COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

APMA6302: Numerical Analysis for PDEs:
Final Project

Learning Numerical Dissipation in Finite Volume Methods for Coarse-Grid Conservation Laws

Student:
Tushnim Yuvaraj, ty2547

Instructor:
Professor Qiang Du

Department of Applied Physics and Applied Mathematics

December 21, 2025

Contents

1 Objective

In this project, we study numerical techniques for solving partial differential equations arising in fluid dynamics, with a particular focus on finite volume methods for nonlinear conservation laws. Finite volume schemes are widely used due to their conservative structure and robustness in the presence of discontinuities [9, 14]; their behavior is determined almost entirely by the choice of the numerical flux, which encodes stability, entropy admissibility, and dissipation.

We first review the finite volume framework and the role of numerical flux functions, using the Rusanov flux as our example. While such fluxes are designed to ensure stability under worst-case conditions, they often introduce excessive numerical dissipation, leading to degraded accuracy on coarse computational grids. This trade-off between robustness and resolution motivates the central question of this project: whether numerical dissipation can be adapted using Neural Networks without sacrificing the benefits of FVMs.

To this end, we propose an NN–based numerical flux. The network is trained in a PINN-inspired fashion using high-resolution reference solutions, while preserving exact conservation, consistency, and the underlying finite volume update. Importantly, the neural network learns a scheme that improves coarse-grid accuracy within a fixed discretization.

Finally, we investigate the numerical behavior of the resulting scheme, including stability, error propagation, and long-time performance, and discuss its interpretation from a modified-equation perspective. We conclude by outlining limitations of the current approach and potential extensions, including connections to entropy-stable schemes, provable convergence guarantees, and higher-dimensional systems.

2 FVMs for Conservation laws

We briefly recall the finite volume framework for one-dimensional conservation laws; some derivations were covered in class, but we will re-derive to explicitly to demonstrate how conservation laws benefit from this formulation.

Given a partition of the domain into control volumes, the solution is represented by cell averages and evolves according to a conservative flux-difference update. The choice of a *numerical flux*, which is a function that approximates

the flux in the numerical method, determines stability, accuracy, and dissipation.

In this project, we treat the finite volume discretization as fixed and focus exclusively on modifying the numerical flux.

2.1 Discretization scheme

This conservative flux-difference formulation dates back to Godunov's seminal work [4] and underlies modern shock-capturing schemes. We follow LeVeque's exposition in chapter 4 of [9]. Consider the conservation law in one dimension. We have a uniform mesh $M_{\Delta x} = \{x_i\}$, $0 \leq i \leq M$, with $x_0 = 0$ and $x_M = 1$. We also have $x_{i+1} - x_i = \Delta x$ (A non-uniform mesh can also be considered). Now, define $x_{i+\frac{1}{2}} = x_i + \frac{1}{2}\Delta x$, and the dual cell $C_i = [\, x_{i-\frac{1}{2}},\, x_{i+\frac{1}{2}}\,]$

We have:

$$u_t + (f(u))_x = 0 \tag{1}$$

where f is called a flux function. First, Integrating over a dual cell and dividing by Δx

$$\frac{\partial \langle u \rangle_{C_i}}{\partial t} + \frac{1}{\Delta x}\left(f(u(x_{i+\frac{1}{2}})) - f(u(x_{i-\frac{1}{2}}))\right) = 0 \tag{2}$$

Now, we discretize this semi-discrete equation in time. First, we discretize in time, $\{t_n\}$, $0 \leq n \leq N$, with $t_0 = 0$ and $t_{n+1} - t_n = \Delta t$. With

$$\langle\, u(t_n, x_i)\,\rangle_{C_i} = U_i^n$$

we have, simply using Forward-Euler for the first term,

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\left(f(u(x_{i+\frac{1}{2}})) - f(u(x_{i-\frac{1}{2}}))\right) \tag{3}$$

The points like $u_{i+\frac{1}{2}}$ lie on the interface of the the cells, so we don't have a way to approximate $f(u_{i+\frac{1}{2}})$. These terms, represented by $F^n_{i+\frac{1}{2}}$, can be thought of as the average flux flowing through the interface over the time interval

$$F^n_{i+\frac{1}{2}} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(u_{i+\frac{1}{2}})\, dt$$

Once again, we need a way to approximate the interface values. Note that for adjacent cells, $u(x_{i+\frac{1}{2}}, t_n)$ lies on the right interface of C_i and on the left interface of C_{i+1}. The approximation in C_i is thus called u_R and the one in C_{i+1} is called u_L. Further, it seems reasonable to assume that those values can be derived from the values at the grid points Δx in front of and behind it.

$$F^n_{i+\frac{1}{2}} = \mathcal{F}(U^n_i, U^n_{i+1})$$

$\mathcal{F}$ is called the numerical flux function. The way we define this flux function defines the numerical method, and more importantly, it preserves the conservative nature of the PDE. This can be seen by summing over all cells; All the fluxes cancel out, except the fluxes at the extreme edges.

3 Convergence, Accuracy, and Stability

In this section, we briefly summarize standard concepts from the numerical analysis of finite volume methods that are required to interpret the behavior of the proposed scheme. These are included here to establish notation and to provide context for the analysis of the learned numerical flux later.

Throughout, we consider one–dimensional scalar conservation laws and assume sufficient regularity of the exact solution unless stated otherwise. Boundary effects are ignored for simplicity.

3.1 Cell Averages and Error Measurement

Finite volume methods evolve cell averages rather than pointwise values. Let $u(x, t)$ denote the exact solution and we have the cell average

$$U^n_i := \langle u(x_i, t^n) \rangle = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x, t^n)\, dx.$$

If $u(\cdot, t^n)$ is smooth, then a Taylor expansion shows that

$$U^n_i = u(x_i, t^n) + \mathcal{O}(\Delta x^2),$$

so for first- or second-order schemes (which is what we will cover) it is standard to measure error by comparing numerical cell averages with pointwise exact values at cell centers.

Global error at a fixed final time T is measured using discrete norms. For conservation laws, the discrete L^1 norm is relevant, as it remains meaningful even in the presence of discontinuities.

3.2 Consistency and Convergence

Let the finite volume method be written abstractly as

$$U^{n+1} = \mathcal{N}_{\Delta x,\Delta t}(u^n),$$

where $\mathcal{N}_{\Delta x,\Delta t}$ denotes one time step of the numerical scheme. The method is said to be *consistent* if, when applied to smooth exact solutions, the local truncation error vanishes as $\Delta t \to 0$.

A method is *convergent* if the numerical solution converges to the exact solution in an appropriate norm as the mesh is refined under a fixed CFL ratio. [8, 2]

3.3 Stability and Nonsmooth Solutions

Solutions of nonlinear hyperbolic conservation laws generally develop discontinuities even from smooth initial data. As a result, classical truncation error analysis alone is insufficient to guarantee convergence. Stability notions based on nonlinear estimates such as monotonicity, total-variation diminishing (TVD) properties, or discrete entropy inequalities [13] play a central role in ensuring convergence to the physically relevant entropy solution.

In practice, many robust finite volume schemes achieve stability by introducing numerical dissipation through the numerical flux, combined with suitable CFL restrictions on the time step.

3.4 Modified Equation Perspective

Additional insight into the accuracy and qualitative behavior of finite volume schemes can be obtained through *modified equation analysis* [15]. Rather than analyzing the discrete update directly, one seeks a continuous equation that the numerical solution satisfies up to higher-order terms.

For first-order monotone schemes, the leading truncation error often appears as an artificial diffusion term proportional to the mesh size,

$$u_t + f(u)_x = \partial_x(\nu_{\text{num}}(u)\, u_x) + \mathcal{O}(\Delta x^2),$$

where the numerical viscosity ν_{num} depends on the dissipation built into the numerical flux.

This perspective will play a central role in interpreting the neural-network-based numerical flux introduced in Section 4, where the dissipation is learned rather than prescribed by a worst-case bound.

3.5 Entropy and Entropy Solutions

Shock formation often leads to cases where classical solutions cease to exist globally, and solutions must be interpreted in a weak (distributional) sense. However, weak solutions are not unique, and additional admissibility criteria are required to select the physically relevant solution.

From [7], an *entropy–entropy flux pair* (η, q) consists of a convex entropy function $\eta : \mathbb{R} \to \mathbb{R}$ and an associated entropy flux $q : \mathbb{R} \to \mathbb{R}$ satisfying

$$q'(u) = \eta'(u)\, f'(u).$$

A weak solution $u(x,t)$ is said to satisfy the *entropy condition* if, for every convex entropy η, it holds in the distributional sense that

$$\eta(u)_t + q(u)_x \leq 0.$$

This inequality expresses the irreversibility of shock formation and rules out nonphysical weak solutions, such as expansion shocks.

For scalar conservation laws, it can be shown that there exists a unique weak solution satisfying all entropy inequalities; this solution is referred to as the *entropy solution* [3]. Entropy solutions coincide with classical solutions as long as the latter exist and provide the correct continuation beyond shock formation.

In the numerical setting, convergence to the entropy solution is not guaranteed by consistency and conservation alone. Additional nonlinear stability properties are required, such as monotonicity, total-variation-diminishing (TVD) behavior, or discrete entropy inequalities. Finite volume schemes that are conservative, consistent, and entropy stable are known to converge to the entropy solution under appropriate CFL conditions. These considerations play a central role in the design and analysis of robust numerical fluxes for nonlinear conservation laws.

4 Problem Setting and Numerical Fluxes

4.1 Governing Equation

We consider scalar conservation laws of the form

$$\partial_t u(x,t) + \partial_x f(u(x,t)) = \mu\, \partial_{xx} u(x,t), \tag{4}$$

posed on a one-dimensional periodic domain, where $u(x,t)$ denotes the conserved quantity, $f(u)$ is a nonlinear flux function, and $\mu \geq 0$ is a (possibly small) viscosity parameter. In this work, we focus primarily on the viscous Burgers equation [1], for which

$$f(u) = \frac{1}{2}u^2. \tag{5}$$

Equation (4) serves as a canonical model for nonlinear wave propagation and shock formation. The goal is to learn a flux function that does better than classical schemes over coarse grids.

4.2 Finite Volume Discretization

Integrating (4) over a control volume and approximating the fluxes at cell interfaces yields the semi-discrete finite volume scheme

$$\frac{d\langle u_i \rangle}{dt} = -\frac{1}{\Delta x}\left(F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}\right) + \mu\, D_{xx} u_i, \tag{6}$$

where $F_{i+\frac{1}{2}}$ is a numerical flux approximating $f(u(x_{i+\frac{1}{2}}, t))$, and D_{xx} denotes a consistent discrete Laplacian. The conservative structure of (6) ensures exact conservation at the discrete level.

4.3 Rusanov (Local Lax–Friedrichs) Flux

A widely used choice of numerical flux is the Rusanov [12], or local Lax–Friedrichs, flux, defined by

$$F^{\mathrm{Rus}}(u_L, u_R) = \frac{1}{2}\big(f(u_L) + f(u_R)\big) - \frac{1}{2}\alpha(u_L, u_R)\,(u_R - u_L), \tag{7}$$

where u_L and u_R denote the left and right states at a cell interface, and

$$\alpha(u_L, u_R) = \max\big(|f'(u_L)|, |f'(u_R)|\big) \tag{8}$$

is an upper bound on the local characteristic speed. For Burgers' equation, this reduces to $\alpha = \max(|u_L|, |u_R|)$.

The Rusanov flux is consistent and conservative, and introduces *numerical dissipation* via the α term, sufficient to guarantee nonlinear stability for scalar conservation laws under an appropriate CFL condition. However, this dissipation is designed as a worst-case bound and often leads to excessive numerical diffusion.

4.3.1 Consistency and Modified-Equation Analysis for Rusanov

We consider the scalar conservation law

$$u_t + f(u)_x = 0,$$

and a first-order conservative finite volume scheme of the form

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\Big(F(u_i^n, u_{i+1}^n) - F(u_{i-1}^n, u_i^n)\Big), \qquad \lambda := \frac{\Delta t}{\Delta x}.$$

Consistency from flux consistency. A numerical flux $F(u_L, u_R)$ is said to be *consistent* with the physical flux f if

$$F(u, u) = f(u) \qquad \forall u.$$

We sketch why this implies consistency of the spatial discretization in smooth regions. Let $u(x,t)$ be a smooth exact solution and set $u_i^n := u(x_i, t^n)$. At an interface $x_{i+\frac{1}{2}}$ define $u := u(x_{i+\frac{1}{2}}, t^n)$ and note that

$$u_i^n = u - \frac{\Delta x}{2}u_x + \mathcal{O}(\Delta x^2), \qquad u_{i+1}^n = u + \frac{\Delta x}{2}u_x + \mathcal{O}(\Delta x^2),$$

so $(u_i^n, u_{i+1}^n) = (u, u) + \mathcal{O}(\Delta x)$. Assuming F is C^1 near the diagonal $\{(u,u)\}$, a Taylor expansion gives

$$F(u_i^n, u_{i+1}^n) = F(u, u) + \mathcal{O}(\Delta x) = f(u) + \mathcal{O}(\Delta x),$$

and similarly $F(u_{i-1}^n, u_i^n) = f(u(x_{i-\frac{1}{2}}, t^n)) + \mathcal{O}(\Delta x)$. Therefore,

$$\begin{aligned}\frac{1}{\Delta x}\Big(F(u_i^n, u_{i+1}^n) - F(u_{i-1}^n, u_i^n)\Big) &= \frac{1}{\Delta x}\Big(f(u(x_{i+\frac{1}{2}}, t^n)) - f(u(x_{i-\frac{1}{2}}, t^n))\Big) + \mathcal{O}(\Delta x) \\ &= f(u)_x(x_i, t^n) + \mathcal{O}(\Delta x).\end{aligned}$$

Hence the finite volume spatial operator is consistent with $f(u)_x$ (and with a consistent time integrator yields a consistent fully discrete method).

Rusanov flux and modified equation. The Rusanov numerical flux is

$$F^{\mathrm{Rus}}(u_L, u_R) = \frac{1}{2}\big(f(u_L) + f(u_R)\big) - \frac{1}{2}\,\alpha_{LR}(u_R - u_L),$$

where $\alpha_{LR} \geq \max(|f'(u_L)|, |f'(u_R)|)$ is a bound on characteristic speed. To derive a clean modified equation, we assume we are in a smooth region and treat α_{LR} as *locally* constant to leading order: $\alpha_{i+\frac{1}{2}} = \alpha(u(x_{i+\frac{1}{2}}, t)) + \mathcal{O}(\Delta x)$.

Let $u_i^n = u(x_i, t^n)$ and expand

$$u_{i\pm1}^n = u \pm \Delta x\, u_x + \frac{\Delta x^2}{2} u_{xx} \pm \frac{\Delta x^3}{6} u_{xxx} + \mathcal{O}(\Delta x^4),$$

where u and its derivatives are evaluated at (x_i, t^n). Using Taylor expansions of $f(u_{i\pm1}^n)$ about u and collecting terms, one obtains the following expansion for the flux-difference operator:

$$-\frac{1}{\Delta x}\Big(F_{i+\frac{1}{2}}^{\mathrm{Rus}} - F_{i-\frac{1}{2}}^{\mathrm{Rus}}\Big) = -f'(u)\,u_x + \frac{\alpha\,\Delta x}{2}\,u_{xx} + \Delta x^2\left(-\frac{f'(u)}{6}u_{xxx} - \frac{f''(u)}{2}u_x u_{xx} - \frac{f'''(u)}{6}u_x^3\right) + \mathcal{O}(\Delta x^3).$$

Therefore, at the semi-discrete level, the Rusanov scheme is consistent with the *modified equation*

$$u_t + f(u)_x = \frac{\alpha\,\Delta x}{2}\,u_{xx} + \Delta x^2\left(-\frac{f'(u)}{6}u_{xxx} - \frac{f''(u)}{2}u_x u_{xx} - \frac{f'''(u)}{6}u_x^3\right) + \mathcal{O}(\Delta x^3).$$

In particular, the leading correction is a numerical viscosity term with effective coefficient

$$\nu_{\mathrm{num}}(u) \;=\; \frac{\alpha\,\Delta x}{2},$$

which explains the diffusive behavior of Rusanov on coarse grids.

Fully discrete (Forward Euler) correction. If Forward Euler time stepping is used, then the time discretization contributes an additional $O(\Delta t)$ term. Using

$$\frac{u(x_i, t^{n+1}) - u(x_i, t^n)}{\Delta t} = u_t + \frac{\Delta t}{2} u_{tt} + \mathcal{O}(\Delta t^2),$$

and eliminating u_{tt} via the PDE $u_t = -f'(u)u_x$, one finds

$$u_{tt} = f'(u)^2 u_{xx} + 2f'(u)f''(u)u_x^2.$$

Thus the leading-order fully discrete modified equation can be written as

$$u_t + f(u)_x = \frac{\alpha\,\Delta x}{2}\,u_{xx} - \frac{\Delta t}{2}\Big(f'(u)^2 u_{xx} + 2f'(u)f''(u)u_x^2\Big) + \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta t^2) + \mathcal{O}(\Delta x \Delta t).$$

Equivalently, in conservative form,

$$u_t + f(u)_x = \frac{\alpha\,\Delta x}{2}\,u_{xx} - \frac{\Delta t}{2}\,\partial_x\big(f'(u)^2 u_x\big) + \text{higher-order terms}.$$

4.4 Neural-Network-Based Numerical Flux

The central idea of this project is to replace the hand-designed dissipation coefficient in (7) with a learned, state-dependent quantity while preserving the conservative finite volume structure. Specifically, we define a neural-network-based numerical flux of the form

$$F^\theta(u_L, u_R; \mu) = \frac{1}{2}\big(f(u_L) + f(u_R)\big) - \frac{1}{2}\,\phi_\theta(u_L, u_R, \log\mu)\,(u_R - u_L), \qquad (9)$$

where ϕ_θ is a neural network with trainable parameters θ. The network takes as input the local left and right states and the viscosity parameter, and outputs a scalar numerical dissipation coefficient.
The neural network learns a data-informed numerical dissipation model that adapts to the local solution structure, with the goal of reducing excess diffusion while maintaining stability. Our modified equation for Rusanov also holds in a similar manner.

5 Analysis

In this section, we analyze the proposed neural-network-based finite volume method from the perspective of classical numerical analysis, following the framework developed in Chapter 8 of LeVeque [9]. The discussion emphasizes the roles of conservation, numerical dissipation, nonlinear stability, and modified-equation behavior in determining the accuracy and robustness of the scheme.

5.1 Consistency and Conservation

The finite volume update for the neural-network-based method is written in conservative flux-difference form,

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}\big(F^\theta_{i+\frac{1}{2}} - F^\theta_{i-\frac{1}{2}}\big) + \mu\,\Delta t\, D_{xx} u_i^n, \tag{10}$$

where F^θ is defined in (9). Since the numerical flux satisfies

$$F^\theta(u,u) = f(u) \tag{11}$$

for all admissible states u, the scheme is consistent with the inviscid flux. Furthermore, the telescoping structure of the flux differences implies exact discrete conservation under periodic boundary conditions. These properties hold independently of the choice of neural network parameters. Hence, we have consistency and conservativeness.

5.2 Nonlinear Stability Considerations

For scalar conservation laws, classical nonlinear stability results are closely tied to the monotonicity of the numerical flux. Fluxes of the Rusanov or local Lax–Friedrichs type,

$$F(u_L,u_R) = \frac{1}{2}\big(f(u_L) + f(u_R)\big) - \frac{1}{2}a(u_L,u_R)(u_R - u_L), \tag{12}$$

are known to be monotone [2, 5] under a suitable CFL condition provided that the dissipation coefficient $a(u_L,u_R)$ dominates the local characteristic speed. For Burgers' equation, a sufficient condition is

$$a(u_L,u_R) \geq \max(|u_L|,|u_R|). \tag{13}$$

Under such conditions, the resulting finite volume method is total-variation-diminishing (TVD), satisfies a discrete maximum principle, and converges to the entropy solution as $\Delta x \to 0$ per [5].

In the present work, the NN-based dissipation coefficient $\phi_\theta(u_L,u_R,\log\mu)$ is not explicitly constrained to satisfy a global wave-speed bound. As a result, the scheme does not, in general, fall into the class of monotone methods for which unconditional nonlinear stability can be proven. Instead, stability is ensured through a combination of mechanisms commonly employed

in high-resolution schemes, including CFL-controlled time stepping, the use of strong-stability-preserving Runge–Kutta methods, and explicit monitoring of solution diagnostics such as total variation, entropy production, and boundedness.

From the perspective of LeVeque's analysis, the learned flux may be interpreted as introducing an adaptive, nonlinear dissipation mechanism analogous to flux limiters in classical high-resolution finite volume methods. In smooth regions, the dissipation may be reduced to improve accuracy, while in regions of strong gradients or shocks, the dissipation increases to suppress spurious oscillations.

5.3 Accuracy and Modified Equation Interpretation

The accuracy of first-order finite volume schemes is often understood through a modified-equation analysis. We have shown that for schemes such as the Rusanov method, the leading truncation error can be interpreted as an artificial diffusion term proportional to the mesh size,

$$\nu_{\text{num}} = \frac{\alpha \Delta x}{2} \tag{14}$$

Excess numerical viscosity leads to shock smearing and reduced accuracy on coarse grids.

In the neural-network-based method, the effective numerical viscosity is state-dependent and given, up to leading order, by

$$\nu_{\text{num}}^{\theta} \sim \frac{\Delta x}{2}\, \phi_\theta(u_L, u_R, \log \mu). \tag{15}$$

By learning ϕ_θ from data, the method adapts the local numerical dissipation to typical solution structures rather than relying on worst-case bounds. As emphasized in Chapter 8 of [9], such adaptive dissipation mechanisms are the key to improving resolution while maintaining nonlinear stability. The resulting scheme retains the same formal order of accuracy as the baseline method, but with a reduced error constant, leading to significantly improved coarse-grid performance.

5.4 Convergence Considerations

For scalar conservation laws, monotone conservative schemes are known to converge to the entropy solution as the mesh is refined. If the learned dissi-

pation coefficient satisfies a suitable wave-speed bound uniformly, then the neural-network-based method falls into this class and inherits the corresponding convergence guarantees.

In the absence of an explicit monotonicity constraint, convergence of the neural-network-based scheme is assessed empirically through grid refinement studies and long-time simulations. Numerical experiments indicate that the method converges to the same weak solution as the baseline Rusanov scheme, with reduced error at fixed coarse resolution. This behavior is consistent with the interpretation of the neural network as a mechanism for reducing excess numerical viscosity while preserving the conservative structure of the discretization.

5.5 Designing a Neural Network that gives strong guarantees

Our current NN-FVM is conservative and consistent. We assess it's stability empirically via entropy checks and TVD. It is however, possible to design NNs which learn fluxes that can give us stronger guarantees.

5.5.1 Nonlinear stability via monotonicity (TVD) conditions

For scalar conservation laws, a classical route to nonlinear stability is *monotonicity* of the one-step update map. A sufficient condition is that the numerical flux is nondecreasing in its left argument and nonincreasing in its right argument. Since our scheme is similar to Rusanov, which is monotone, our expression is identical to (13).

$$\phi_\theta(u_L, u_R) \;\geq\; \max\big(|f'(u_L)|, |f'(u_R)|\big).$$

Under this condition and the CFL constraint

$$\lambda \max_i \phi_{\theta, i+\frac{1}{2}} \leq 1,$$

the resulting scheme is monotone, satisfies a discrete maximum principle, and is TVD. In particular, it admits uniform ℓ^∞ bounds and total-variation bounds, providing nonlinear stability in the sense relevant for hyperbolic conservation laws.

How to build an NN to enforce this. :
A practical way to embed the wave-speed bound into the architecture is to parameterize

$$\phi_\theta(u_L, u_R) = \alpha(u_L, u_R) + \mathrm{softplus}\big(g_\theta(u_L, u_R)\big),$$
$$\alpha(u_L, u_R) = \max\big(|f'(u_L)|, |f'(u_R)|\big),$$

where g_θ is an unconstrained neural network and $\mathrm{softplus}(z) = \log(1 + e^z) \geq 0$. This guarantees $\phi_\theta \geq \alpha$ *for all inputs*, so the learned method inherits the same monotonicity/TVD stability mechanism as Rusanov, while still allowing the network to adapt the dissipation in a controlled way (e.g. it can be close to α in smooth regions and larger near steep gradients). If one instead wishes to *reduce* dissipation while retaining a safety margin, one can use

$$\phi_\theta(u_L, u_R) = \alpha(u_L, u_R) - \sigma_\theta(u_L, u_R), \qquad 0 \leq \sigma_\theta \leq \eta\,\alpha, \;\; 0 < \eta < 1,$$

implemented by $\sigma_\theta = \eta\,\alpha\,\mathrm{sigmoid}(g_\theta)$ so that $\phi_\theta \geq (1-\eta)\alpha$. This relaxes the worst-case dissipation while preserving a uniform lower bound proportional to the characteristic speed.

5.5.2 Convergence: conservative + consistent + (entropy/TVD) stability

A central convergence theorem for scalar conservation laws states that *any conservative, consistent, monotone (hence TVD) scheme converges, as* $\Delta x \to 0$ *under an appropriate CFL condition, to the unique entropy solution* [2]. Therefore if we have:

1. **Conservation** (flux-difference form) gives the correct weak formulation in the limit;
2. **Consistency** $F^\theta(u, u) = f(u)$ ensures the truncation error vanishes for smooth solutions;
3. **Nonlinear stability** (e.g. monotonicity/TVD or a discrete entropy inequality) provides compactness and selects the entropy solution.

Therefore, to place the learned scheme within a class with rigorous convergence guarantees, it suffices to design ϕ_θ so that the induced flux is monotone (or entropy stable).

How to build an NN to promote convergence beyond monotonicity. Even when strict monotonicity is relaxed (to reduce numerical diffusion), one can still promote entropy stability [13] by augmenting training with penalties that enforce discrete entropy dissipation. For a convex entropy $\eta(u)$ with entropy flux $q(u)$ satisfying $q'(u) = \eta'(u)f'(u)$, a common discrete requirement is an inequality of the form

$$\eta(u_i^{n+1}) \;\le\; \eta(u_i^n) - \lambda\Big(Q^n_{i+\frac{1}{2}} - Q^n_{i-\frac{1}{2}}\Big),$$

for a suitable numerical entropy flux $Q^n_{i+\frac{1}{2}}$ consistent with $q(u)$. Architecturally, one may enforce a *minimum* dissipation level,

$$\phi_\theta(u_L, u_R) \ge \alpha(u_L, u_R),$$

to recover the classical monotone/entropy-dissipative regime, or enforce a *soft* version via a loss term such as

$$\mathcal{L}_{\text{stab}} = \sum_{i,n} \Big[\, \max\{0, \alpha^n_{i+\frac{1}{2}} - \phi^n_{\theta,i+\frac{1}{2}}\}\Big]^2,$$

which discourages violations of the wave-speed bound while still permitting occasional controlled reductions if they improve accuracy and do not destabilize rollouts. Similarly, one may penalize growth of total variation

$$\mathcal{L}_{\text{TV}} = \sum_n \max\Big\{0, \text{TV}(u^{n+1}) - \text{TV}(u^n)\Big\}, \qquad \text{TV}(u^n) := \sum_i |u^n_{i+1} - u^n_i|,$$

to bias the learned flux toward TVD-like behavior.

5.6 Summary

From the viewpoint of classical finite volume analysis, the proposed neural-network-based flux defines a consistent and conservative scheme with a learned, adaptive dissipation mechanism. While unconditional nonlinear stability and convergence theorems require additional structural constraints, the method closely parallels high-resolution schemes discussed by LeVeque, in which accuracy improvements are achieved by controlling numerical dissipation while maintaining stability through CFL restrictions and careful time integration.

6 Data Generation and Training Procedure

6.1 Reference Solution Generation

Training data are generated using high-resolution numerical solutions of the governing equation (4). For each training case, the viscous Burgers equation is solved on a fine spatial grid using a high-order shock-capturing scheme (WENO5), which is a WENO scheme [6, 10], combined with a stable explicit time integrator. These fine-grid solutions serve as numerical references and are assumed to accurately resolve the relevant solution features.

Importantly, the reference solutions are not used directly during training. Instead, they are projected onto a coarse grid by averaging over fine-grid cells to obtain reference coarse cell averages. This ensures that all training targets are defined at the same resolution as the learned finite volume scheme, and avoids introducing information unavailable to a coarse-grid solver.

6.2 Coarse-Grid Targets

Let $u^{\text{fine}}(x,t)$ denote the fine-grid reference solution. The corresponding coarse-grid cell averages are defined by

$$u_i^{\text{ref}}(t) = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u^{\text{fine}}(x,t)\, dx. \tag{16}$$

These coarse averages represent the solution that an ideal coarse-grid finite volume method would produce if numerical dissipation were perfectly calibrated.

Training data consist of pairs of coarse-grid states and their evolved values after one or more time steps, computed from the reference coarse averages. No pointwise fine-scale information is provided to the neural network.

6.3 Neural Network Architecture

The neural network is used to parameterize the numerical dissipation coefficient in the flux (9). It is implemented as a lightweight fully connected feedforward network

$$\phi_\theta : (u_L, u_R, \log \mu) \mapsto \mathbb{R},$$

with 2 hidden layers with 64 nodes each and smooth (tanh) activation functions. The input features consist of the left and right interface states and

the logarithm of the viscosity parameter, which together encode the local solution structure and regime of interest. The output is a scalar dissipation coefficient that modulates the diffusive term in the numerical flux.

The architecture is deliberately kept low-dimensional and local, reflecting the fact that numerical fluxes depend only on interface-adjacent states. No spatial context beyond the immediate interface is provided to the network, ensuring that the learned flux remains local and compatible with the finite volume framework.

6.4 Staged Training Strategy

Training is performed using a staged, or curriculum-based, strategy designed to promote stability and generalization of the learned flux when embedded in a time-stepping finite volume solver. All stages share the same underlying discretization, time integrator, and CFL constraints.

Stage I: Local Flux Regularization. In the initial stage, the network is trained on randomly sampled interface states to encourage admissible local behavior of the dissipation coefficient. Regularization terms penalize excessively large or negative dissipation values and promote smooth dependence on the inputs. This stage serves to initialize the network in a regime consistent with classical numerical flux behavior.

$$\mathcal{L}_{\text{stage 1}} = \underbrace{||\nabla \phi_\theta||^2}_{\mathcal{L}_{\text{grad}}} + \underbrace{\text{relu}\left(\phi_\theta(u_R - u_L)\right)}_{\mathcal{L}_{\text{diss}}}$$

Stage II: One-Step Operator Matching. In the second stage, the network is trained to match the evolution of coarse-grid cell averages over a single time step. Given a coarse-grid state $\mathbf{u}^n$, the learned finite volume operator produces an update

$$\mathbf{u}_\theta^{n+1} = \mathcal{T}^\theta_{\Delta x, \Delta t}(\mathbf{u}^n),$$

which is compared against a reference coarse-grid update $\mathbf{u}_{\text{ref}}^{n+1}$ obtained from high-resolution solutions, effectively making the inaccuracy in the physics our error, as seen in PINNs [11]. The primary loss is an L^2 discrepancy between these states.

Stage III: Short-Horizon Rollout Training. In the final stage, training incorporates short multi-step rollouts of the finite volume solver. The loss penalizes accumulated discrepancies between the learned and reference solutions over a small number of consecutive time steps. This stage mitigates error accumulation and encourages the learned flux to remain stable under repeated application.

6.5 Loss Functions

Across stages, the total training loss is a weighted combination of the form

$$\mathcal{L} = \lambda_{\text{reg},1}\mathcal{L}_{\text{stage 1}} + \lambda_{\text{reg},2}\,\mathcal{L}_{\text{stage 2}}, +\lambda_{\text{reg},3}\mathcal{L}_{\text{stage 3}} \tag{17}$$

The weights are adjusted between stages to gradually shift emphasis from local behavior to long-horizon accuracy and stability.

6.6 Summary

The combination of a local, low-capacity neural network and a staged training strategy ensures that learning remains tightly coupled to the finite volume operator. Rather than approximating the PDE solution directly, the network learns a calibrated numerical dissipation model that can be robustly embedded within a classical solver.

Throughout training, the underlying discretization, time step selection, and CFL constraints are kept identical to those of the baseline Rusanov scheme. As a result, any observed performance gains can be attributed solely to the learned numerical flux rather than changes in the solver infrastructure.

7 Results

This section summarizes the empirical performance of the proposed neural-network-based flux (NN-FVM) relative to the classical Rusanov finite-volume baseline. Unless stated otherwise, all experiments use the same spatial discretization with periodic BCs, SSP-RK3 time integration, and identical CFL logic; the only change between methods is the numerical flux. Reference solutions are obtained from high-resolution simulations (spatial discritization with 1000 points) and projected onto the evaluation grid (spatial discritization with 100 points) for fair comparison.

7.1 Qualitative profiles

We first compare solution profiles produced by the NN-FVM and the Rusanov method at representative times. While the differences are small, it can be seen that the NN-FVM does a better job at approximating the true solution than the FVM that's using Rusanov's flux.

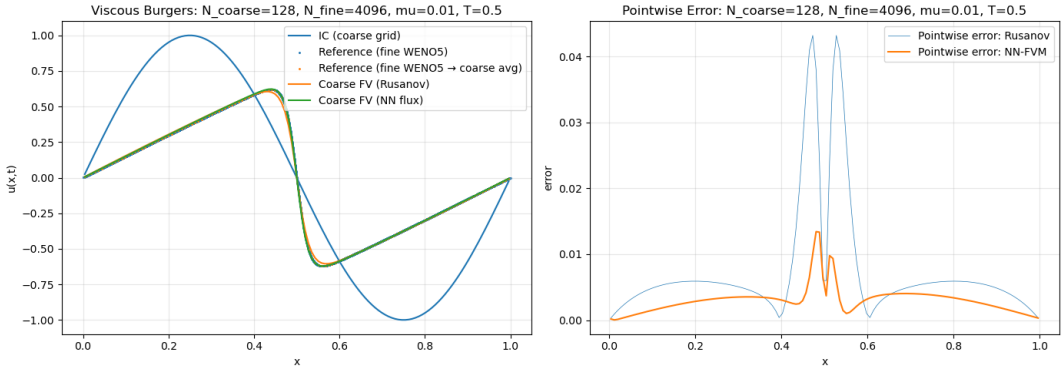

Figure 1: Representative solution profiles at selected times for the NN-FVM and Rusanov baseline, compared to a high-resolution reference projected onto the coarse evaluation grid, along with a plot of the pointwise difference.

7.2 Behaviour vs discretization

To compare overall behaviour, we consider the median L^2 error ratio to the discretization. N represents the number of points in the spatial mesh and K represents the number of time steps in the simulation The distribution of L^2 error ratios for a 186 trajectories is shown below. With NN-FVM, in this case, has a win rate of 100%, i.e., the error is better for the NN-FVM than the FVM in every case

7.3 Behviour vs viscosity

NN-FVM manages to outperform FVM irrespective of viscosity; whether it came from within the initial training set or outside of it.

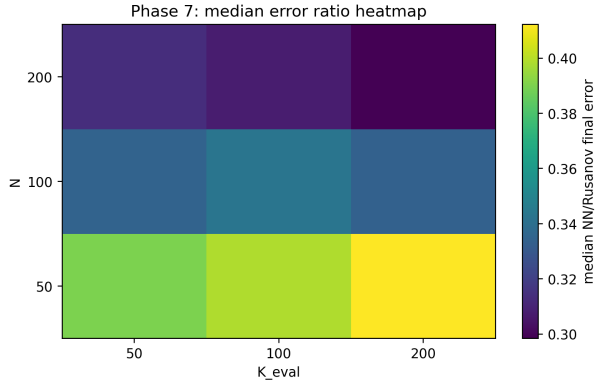

Figure 2: The median error ratio for different discretizations. Even in the coarsest case, the error in NN-FVM stays much less than the error in FVM

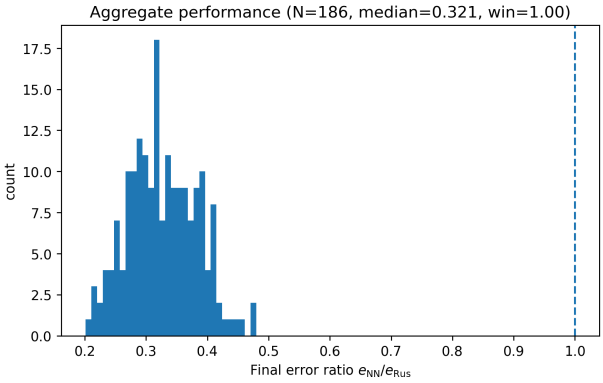

Figure 3: The NN-FVM has lower L^2 error in every run, and it outperforms the FVM's error by a factor of > 2

7.4 Error growth and stability diagnostics

We present stability diagnostics like the variation of the total variation, and the entropy/mass drift for a representative case and the worst case in Figure 5

We also have the error L^2 error growth vs time and a time-space heatmap comparing the absolute errors of the NN-FVM and Rusanov at different time and space points in Figure

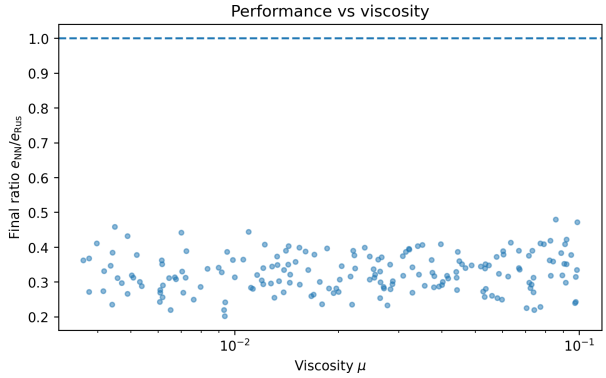

Figure 4: The error ratios for different values of μ. The lower values are harder and more prone to shock formation than the higher values

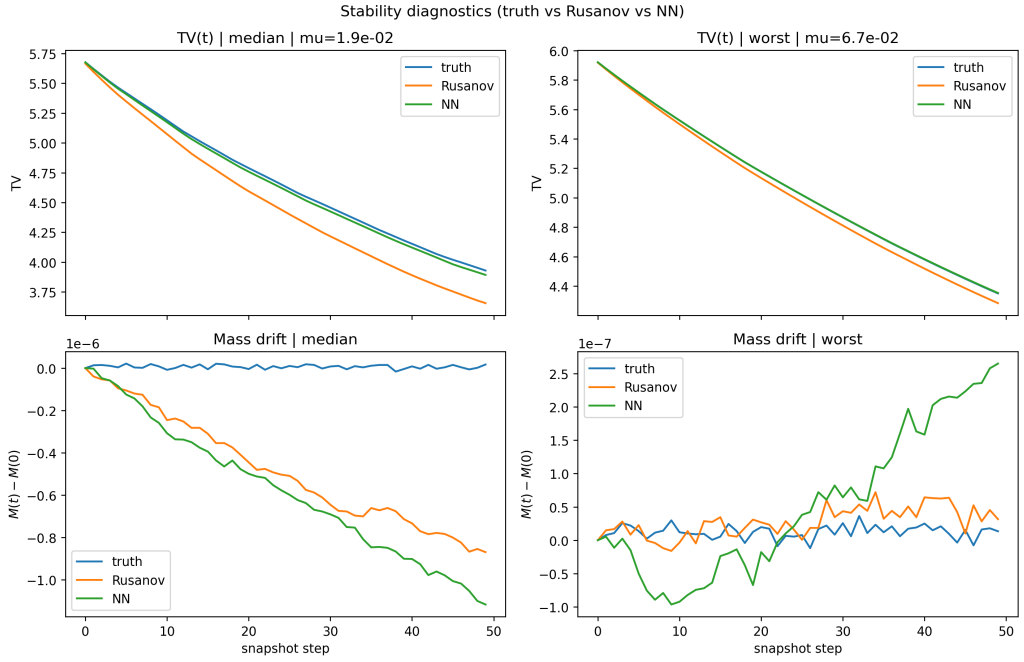

Figure 5: The diminishing TV, along with convergence and consistency, implies that the numerical solution is entropically stable

7.5 Summary of Findings

Overall, the results indicate that replacing the hand-designed Rusanov dissipation with a learned, state-dependent alternative yields a consistent im-

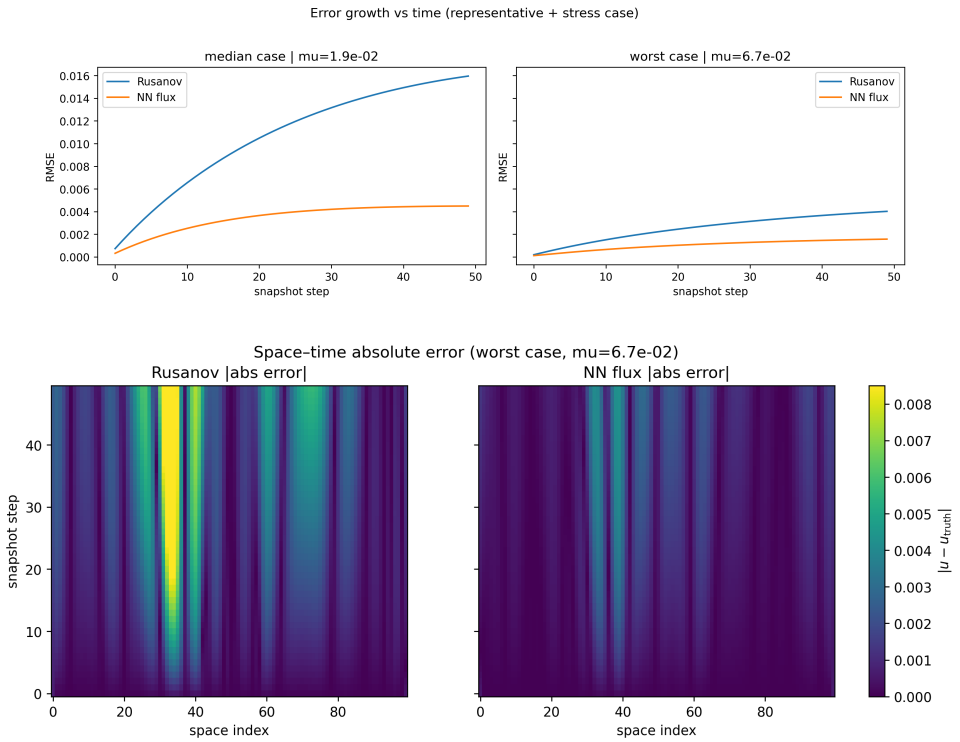

Figure 6: Error growth in a representative sample and worst case along with a space-time diagram for absolute error of both schemes for the worst case

provement in coarse-grid accuracy. The gains are most pronounced in regimes where the Rusanov method exhibits excess numerical diffusion. Importantly, these improvements are obtained without altering the conservative finite-volume update or the time-stepping scheme, suggesting that the neural network primarily acts as a calibrated numerical viscosity model embedded within a classical solver.

8 Concluding Remarks

In this project, we investigated the use of Neural Networks to improve the numerical solution of partial differential equations arising in fluid dynamics, while preserving the fundamental structure and guarantees of classical finite volume methods.

Starting from a standard finite volume discretization of the viscous Burgers equation, we reviewed the construction and theoretical properties of the Rusanov flux. Motivated by the observation that the leading error on coarse grids is dominated by excess numerical viscosity, we introduced a neural-network-based flux that replaces the hand-designed dissipation coefficient with a learned, state-dependent alternative, while leaving the discretization, time integration, and CFL constraints unchanged.

The resulting method is exactly conservative and consistent by construction. Although the learned flux does not, in its unconstrained form, admit unconditional monotonicity or entropy stability guarantees, its behavior can be interpreted through the lens of classical high-resolution finite volume schemes. In particular, the neural network acts as an adaptive dissipation mechanism analogous to flux limiters, reducing numerical diffusion in smooth regions while increasing dissipation near sharp gradients to maintain stability. Numerical experiments demonstrate improved coarse-grid accuracy, reduced error growth, and stable long-time behavior relative to the Rusanov scheme.

From a numerical analysis perspective, the improvements observed can be understood as a reduction in the effective numerical viscosity appearing in the modified equation associated with the finite volume method. The learned flux therefore yields a smaller truncation error constant without altering the formal order of accuracy, consistent with established theory for nonlinear conservation laws. Convergence is supported empirically through grid refinement studies, and classical convergence theorems apply under additional structural constraints on the learned dissipation.

There are several promising directions for future research. Enforcing monotonicity or entropy-stability constraints within the learning architecture would enable rigorous convergence proofs, while extending the approach to entropy-stable flux formulations provides a natural pathway to systems of conservation laws such as the shallow water equations. More broadly, this project illustrates how machine learning can be used not as a replacement for numerical analysis, but as a tool for improving classical schemes within well-understood frameworks.

Code and Data Availability

All simulation code, numerical routines, and plotting scripts used in this study are available in the following GitHub repository:
`https://github.com/Tushnimy/Neural_flux_for_coarse_grid_accuracy`

References

[1] J. Burgers, *A mathematical model illustrating the theory of turbulence*, vol. 1 of Advances in Applied Mechanics, Elsevier, 1948, pp. 171–199.

[2] M. G. Crandall and A. Majda, *Monotone difference approximations for scalar conservation laws*, Mathematics of Computation, 34 (1980), pp. 1–21.

[3] C. M. Dafermos, *Hyperbolic Conservation Laws in Continuum Physics*, vol. 325 of Grundlehren der mathematischen Wissenschaften, Springer, 4 ed., 2016.

[4] S. K. Godunov and I. O. Bohachevsky, *Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics*, 1959.

[5] A. Harten, *High resolution schemes for hyperbolic conservation laws*, Journal of Computational Physics, 49 (1983), pp. 357–393.

[6] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted eno schemes*, Journal of Computational Physics, 126 (1996), pp. 202–228.

[7] S. N. Kruzhkov, *First order quasilinear equations with several independent variables*, Matematicheskii Sbornik, 81 (1970), pp. 228–255.

[8] P. Lax and B. Wendroff, *Systems of conservation laws*, Communications on Pure and Applied Mathematics, 13 (1960), pp. 217–237.

[9] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.

[10] X.-D. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics, 115 (1994), pp. 200–212.

[11] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations*, 2017.

[12] V. Rusanov, *The calculation of the interaction of non-stationary shock waves and obstacles*, Ussr Computational Mathematics and Mathematical Physics, 1 (1962), pp. 304–320.

[13] E. Tadmor, *Numerical viscosity of entropy stable schemes for systems of conservation laws. final report*, tech. rep., National Aeronautics and Space Administration, Hampton, VA (USA). Langley Research Center, 10 1985.

[14] E. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, 01 2009.

[15] R. Warming and B. Hyett, *The modified equation approach to the stability and accuracy analysis of finite-difference methods*, Journal of Computational Physics, 14 (1974), pp. 159–179.