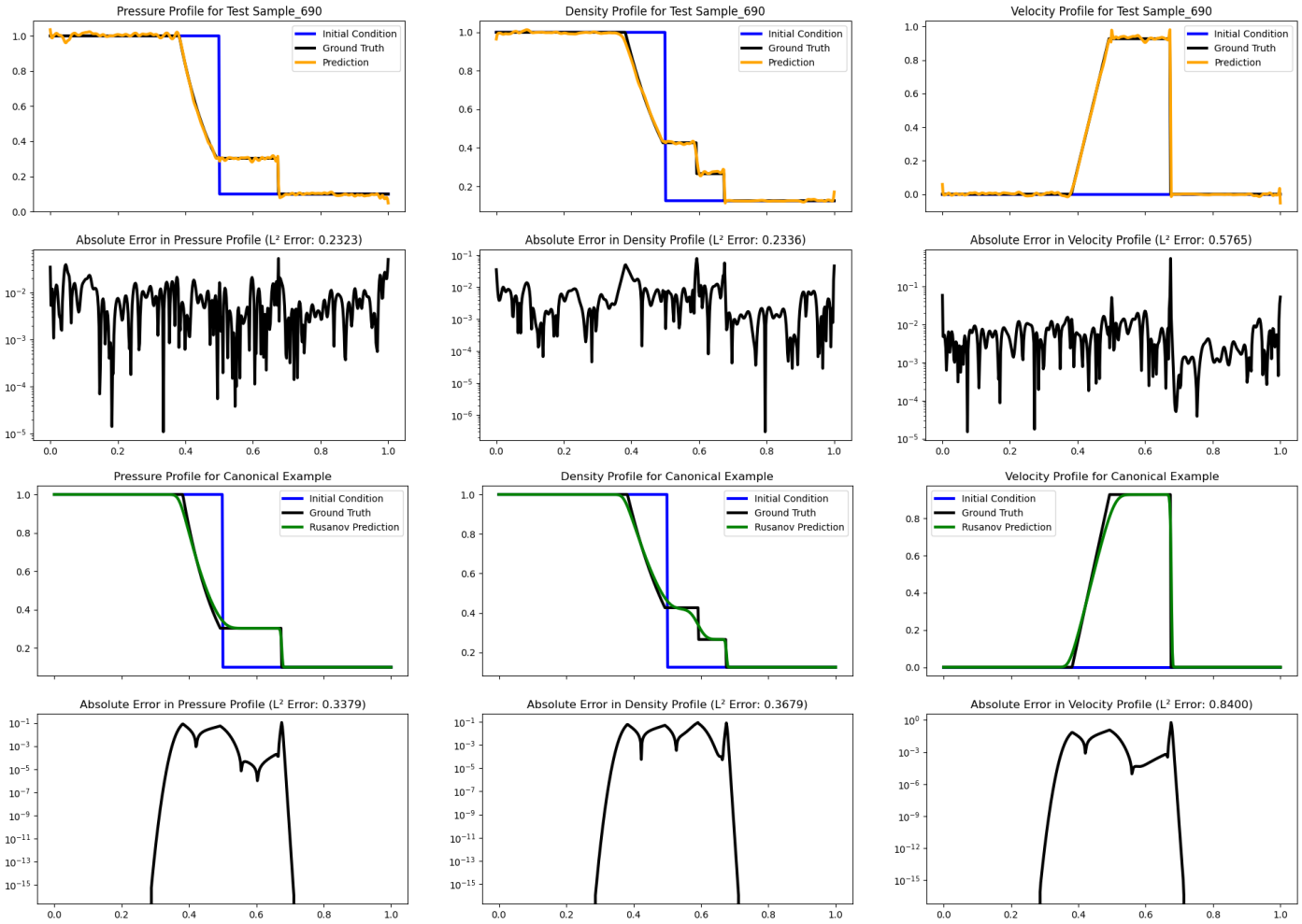


Figure 3: FNO and Rusanov FDM results for the standard Sod shock tube problem. Top Two: Initial conditions, ground truth solution, and FNO prediction with corresponding point-wise absolute and L^2 errors between FNO prediction and ground truth. Bottom Two: Initial conditions, ground truth solution, and Rusanov prediction with corresponding point-wise absolute error and and L^2 error between Rusanov prediction and ground truth.

This project can be naturally extended in several ways. First, the dataset we generated has only one degree of freedom (the pressure of the left state). We hypothesize that our extremely small model performed well in part due to this, and thus training on a higher degree of freedom dataset would be an interesting step to see if the FNO can generalize outside of our specific situation. Retaining good accuracy in this case would likely require increasing the width of our Fourier layers and the number of Fourier modes. Second, this model could be scaled up to see if the FNO can better model the dynamics at the discontinuities in the density and profiles profile. One possible way to conduct this is in the form of an activation function and hyperparameter sweep, testing all combinations of activation functions, batch size, the number of Fourier modes, and the depth and width of the Fourier layers. Such an experiment would allow for the verification of the most performance-significant features of the FNO while training a larger and deeper model. Another immediate step would be to verify the discretization invariance of the FNO, testing our FNO's predictions for initial conditions given on higher-resolution, non-uniform discretization of the grid.

In summary, the FNO architecture is capable of solving the 1D Sod shock tube problem with comparable accuracy to the Rusanov FDM scheme at low parameter counts. However, the two methods show different types of errors and the FNO does not retain the physical guarantees of the Rusanov scheme.

References

[1] Zongyi Li et al. "Fourier Neural Operator for Parametric Partial Differential Equations". In: *CoRR* abs/2010.08895 (2020). arXiv: 2010.08895. URL: https://arxiv.org/abs/2010.08895.

[2] Jorge Balbás. *Examples- 1D Euler Equations: Sod Problem — csun.edu*. https://www.csun.edu/~jb715473/examples/euler1d.htm#density. [Accessed 11-05-2025].

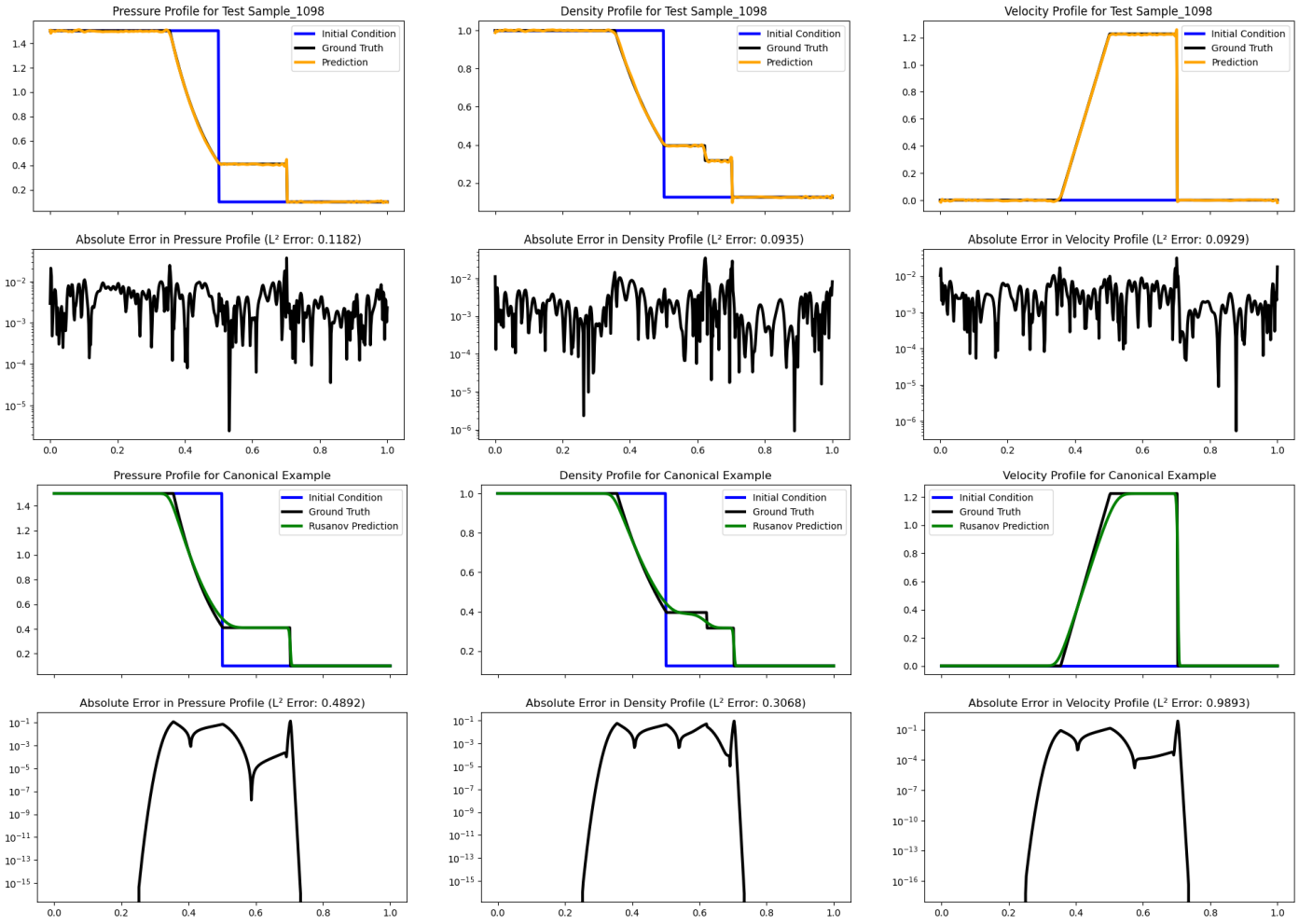
[3] V.V Rusanov. "The calculation of the interaction of non-stationary shock waves and obstacles". In: *USSR Computational Mathematics and Mathematical Physics* 1.2 (1962), pp. 304–320. ISSN: 0041-5553. DOI: https://doi.org/10.1016/0041-5553(62)90062-9. URL: https://www.sciencedirect.com/science/article/pii/0041555362900629.

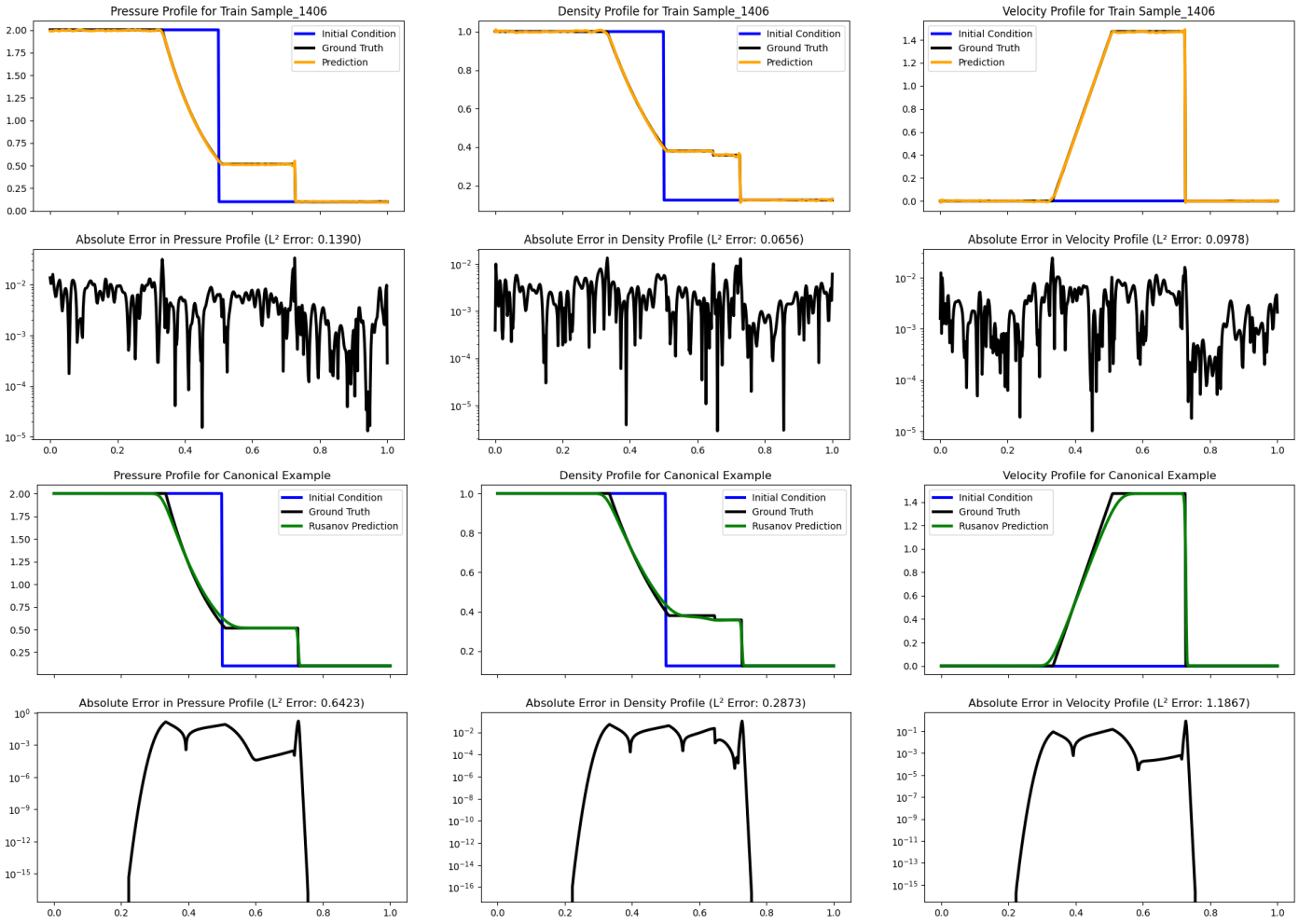
[4] Isaac Jonathan Backus. *GitHub - ibackus/sod-shocktube: A simple pythonic implementation of a Riemann solver for the analytic solution of the sod shock tube. — github.com*. https://github.com/ibackus/sod-shocktube. [Accessed 11-05-2025].

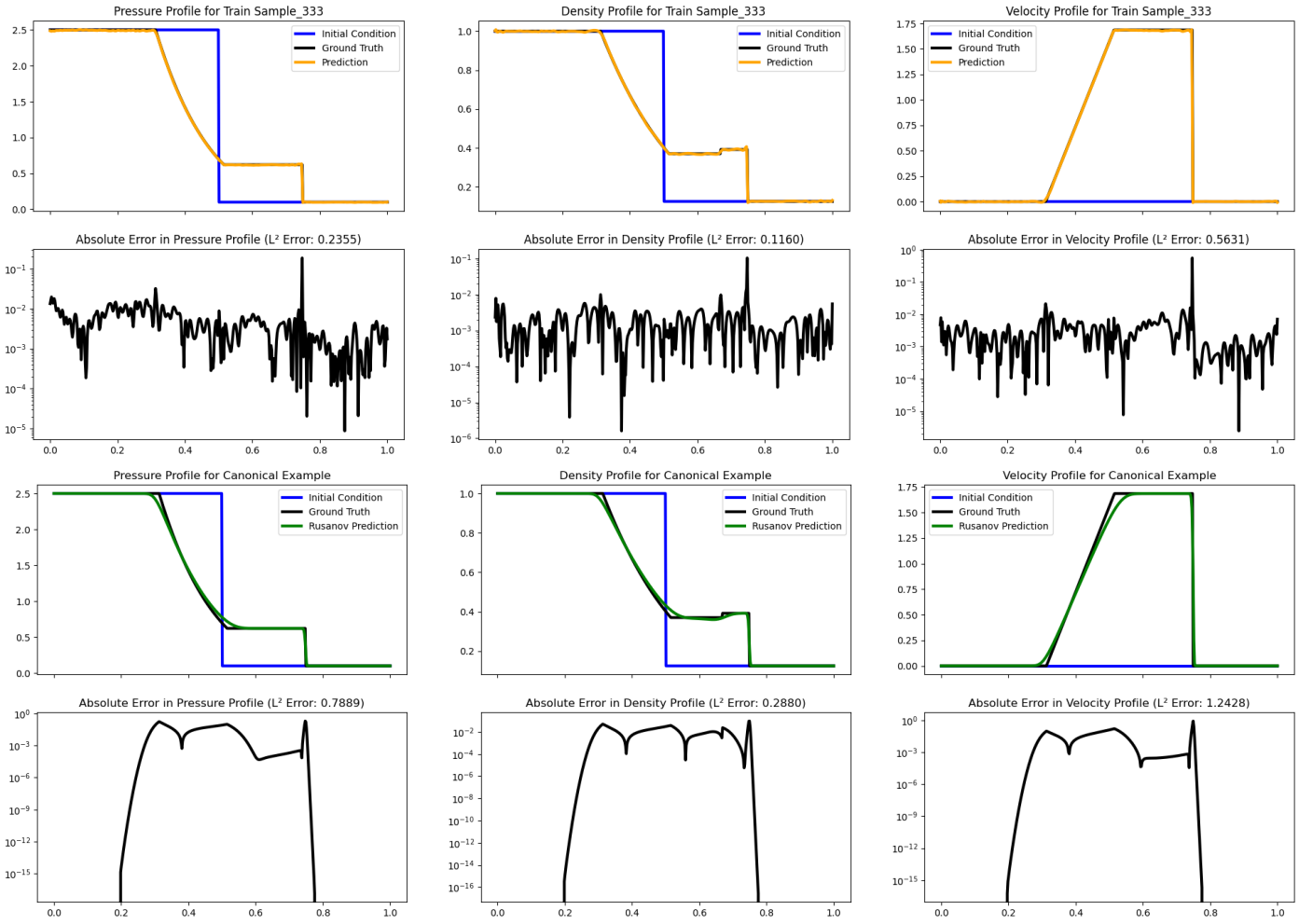
[5] K. Srinivas, J. Gururaja, and K. Krishna Prasad. "An assessment of the quality of selected finite difference schemes for time dependent compressible flows". In: *Journal of Computational Physics* 20.2 (1976), pp. 140–159. ISSN: 0021-9991. DOI: https://doi.org/10.1016/0021-9991(76)90060-7. URL: https://www.sciencedirect.com/science/article/pii/0021999176900607.

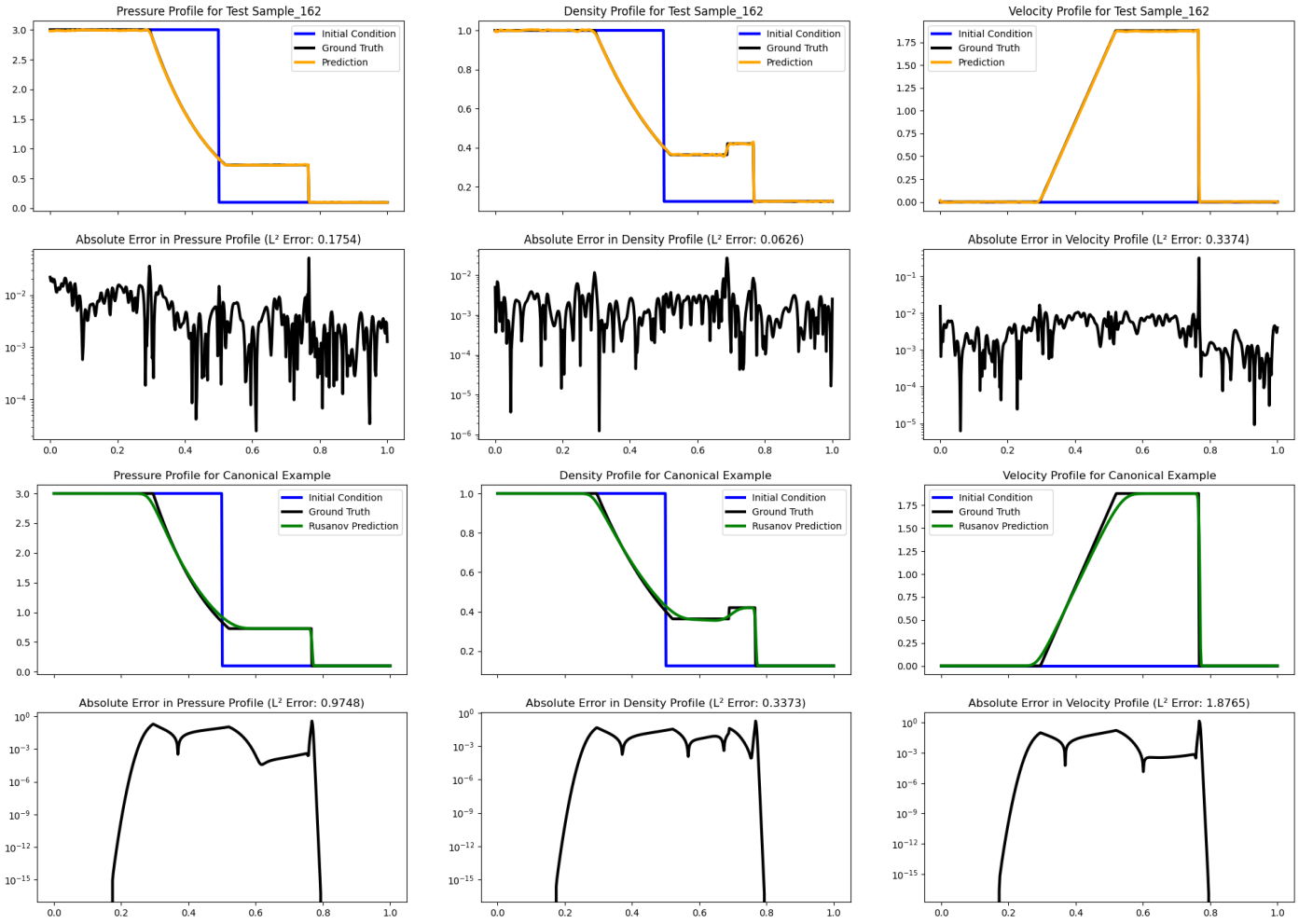
6 Appendix

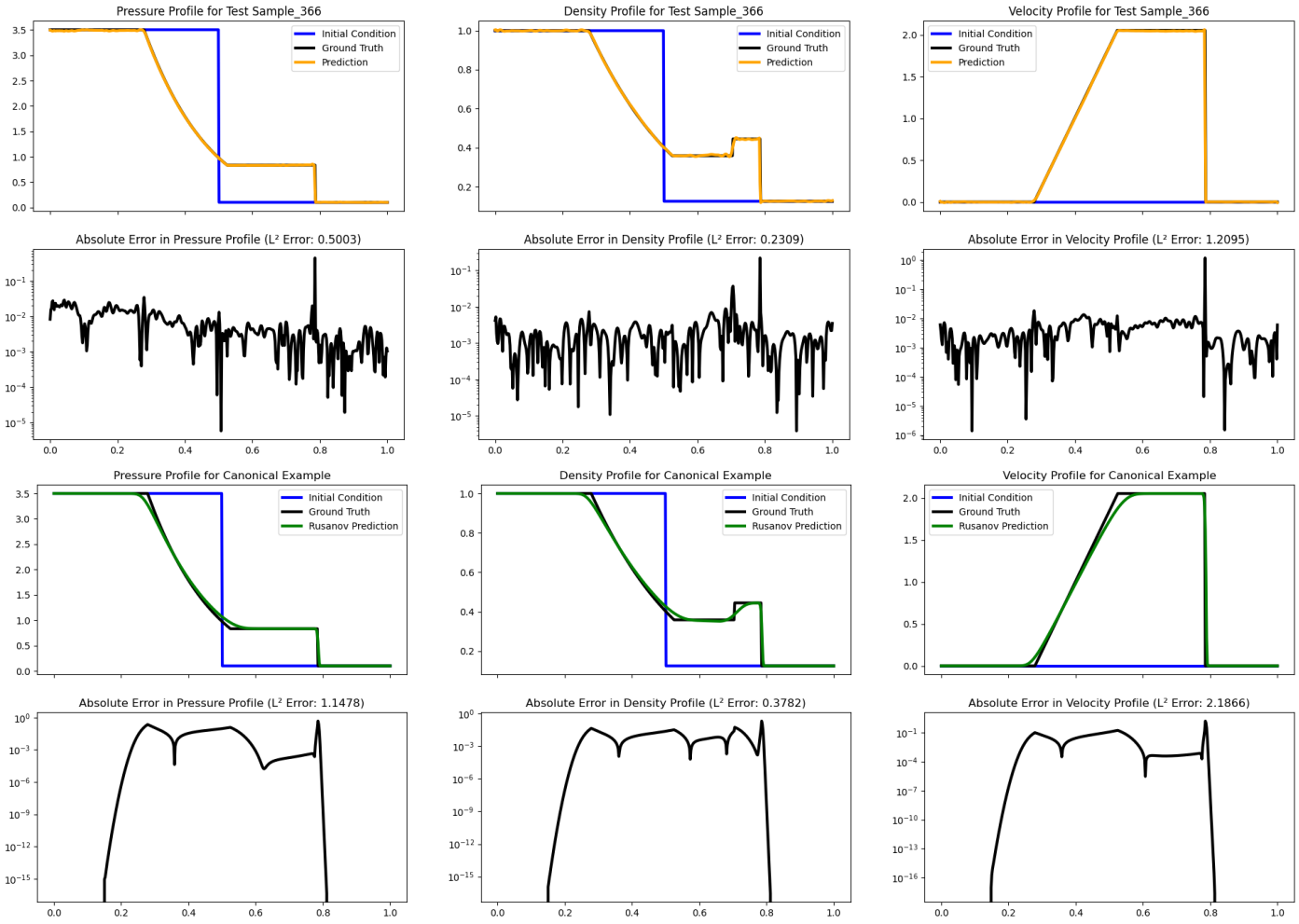
Selected comparisons between Rusanov numerical solution, FNO prediction, and ground truth solution, with the initial left pressure ranging in $[1.5, 5]$ in 0.5 increments.

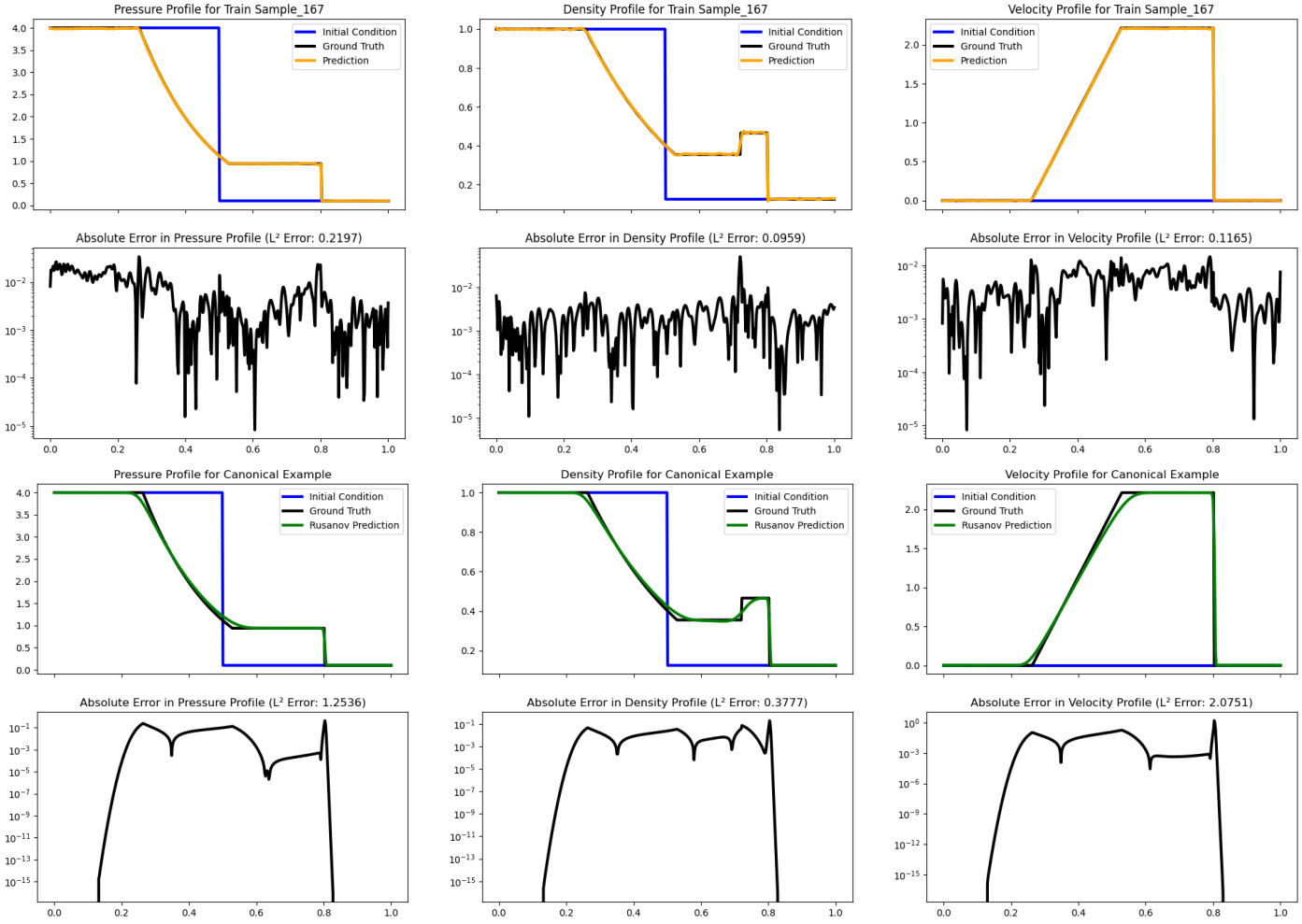

Pressure Profile for Test Sample_1098
Initial Condition
Ground Truth
Prediction
Density Profile for Test Sample_1098
Initial Condition
Ground Truth
Prediction
Velocity Profile for Test Sample_1098
Initial Condition
Ground Truth
Prediction
Absolute Error in Pressure Profile (L² Error: 0.1182)
Absolute Error in Density Profile (L² Error: 0.0935)
Absolute Error in Velocity Profile (L² Error: 0.0929)
Pressure Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Density Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Velocity Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Absolute Error in Pressure Profile (L² Error: 0.4892)
Absolute Error in Density Profile (L² Error: 0.3068)
Absolute Error in Velocity Profile (L² Error: 0.9893)

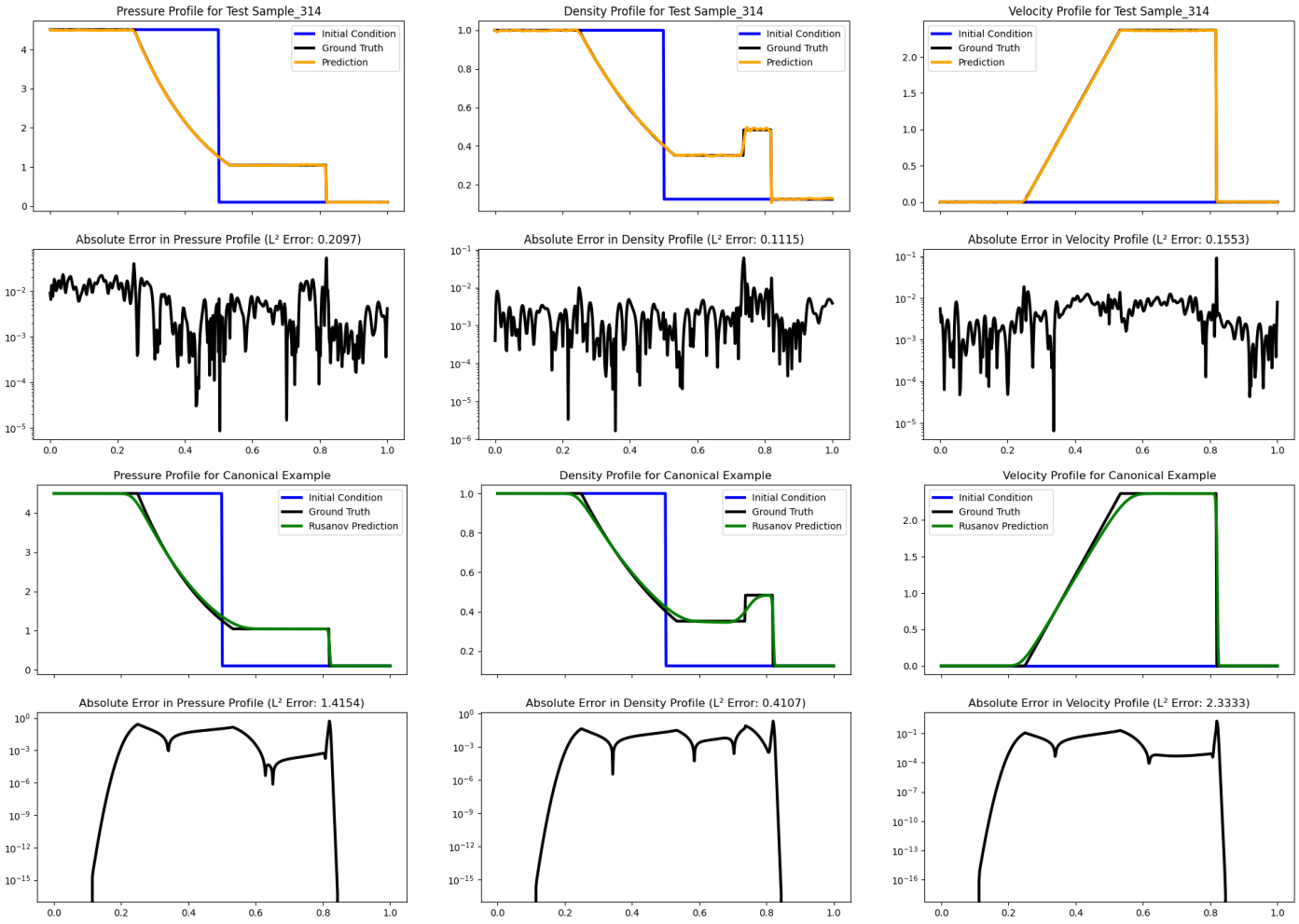

Pressure Profile for Train Sample_1406
Initial Condition
Ground Truth
Prediction
Density Profile for Train Sample_1406
Initial Condition
Ground Truth
Prediction
Velocity Profile for Train Sample_1406
Initial Condition
Ground Truth
Prediction
Absolute Error in Pressure Profile (L² Error: 0.1390)
Absolute Error in Density Profile (L² Error: 0.0656)
Absolute Error in Velocity Profile (L² Error: 0.0978)
Pressure Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Density Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Velocity Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Absolute Error in Pressure Profile (L² Error: 0.6423)
Absolute Error in Density Profile (L² Error: 0.2873)
Absolute Error in Velocity Profile (L² Error: 1.1867)

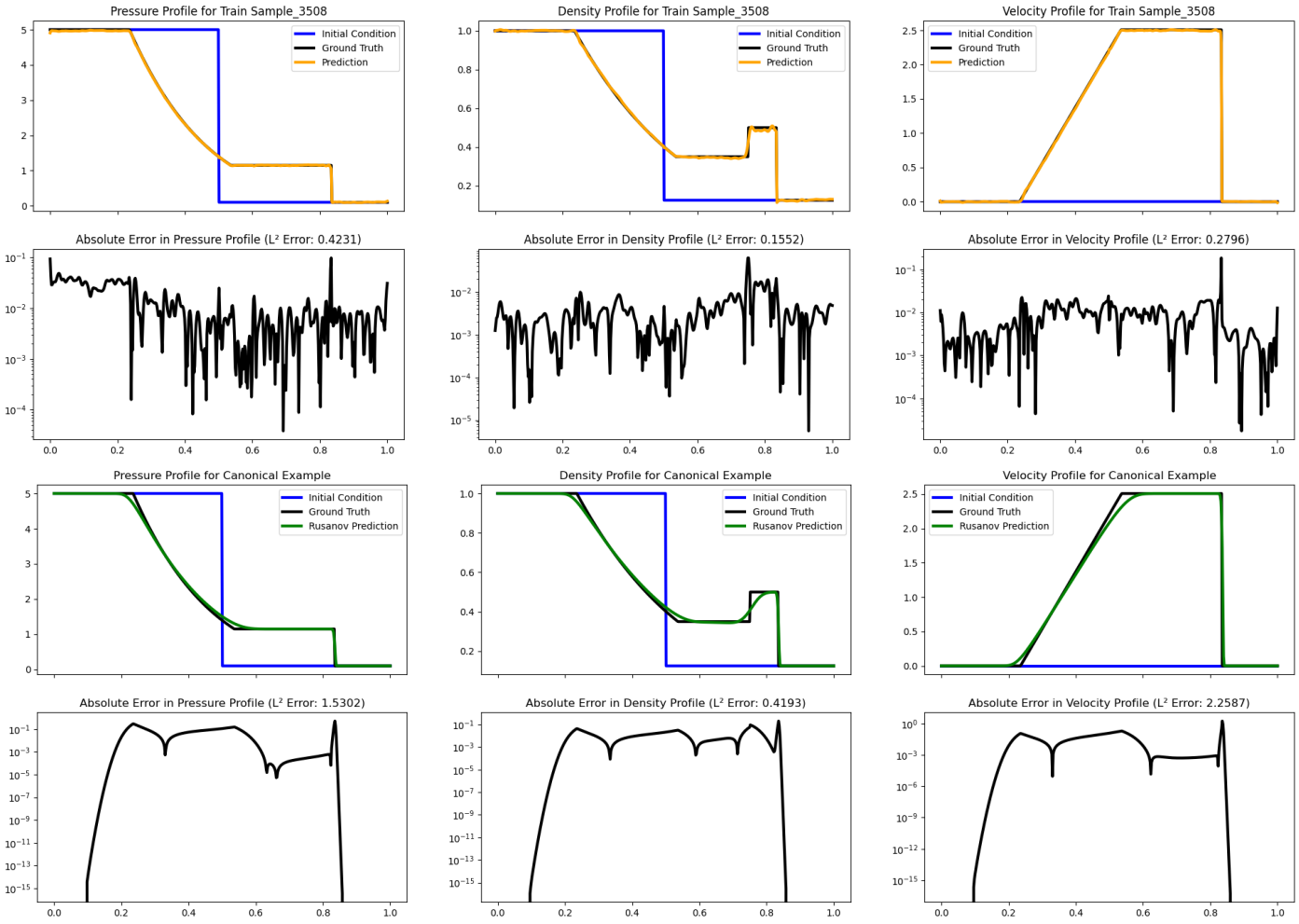

Pressure Profile for Train Sample_333
Initial Condition
Ground Truth
Prediction
Density Profile for Train Sample_333
Initial Condition
Ground Truth
Prediction
Velocity Profile for Train Sample_333
Initial Condition
Ground Truth
Prediction
Absolute Error in Pressure Profile (L² Error: 0.2355)
Absolute Error in Density Profile (L² Error: 0.1160)
Absolute Error in Velocity Profile (L² Error: 0.5631)
Pressure Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Density Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Velocity Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Absolute Error in Pressure Profile (L² Error: 0.7889)
Absolute Error in Density Profile (L² Error: 0.2880)
Absolute Error in Velocity Profile (L² Error: 1.2428)


Pressure Profile for Test Sample_162
Initial Condition
Ground Truth
Prediction
Density Profile for Test Sample_162
Initial Condition
Ground Truth
Prediction
Velocity Profile for Test Sample_162
Initial Condition
Ground Truth
Prediction
Absolute Error in Pressure Profile (L² Error: 0.1754)
Absolute Error in Density Profile (L² Error: 0.0626)
Absolute Error in Velocity Profile (L² Error: 0.3374)
Pressure Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Density Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Velocity Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Absolute Error in Pressure Profile (L² Error: 0.9748)
Absolute Error in Density Profile (L² Error: 0.3373)
Absolute Error in Velocity Profile (L² Error: 1.8765)


Pressure Profile for Test Sample_366
Initial Condition
Ground Truth
Prediction
Density Profile for Test Sample_366
Initial Condition
Ground Truth
Prediction
Velocity Profile for Test Sample_366
Initial Condition
Ground Truth
Prediction
Absolute Error in Pressure Profile (L² Error: 0.5003)
Absolute Error in Density Profile (L² Error: 0.2309)
Absolute Error in Velocity Profile (L² Error: 1.2095)
Pressure Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Density Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Velocity Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Absolute Error in Pressure Profile (L² Error: 1.1478)
Absolute Error in Density Profile (L² Error: 0.3782)
Absolute Error in Velocity Profile (L² Error: 2.1866)


Pressure Profile for Train Sample_167
Initial Condition
Ground Truth
Prediction
Density Profile for Train Sample_167
Initial Condition
Ground Truth
Prediction
Velocity Profile for Train Sample_167
Initial Condition
Ground Truth
Prediction
Absolute Error in Pressure Profile (L² Error: 0.2197)
Absolute Error in Density Profile (L² Error: 0.0959)
Absolute Error in Velocity Profile (L² Error: 0.1165)
Pressure Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Density Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Velocity Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Absolute Error in Pressure Profile (L² Error: 1.2536)
Absolute Error in Density Profile (L² Error: 0.3777)
Absolute Error in Velocity Profile (L² Error: 2.0751)


Pressure Profile for Test Sample_314
Initial Condition
Ground Truth
Prediction
Density Profile for Test Sample_314
Initial Condition
Ground Truth
Prediction
Velocity Profile for Test Sample_314
Initial Condition
Ground Truth
Prediction
Absolute Error in Pressure Profile (L² Error: 0.2097)
Absolute Error in Density Profile (L² Error: 0.1115)
Absolute Error in Velocity Profile (L² Error: 0.1553)
Pressure Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Density Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Velocity Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Absolute Error in Pressure Profile (L² Error: 1.4154)
Absolute Error in Density Profile (L² Error: 0.4107)
Absolute Error in Velocity Profile (L² Error: 2.3333)


Pressure Profile for Train Sample_3508
Initial Condition
Ground Truth
Prediction
Density Profile for Train Sample_3508
Initial Condition
Ground Truth
Prediction
Velocity Profile for Train Sample_3508
Initial Condition
Ground Truth
Prediction
Absolute Error in Pressure Profile (L² Error: 0.4231)
Absolute Error in Density Profile (L² Error: 0.1552)
Absolute Error in Velocity Profile (L² Error: 0.2796)
Pressure Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Density Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Velocity Profile for Canonical Example
Initial Condition
Ground Truth
Rusanov Prediction
Absolute Error in Pressure Profile (L² Error: 1.5302)
Absolute Error in Density Profile (L² Error: 0.4193)
Absolute Error in Velocity Profile (L² Error: 2.2587)