A Tutorial on Neural Network-Based Solvers for Hyperbolic Conservation Laws: Supervised vs. Unsupervised Learning, and Applications to Traffic Modeling

Alexi Canesse† ALEXI.CANESSE@BERKELEY.EDU
Zhe Fu† ZHEFU@BERKELEY.EDU
Nathan Lichtlé† LICHTLE@BERKELEY.EDU
Hossein Nick Zinat Matin† H-MATIN@BERKELEY.EDU
Zihe Liu ZIHE_LIU@BERKELEY.EDU
Maria Laura Delle Monache MLDELLEMONACHE@BERKELEY.EDU
Alexandre M. Bayen BAYEN@BERKELEY.EDU
University of California, Berkeley.

Editors: G. Pappas, P. Ravikumar, S. A. Seshia

Abstract

Neural networks (NNs) are powerful tools for solving complex partial differential equations (PDEs) with high accuracy. However, many NN-based solvers are designed as general-purpose models or lack theoretical grounding, limiting their ability to capture essential solution properties such as regularity, conservation, and entropy conditions. This issue is especially critical for hyperbolic conservation laws, which govern wave propagation and shock formation, and are among the most challenging PDEs to solve accurately. This tutorial examines both supervised and unsupervised NN-based solvers from computational and theoretical perspectives, with a focus on NN-based finite volume methods (FVMs) tailored to conservation laws. In the supervised setting, NN solvers learn from available solution data, such as Riemann problems, to capture characteristic solution structures, while the unsupervised approach employs a weak formulation loss to enforce the correct weak solution behavior. In practice, both the supervised and unsupervised variants tend to learn the entropic solution, effectively handling discontinuities and shocks, and outperforming comparable numerical schemes in accuracy. This tutorial aims to provide a deeper understanding of NN-based solvers for PDEs and to present structure-preserving neural methods for scientific computing.

Keywords: Neural networks, PDEs, Unsupervised Learning, Weak Formulation, PINNs, Conservation Laws.

1. Introduction and Related Work

In this tutorial, we provide an overview of NN-based approaches for approximating solutions to PDEs, with a particular focus on hyperbolic equations. Specifically: (i) We review both supervised and unsupervised learning methods, highlighting their strengths and limitations. (ii) We examine how core analytical properties of PDE solutions, such as regularity, stability, and conservation laws, influence the design of NN architectures. Particular attention is given to the challenges posed by hyperbolic PDEs, which often exhibit discontinuities and other irregular behaviors. (iii) We also discuss recent hybrid approaches that integrate classical numerical schemes with NNs to mitigate

†. These authors contributed equally to this tutorial and are listed in alphabetical order.

discretization-induced errors. These methods represent important progress toward bridging the gap between PDE theory and NN-based solvers, and provide insight into how learning-based methods can approximate true PDE solutions.

PDEs model a vast range of physical and engineering systems, from fluid dynamics to electromagnetism, making it crucial to approximate their solutions efficiently and accurately (Evans, 2022). We start with a general time-evolutionary PDE in the form of

$$\begin{cases} \partial_t u + \boldsymbol{L}u = \xi, & (t,x) \in [0,T] \times \mathcal{U}, \\ u(0,x) = u_\circ(x), & x \in \mathcal{U}, \\ u(t,x) = g(t), & (t,x) \in [0,T] \times \partial\mathcal{U}, \end{cases} \tag{1}$$

where $t \in [0,T]$ is the time variable, with final time $T > 0$, $x \in \mathcal{U} \subset \mathbb{R}^d$ is the spatial variable, $\boldsymbol{L}$ is a differential operator acting on u, ξ is a source term, $u_\circ$ and g are the initial and boundary value functions, $\partial\mathcal{U}$ denotes the boundary of the spatial domain $\mathcal{U}$, and the unknown function $u : [0,T] \times \mathcal{U} \to \mathbb{R}^n$ (with $n = 1$ for scalar equations and $n > 1$ for systems of PDEs) is a solution of (1) if it satisfies the equations in some appropriate sense (e.g., classically or weakly). Closed form solutions of model (1) exist for only a few simple models, such as the Lax-Hopf formula for conservation laws and Hamilton-Jacobi equations (Claudel and Bayen, 2010), and in the majority of real-world problems one needs to approximate the solution through numerical methods.

Classical numerical schemes, including finite difference, finite volume (FV), and finite element (FE) methods (LeVeque, 2002; Cockburn et al., 2012; Muftu, 2022), approximate PDE solutions through spatial and temporal discretization. These methods have been extensively studied and, in some cases, provide well-established convergence guarantees. However, they are limited by discretization errors that propagate through computations and accumulate over time. Moreover, in high-dimensional settings, numerical methods suffer from the curse of dimensionality, leading to high computational costs (Cohen and DeVore, 2015).

NNs are a promising alternative for approximating PDE solutions (Sirignano and Spiliopoulos, 2018; Sirignano et al., 2020; Fan et al., 2020), offering flexibility in handling high-dimensional systems, integrating large datasets (Shi et al., 2021), and exploring solutions over parameter spaces in mesh-free representations. However, despite these advantages, NN-based approaches can struggle to preserve fundamental solution properties, such as stability, conservation laws, and uniqueness, leading to potential accuracy loss and instability (Kutyniok et al., 2022; Ruthotto and Haber, 2020).

Various NN architectures have been proposed for solving PDEs. In **supervised learning**, the model can be trained on PDE data consisting of initial problems and their exact (or high-fidelity) solutions, using regression-based loss functions (Li et al., 2020; Lu et al., 2021). In **unsupervised learning**, the residual of the PDE is minimized in the loss of a Physics-Informed NN (PINN) (Raissi et al., 2017, 2019), allowing the model to learn PDEs without requiring any data. Finally, **semi-supervised learning** combines both methods, balancing a data-driven term and a physics-informed regularization term in the loss function (Shi et al., 2022; Sharpless et al., 2025).

1.1. Supervised Learning Approaches

In supervised learning, a NN is trained to approximate a PDE by minimizing a regression loss based on a dataset of exact or high-fidelity numerical solutions. The main strength of the supervised learning approach is that one does not require any knowledge of PDE equations; one only needs sufficient data. The training dataset consists of pairs of input conditions (such as initial conditions

(ICs) and boundary conditions) and their corresponding exact solutions. The NN then learns to minimize the difference between its predicted solutions and the ground truths.

This approach allows the model to learn a mapping from input conditions to solutions, effectively acting as a solver of the PDE it is trained on. Different deep learning approaches can be leveraged to accomplish this. One of the earliest applications was convolutional NNs (CNNs) (LeCun et al., 1998) for surrogate modeling (Zhu and Zabaras, 2018), where the authors demonstrated that deep CNNs could learn complex solution mappings with high accuracy while reducing computational costs. Graph NNs (GNNs) (Bronstein et al., 2017) have been introduced as a powerful architecture for PDEs on unstructured grids (Brandstetter et al., 2022), which operate on graph-based representations, making them well-suited for irregular mesh discretizations. Beyond traditional neural architectures, neural operators have gained significant attention for their ability to learn mappings between infinite-dimensional function spaces. More precisely, let $\mathcal{A}$ and $\mathcal{B}$ be separable Banach spaces. Suppose we have observations (a_j, b_j) where $a_j \sim \lambda$ is an independent and identically distributed sequence from a probability measure λ supported on $\mathcal{A}$, and $b_j = \boldsymbol{F}^*(a_j)$. The goal is to construct the solution operator $\boldsymbol{F}^*$ of the parametric PDE. Let Θ be a finite dimensional parameter space and $\boldsymbol{F}_\theta : \mathcal{A} \to \mathcal{B}$ be a parametric (nonlinear) operator. The goal is then to find $\theta \in \Theta$ such that $\boldsymbol{F}_\theta \approx \boldsymbol{F}^*$. This can then be described by

$$\min_{\theta \in \Theta} \mathbb{E}_{a \sim \lambda}[C(\boldsymbol{F}_\theta(a), \boldsymbol{F}^*(a))]$$

where $C : \mathcal{B} \times \mathcal{B} \to \mathbb{R}$ is a cost function. Two established methods that follow this concept are:

Fourier Neural Operator (FNO) (Li et al., 2020), which learns an operator in Fourier space, capturing complex frequency-domain patterns. Unlike traditional grid-based PDE solvers, it maps ICs to solutions directly, offering potential speedups and resolution invariance, enabling zero-shot super-resolution. However, it demands large, high-quality datasets, especially for complex PDEs.

Deep Operator Network (DeepONet) (Lu et al., 2021), which learns nonlinear operators, including PDE solutions, using two subnetworks: a branch network encoding ICs and a trunk network encoding output locations. Given an IC and a query point, DeepONet directly predicts the solution at that point, enabling efficient, grid-free evaluations.

1.2. Unsupervised Learning Approaches

Unsupervised learning is useful when data is scarce or costly to generate. For PDEs, explicit analytical solutions are often unavailable and numerical solvers are computationally expensive, making it infeasible to generate large-scale training datasets (Guo et al., 2020). To overcome these challenges, unsupervised neural solvers aim to solve PDEs without relying on solution data, instead enforcing physical constraints directly within the loss function.

One of the most established unsupervised learning approaches for solving PDEs is PINNs (Raissi et al., 2019). PINNs can be constructed from most NN-based PDE solvers by augmenting or replacing their supervised loss. For conservation laws equations, the loss function reads:

$$\mathcal{L}(\theta) = \|\partial_t \hat{u}_\theta + \partial_x \psi(\hat{u}_\theta)\|^2_{L^2([0,T]\times\mathcal{U})},$$

where $\hat{u}_\theta$ is the NN approximation, θ is the NN parameters, and ψ is a flow function. The most common framework is based on NNs that take x, t or any other variables as input, and output the solution at the given point. In this case, the loss can be easily computed and backpropagated through automatic differentiation (Raissi et al., 2017).

While PINNs have been successfully applied to elliptic and parabolic equations (Shin et al., 2020), their performance deteriorates when applied to hyperbolic PDEs such as conservation laws, which contain shocks and expansion waves. Mishra and Molinaro (2023); De Ryck et al. (2024) provided rigorous theoretical analysis showing that standard PINNs fail to properly approximate weak solutions for highly irregular hyperbolic problems (see Section 2) that contain high-frequency oscillations or sharp discontinuities. The primary reason is that automatic differentiation struggles with discontinuous functions, leading to large approximation errors in shock-dominated regimes. Relatedly, recent work by Park et al. (2024) highlights a broader structural issue in PINNs, namely, their inability to regulate solution derivatives, which may also contribute to instability in challenging regimes. Although their proposed variable splitting strategy targets second-order PDEs, it underscores the need for architectural innovations to enhance robustness across PDE classes.

To address these issues, weak formulations have been proposed as an alternative loss function for unsupervised learning in hyperbolic PDEs (Yang and Foster, 2021; Shang et al., 2023):

$$\mathcal{L}(\theta) = \mathbb{E}\left[\left(\int_{[0,T]}\int_{\mathcal{U}} \hat{u}_\theta \partial_t \varphi + \int_{[0,T]}\int_{\mathcal{U}} \psi(\hat{u}_\theta)\partial_x \varphi\right)^2\right], \tag{2}$$

where the expectation is taken over the probability space $(\Phi, \mathcal{B}, \mathbb{P})$ with $\Phi = C_c^1([0,T]\times\mathcal{U};\mathbb{R})$, Borel σ-algebra $\mathcal{B}$, and probability measure $\mathbb{P}$. In practice, the set Φ is composed of compactly supported polynomials. *The key advantage* of the weak formulation (2) is that the differential operators are only applied to the test functions, which are chosen to be sufficiently smooth to avoid noisy numerical differentiation. This approach aligns with classical numerical schemes, such as FVMs and discontinuous Galerkin methods, which rely on weak formulations for stability and accuracy in hyperbolic problems (Cockburn and Shu, 1998).

Recent advancements in neural weak solvers have further improved the robustness of unsupervised PDE learning. Neural Galerkin methods (Bruna et al., 2024) integrate variational principles into neural architectures, enabling more structured representations of PDE solutions. Future research aims to explore adaptive test function selection, hybrid weak-supervised approaches, and memory-efficient neural architectures for high-dimensional PDEs.

2. Structural-Based NN Solvers

While current approaches primarily focus on the computational efficiency of NN solvers, they often fail to preserve the theoretical properties of specific PDE classes (the differential operator $\boldsymbol{L}$ in (1)). Ensuring the convergence of NN solvers requires designing them with key analytical PDE properties in mind; e.g., the functional space of existence (regularity), convergence to the unique entropy solution, stability, conservation of quantities (such as mass), bounds, and physical constraints.

NN approximation is generally significantly more accurate for PDEs with regular solutions. Hyperbolic equations present a unique challenge due to singularities such as shocks and expansion waves, making solution approximation difficult near these points. NN-based FVMs aim to address this, as they are designed to preserve essential mathematical properties of PDEs. NN solvers can learn to mitigate accumulation errors arising from discretization in classical numerical schemes. Following the FVM structure can help improve the accuracy and stability of NN solvers, particularly in capturing critical solution behaviors inherent to hyperbolic systems. We begin by reviewing fundamental concepts of this PDE class to highlight its challenges.

Equations of Conservation Laws. We are interested in conservation laws of the form:

$$\begin{cases} \partial_t u(t,x) + \partial_x[\psi(u(t,x))] = 0 \\ u(0,x) = \phi(x) \end{cases} \quad \text{where} \quad (t,x) \in \mathbb{R}_+ \times \mathbb{R}. \tag{3}$$

The flux function ψ is assumed to only depend on the density function $u : \mathbb{R}_+ \times \mathbb{R} \to [0, u_{\max}]$ where $u_{\max}$ is the maximum density. The flux function can be characterized by $\psi : u \mapsto uv(u)$ where $v : [0, u_{\max}] \to [0, v_{\max}]$ is the velocity function which can be any decreasing function such that $v(0) = v_{\max}$ and $v(u_{\max}) = 0$. Let us first recall the solution of conservation laws.

Definition 1 *A function $u \in C([0,T]; L^1_{\text{loc}}(\mathbb{R}))$ (where L^1_{loc} is the space of locally integrable functions) is a **weak solution** of Equation (1) if for any test function $\varphi \in C^1_c((-\infty, T] \times \mathbb{R}, \mathbb{R})$,*

$$\int_0^T \int_{\mathbb{R}} \{u\partial_t\varphi + \psi(u)\partial_x\varphi\} + \int_{\mathbb{R}} u_\circ \varphi(0,\cdot) = 0. \tag{4}$$

*The **Kružkov entropy solution** (Kružkov, 1970) is the unique weak solution such that for all $k \in \mathbb{R}$ and all non-negative test function $\varphi \in C^1_c((-\infty, T] \times \mathbb{R}, \mathbb{R}_+)$, the entropy residual is non-negative:*

$$0 \le \mathcal{R}(u,\varphi,k) \overset{\textit{def}}{=} \int_0^T \int_{\mathbb{R}} |u-k|\partial_t\varphi + \textit{sign}(u-k)[\psi(u)-\psi(k)]\partial_x\varphi + \int_{\mathbb{R}} |u_\circ - k|\varphi(0,\cdot). \tag{5}$$

In particular, a key observation from Definition 1 is that weak solutions are not necessarily unique in the entropic sense. For instance, shock formation or discontinuous initial conditions can lead to multiple weak solutions that fail to satisfy the entropy condition.

Analytical Complexity of the Solution. Given an initial condition $\phi \in L^1 \cap \mathbf{BV}(\mathbb{R})$, the solution to Equation (1) exists in the sense of Definition 1, and in addition for any $t \in \mathbb{R}_+$, $u(t,\cdot) \in \mathbf{BV}(\mathbb{R}; [0, u_{\max}])$, the space of bounded variation functions over $\mathbb{R}$ (Kružkov (1970)). It should be noted that a function of total variation can have singular behavior, and derivatives are only defined in the sense of measures: $u \in \mathbf{BV}(\mathcal{U})$, if and only if there exists a finite Radon measure μ such that

$$\int_{\mathcal{U}} u\partial_x\varphi \,\mathrm{d}x = -\int_{\mathcal{U}} \varphi \,\mathrm{d}\mu;$$

i.e. the Radon measure μ defines the distributional derivative of function $u(t,\cdot)$. This explains that the derivatives in this space are difficult to handle and require careful approximation in computational schemes. This is crucial because many existing learning methods in the literature are not theoretically viable in this setting. For instance, PINNs do not provide a suitable framework for defining a loss for this class of functions.

2.1. Supervised Schemes

While the solutions of hyperbolic PDEs exhibit specific irregularities, there exist FV and FE numerical schemes that are provably strongly stable and convergent. By leveraging key ideas from these schemes, the following provides a theoretical foundation for analyzing and explaining the accuracy improvements of NN solvers over the corresponding numerical methods.

We consider a discretization $(\Delta t, \Delta x)$ of the domain $[0,T]\times[0,L]$ where $T, L > 0$. Let

$$u_i^n = \frac{1}{\Delta x}\int_{I_i} u(t_n, x)\,\mathrm{d}x \quad \text{and} \quad F_{i+1/2}^n = \int_{t_n}^{t_{n+1}} \psi(u(t, x_{i+1/2}))\,\mathrm{d}t$$

denote the average solution on cell $I_i = [x_{i-1/2}, x_{i+1/2}]$ at time $t_n = n\Delta t$, and the time-averaged flux across the cell interface at $x_{i+1/2}$ from t_n to t_{n+1}. Let $\hat{u}_i^n$ and $\hat{F}_{i+1/2}^n$ be the corresponding numerical approximations. In particular, the neural flux is given by $\mathcal{N}(\hat{u}_i^n, \hat{u}_{i+1}^n)$, where $\mathcal{N}$ is the NN. The approximate update rule, which becomes exact when using the true values, is given by

$$\forall i, n, \quad \hat{u}_i^{n+1} = \hat{u}_i^n - \Delta t \Delta x^{-1}(\hat{F}_{i+1/2}^n - \hat{F}_{i-1/2}^n) \quad \text{where} \quad \hat{F}_{i+1/2}^n \overset{\text{def}}{=} \mathcal{N}(\hat{u}_i^n, \hat{u}_{i+1}^n). \tag{6}$$

The main properties of the supervised NN-based solver (6) are as follows (Morand et al., 2024):

- Due to the update structure, the conservation of mass is preserved (hard constraint).
- The neural flux is unique up to a constant. One could force uniqueness of the flux function by considering $\hat{v}_i^n = \mathcal{N}(\hat{u}_i^n, \hat{u}_{i+1}^n)$ and $\hat{F}_{i+1/2}^n = \hat{u}_i^n \hat{v}_i^n$ or adding a soft constraint $\mathcal{N}(0,0) = 0$.
- The NN is trained on Riemann problems (initial conditions with two constant pieces, see Figure 1), ensuring it learns solutions that inherently satisfy the entropy condition.

The performance of the trained supervised model is detailed in Section 4; Table 1 shows it achieves higher accuracy than common FVMs from the literature. In particular, Figure 3 illustrates how, as the mesh refines, the NN consistently outperforms Godunov (Godunov, 1959), a classical FVM known to converge to the true solution of the PDE. More precisely, $\mathcal{E}_\mathcal{N} < \mathcal{E}_G$ for a constant ratio $\Delta t \Delta x^{-1}$, where $\mathcal{E}_\mathcal{N}$ and $\mathcal{E}_G$ are the approximation errors associated with the proposed model and the Godunov updates, respectively. Another notable observation is that the proposed model appears to converge at the same rate as Godunov, i.e. $\mathcal{E}_\mathcal{N} = \Theta(\mathcal{E}_G)$, for a fixed CFL condition.

2.2. Unsupervised Schemes

Consider an unsupervised learning approach that follows the FV-based structure (6), with the weak loss formulation (2). It can be computed as $\mathcal{L}(\theta) = \mathbb{E}_\varphi[\mathcal{L}_\varphi(\theta)^2]$, where

$$\mathcal{L}_\varphi(\theta) \overset{\text{def}}{=} \sum_{n=1}^{N}\sum_{i=1}^{I}\left(\hat{u}_i^n(\theta)\int_{x_{i-1/2}}^{x_{i+1/2}}[\varphi]_{t_n}^{t_{n+1}} + \psi(\hat{u}_i^n(\theta))\int_{t_n}^{t_{n+1}}[\varphi]_{x_{i-1/2}}^{x_{i+1/2}}\right) \tag{7}$$

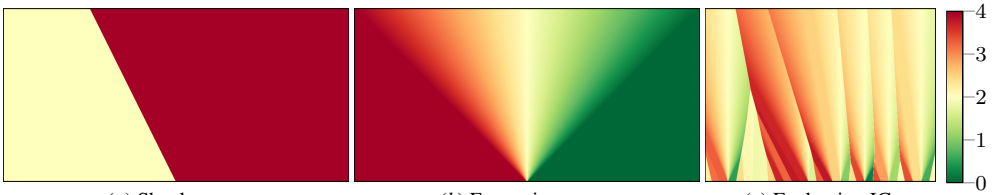
and the expectation is taken over uniformly distributed compactly supported polynomials.

Entropy Solution. From a theoretical point of view, conservation laws admit infinitely many weak solutions, among which there exists a unique entropy solution (it should be noted that the concept of entropy solution in multi-dimension requires more analytical rigor as the entropy solution might not be unique in this case). Naturally, to guide NN solvers toward the entropy solution, one may include a regularization term $\mathcal{L}_k(\theta) = \max_\varphi \mathcal{R}(\hat{u}(\theta), \varphi, k)$ in the loss, over Sobolev space $\varphi \in W_\circ^{1,\infty}([0,T]\times\mathcal{U})$, where $k \in \mathbb{R}$ and the entropy residual $\mathcal{R}$ is defined as in (5).

Let $\mathcal{S} = \{(t_i, x_i)\}_{i\in\{1,\cdots,M\}} \subset [0,T]\times\mathbb{R}$ be the spatiotemporal discretization. Define

$$\mathcal{E}_T(\theta_\mathcal{S}^*, \mathcal{S}, \varphi_\mathcal{S}^*, k_\mathcal{S}^*) \overset{\text{def}}{=} \min_\theta \max_\varphi \max_k (\mathcal{L}_\varphi(\theta)^2 + \mathcal{L}_k(\theta)) \quad \text{and} \quad \mathcal{E}(\theta) \overset{\text{def}}{=} \int_\mathbb{R} |\hat{u}_\theta(T,x) - u(T,x)|dx$$

as the minimum training error achievable, and the prediction error, respectively. The following error bound can then be calculated analytically.



(*a*) Shock wave. (*b*) Expansion wave. (*c*) Evaluation IC.

Figure 1: Exact solution for two Riemann problem and one piecewise-constant IC from the evaluation set.

Theorem 2 (De Ryck et al. (2024)) *With probability of at least* $1-\delta$*, we have that*

$$\mathcal{E}(\theta_{\mathcal{S}}^*) \lesssim \mathcal{E}_T(\theta_{\mathcal{S}}^*, \mathcal{S}, \varphi_{\mathcal{S}}^*, k_{\mathcal{S}}^*) + \boldsymbol{K}_1 \sqrt{\ln\left(\frac{\boldsymbol{K}_2}{\delta \varepsilon^3}\right)} + (1 + \|\hat{u}_{\theta^*}\|_{C^1}) \ln(\varepsilon^{-3})\varepsilon.$$

The constants ε*,* $\boldsymbol{K}_1$*, and* $\boldsymbol{K}_2$ *depend on* M *(number of samples),* L *(NN depth) and* W *(NN width).*

For training purposes, adding an entropy regularization term implies that one needs to deal with a max-min optimization problem, which is computationally expensive. On the other hand, when imposing the weak loss function (2) only, NN solvers appear to learn entropy solutions. From an analytical point of view, it is strongly desirable to understand why such a phenomena happens.

3. Analytical Insights into NN-Based Solvers

This section aims to provide analytical intuition for how NN-based solvers can improve upon traditional numerical schemes, based on empirical observations from trained models. Specifically, Figure 2 compares the neural flux $\mathcal{N}(\cdot,\cdot)$, defined as in (6), with the numerical flux $\mathcal{F}(\cdot,\cdot)$ at cell boundaries in two scenarios: varying u_R with fixed u_L, and vice versa, where u_R and u_L are the two inputs to both $\mathcal{N}$ and $\mathcal{F}$. We denote as u_c the critical density of the flow. The figure shows that the NN solver's flux corrects that of Godunov, leading to higher predictive accuracy. Let us formalize the discussion to some extent. Let us define

$$\mathcal{Y}_u \overset{\text{def}}{=} \{v \in [0, u_{\max}] : \partial_2 \mathcal{F}(u,v) \equiv 0\}, \quad \hat{\mathcal{Y}}_u \overset{\text{def}}{=} \{v \in [0, u_{\max}] : \partial_1 \mathcal{F}(v,u) \equiv 0\} \tag{8}$$

where the derivatives $v \mapsto \partial_2 \mathcal{F}(u,v)$ and $v \mapsto \partial_1 F(v,u)$ are defined almost everywhere. Given u, these sets consist of values of v for which the numerical flux is constant. Next, we define

$$\Gamma_1 \overset{\text{def}}{=} [0, u_c) \cap \Big\{ \bigcup_{r>0} \{u : u - r \in \mathcal{Y}_u\} \Big\}, \quad \Gamma_2 \overset{\text{def}}{=} [u_c, u_{\max}) \cap \Big\{ \bigcup_{r>0} \Big\{u : u + r \in \hat{\mathcal{Y}}_u\Big\} \Big\}. \tag{9}$$

The sets Γ_i, $i = 1, 2$ illustrate the values of $u < u_c$ (resp. $u \geq u_c$), some neighborhood of which belongs to $\mathcal{Y}_u$ (resp. $\hat{\mathcal{Y}}_u$). In particular, for $u \in \Gamma_1$, there exists a $\delta \in (0, r^*)$ with $r^* \overset{\text{def}}{=} \sup\{r : u - r \in \mathcal{Y}_u,\ u \in \Gamma_1\}$, where Γ_1 is defined as in (9), $\eta \in (0,1)$ and $\lambda > 0$, such that with probability of at least $1-\eta$ we have

$$|\mathcal{N}(u, u-\delta) - \psi(u)| \leq |\mathcal{F}(u, u-\delta) - \psi(u)| - \lambda. \tag{10}$$

This means the neural flux is more consistent with the value of the flux in some neighborhood of u. Similarly, for $u \in \Gamma_2$, there exists a $\delta \in (0, r^{**})$ with $r^{**} \overset{\text{def}}{=} \sup\{r : u + r \in \hat{\mathcal{Y}}_u,\ u \in \Gamma_2\}$, where Γ_2 is defined as in (9), $\eta \in (0,1)$ and $\lambda > 0$, such that with probability of at least $1-\eta$ we have

$$|\mathcal{N}(u+\delta, u) - \psi(u)| \leq |\mathcal{F}(u+\delta, u) - \psi(u)| - \lambda. \tag{11}$$

	1st order FVM					Higher order FVM		FEM
	GD	LF	EO	NN SV	NN USV	ENO	WENO	DG
L1	$4.5e^{-2}{}_{\pm 7e^{-3}}$	$3.5e^{-1}{}_{\pm 6e^{-2}}$	$4.6e^{-2}{}_{\pm 7e^{-3}}$	$\mathbf{2.9e^{-2}}_{\pm 5e^{-3}}$	$3.1e^{-2}{}_{\pm 5e^{-3}}$	$4.5e^{-2}{}_{\pm 2e^{-2}}$	$4.6e^{-2}{}_{\pm 2e^{-2}}$	$1.3e^{-2}{}_{\pm 2e^{-3}}$
L2	$7.3e^{-3}{}_{\pm 3e^{-3}}$	$2.2e^{-1}{}_{\pm 7e^{-2}}$	$7.4e^{-3}{}_{\pm 3e^{-3}}$	$\mathbf{3.7e^{-3}}_{\pm 1e^{-3}}$	$4.3e^{-3}{}_{\pm 2e^{-3}}$	$1.2e^{-2}{}_{\pm 6e^{-3}}$	$1.2e^{-2}{}_{\pm 7e^{-3}}$	$9.7e^{-4}{}_{\pm 5e^{-4}}$
Rel.	$4.8e^{-2}{}_{\pm 4e^{-2}}$	$4.0e^{-1}{}_{\pm 4e^{-1}}$	$4.9e^{-2}{}_{\pm 5e^{-2}}$	$\mathbf{3.4e^{-2}}_{\pm 4e^{-2}}$	$3.5e^{-2}{}_{\pm 3e^{-2}}$	$2.3e^{-2}{}_{\pm 7e^{-3}}$	$2.2e^{-2}{}_{\pm 7e^{-3}}$	$1.2e^{-2}{}_{\pm 7e^{-3}}$

Table 1: We compare the performance of the NN models (SV: supervised; USV: unsupervised) against common numerical schemes from the literature. Each method is evaluated on the entire evaluation set (500 ICs) described in Section 4. We report mean and standard deviation of L1 error ($\text{mean}(|u - \hat{u}|)$), L2 error ($\text{mean}((u - \hat{u})^2)$), and relative error ($\text{mean}(|u - \hat{u}|/|\max\{\varepsilon, u\}|)$).

Overall, Equations (10) and (11) aim to provide intuition for why the NN-based solver achieves higher accuracy compared to traditional numerical solvers.

4. Experiments and Discussion

The supervised and unsupervised models trained following the NN-based FVM formulation (Sections 2.1 and 2.2) are compared against the following numerical schemes: first-order FVMs (Lax-Friedrichs (LF) (Lax, 1954), Godunov (GD) (Godunov, 1959), and Engquist-Osher (EO) (Engquist and Osher, 1981)), higher-order FVMs (Essentially Non-Oscillatory (ENO) and Weighted ENO (WENO) (Shu, 1999)), and Discontinuous Galerkin (DG) (Hu and Shu, 1999), an FE method.

4.1. Experimental Setup

Datasets The supervised and unsupervised models are trained on a dataset of equally-spaced Riemann initial problems $\left\{(u_L, u_R) \in \{iN^{-1}u_{\max} \mid i = 0, \dots, N\}^2,\ u_L \neq u_R\right\}$ (i.e., simple shock and expansion waves; see Figure 1), computed on a coarse grid ($\Delta x = \Delta t = 5e^{-2}$). The evaluation set consists of 500 randomly-generated complex piecewise-constant ICs on a finer grid ($\Delta x = 5e^{-3}$ and $\Delta t = 5e^{-4}$) with 30 pieces each (see Figure 1), for $T = 1000$ timesteps. Their exact solutions are computed numerically using the Lax-Hopf algorithm (Simoni and Claudel, 2017).

Training The model is a 1D CNN with six convolution layers (kernel size 1, 16 channels), totaling ~ 1000 parameters, which is equivalent to sliding a fully-connected NN across cells. It is trained to predict T steps autoregressively, with T increased from 10 to 50 and the learning rate decreased from 10^{-4} to 10^{-5} over 50,000 epochs. Training is done with $\Delta x = \Delta t$ (CFL = 1) and takes ~ 30 minutes on an A5000 GPU. The NN surpasses Godunov's accuracy within minutes; the rest is fine-tuning. Full hyperparameters and dataset size were selected by search.

4.2. Performance Evaluation and Analysis of Deep Learning Schemes

We evaluate two first-order NN-based FVMs: a supervised model (Section 1.1) and an unsupervised one (Section 1.2). Table 1 shows that both models outperform standard first-order schemes and

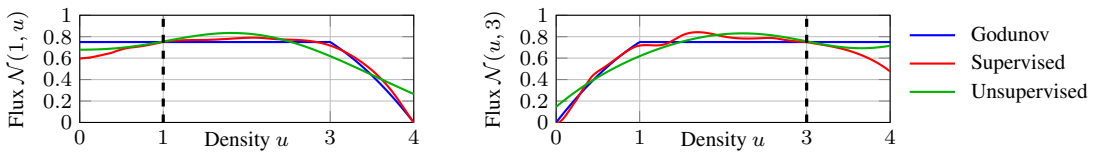


Figure 2: Comparison between the neural flux for the NN models and the numerical flux based on the Godunov scheme.

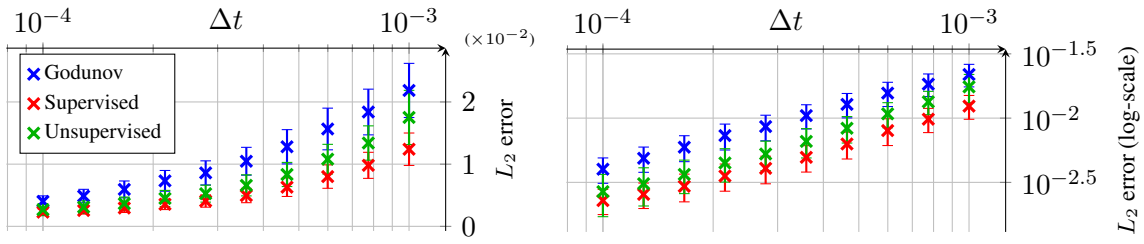


Figure 3: **Convergence plot for the supervised model.** The average L_2 error is computed against the exact solution on the evaluation set, with standard deviation shown as error bars. The ratio $\Delta t/\Delta x = 0.1$ remains constant as the mesh is refined.

even surpass higher-order ENO and WENO methods in L_1 and L_2 errors, with lower variance. This demonstrates the models' ability to generalize from training on coarse Riemann data to evaluation on fine-mesh complex initial problems. As expected, DG achieves the best overall accuracy and stability, however it is significantly more complex and computationally demanding than the simpler FV approaches and requires careful numerical implementation.

Figure 3 shows that both learned models (from Sections 2.1 and 2.2) achieve lower errors than Godunov across mesh sizes while matching its convergence rate, suggesting they approximate entropy solutions, since Godunov converges to the entropy solution as the discretization tends to zero. Figure 4 compares the final timestep of each method to the PDE solution, capturing maximum accumulated error over the whole autoregressive prediction. The learned models are generally more accurate, especially near shocks. In contrast, Godunov suffers from a known sonic glitch when the flux derivative vanishes (e.g., when the density equals 2) (Van Leer et al., 1989).

Finally, the learned numerical flux (6) is analyzed. One key property that a numerical flux $\hat{F}$ must have in most proofs of convergence is being non-decreasing in its first argument and non-decreasing in its second argument (Bertoluzza et al., 2009). Figure 2 illustrates that the Godunov numerical flux respects this condition. However, this is done in an artificial way that also verifies the consistency property: for all u, $\hat{F}(u, u) = \psi(u)$. Overall, the results suggest that NNs provide a competitive alternative to traditional schemes, particularly in accurately capturing discontinuities.

4.3. Applications of Supervised Learning to Traffic Modeling

PDE-based models rely on idealized conditions that often fail in real-world scenarios. NN-based models can be applied beyond PDE-based settings; in this case, we consider their application to highway traffic flow using vehicle trajectory data from a 14.5-minute drone video capturing a highway segment with no entrance or exit ramps (Wu et al., 2022), including approximately two minutes of free flow followed by four observable shock waves. Flow, density, and speed metrics are extracted from this data, with density values given for $t \in [0, 870s]$ and $x \in [0, 400m]$ (see Figure 6, left).

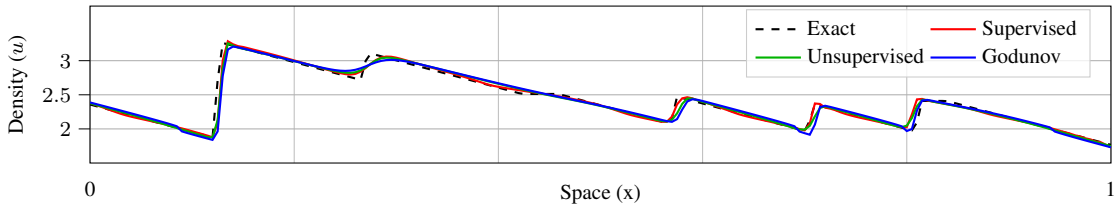


Figure 4: Example final state after $T = 1000$ timesteps for a piecewise-constant IC from the evaluation set.

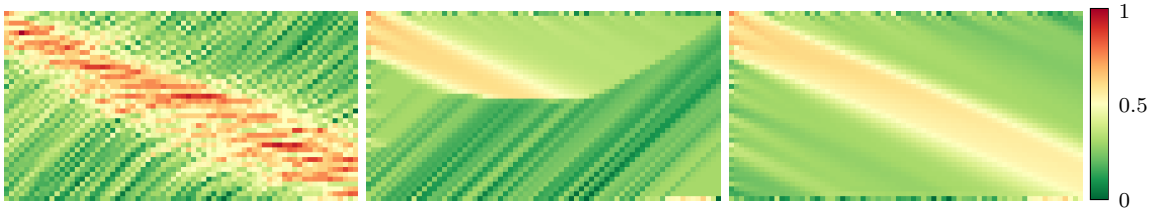


Figure 5: Traffic flow prediction on the training wave. **Left:** Ground Truth. **Mid:** Godunov prediction. **Right:** NN prediction.

Model Training and Evaluation We consider a supervised model (as described in Section 1.1) that was trained on about 8% of the data (the fourth wave, $t \in [800s, 870s]$) and validated on the remaining 92% ($t \in [0, 800s]$). Boundary conditions were imposed from the observed density data, as opposed to synthetic extrapolations, to ensure physical realism and a meaningful evaluation.

Results and Discussion To establish a baseline, widely-used flux functions (Triangular, Greenshield, Trapezoidal, Greenberg, and Underwood) are considered. They are fitted to the real data, then incorporated into the Godunov scheme to generate numerical predictions. It appears that a calibrated triangular flux performs best in approximating the observed traffic dynamics. Figure 5 compares the predictions of the trained NN model and those of Godunov on the training set. Figure 6 (right) then tests the model (trained on only 70 seconds, i.e., 8% of the data) on the entire dataset. While Godunov struggles to generalize due to its fixed flux function, the model adapts without requiring any flux tuning. However, it remains constrained by its 1^{st}-order formulation. Future work can incorporate higher-order inputs, account for time and space dependencies, or employ more complex memory-based or attention-driven NN architectures, to achieve higher accuracy.

5. Conclusion

This tutorial illustrated how NN-based approaches can enhance the approximation of PDE solutions, potentially surpassing classical numerical schemes. However, this comes with a fundamental challenge: NNs typically do not have the theoretical guarantees or convergence properties that have been proven for traditional methods. Several key directions thus remain open for future work. First, ensuring robustness and theoretical guarantees for NN-based approximations is crucial, particularly in preserving physical constraints and conservation laws. Second, extending these methods to more complex PDEs and high-dimensional systems will test their scalability. Third, integrating learning-based solvers with hybrid approaches, combining data-driven techniques with classical numerical methods, may improve accuracy and robustness.

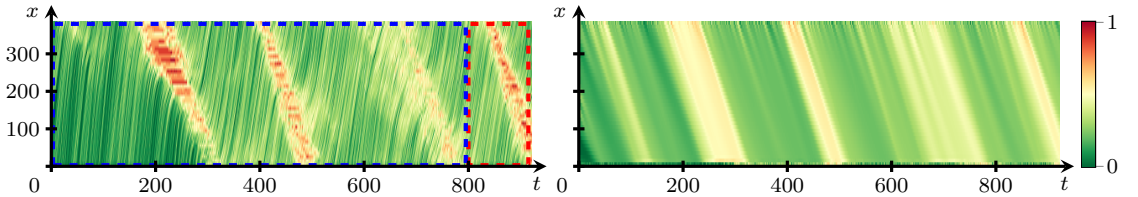


Figure 6: **Left:** Heatmap of the density training dataset where the red (resp. blue) box shows the subset of data used during training (resp. evaluation). **Right:** Autoregressive model prediction on the whole evaluation dataset.

References

Silvia Bertoluzza, Silvia Falletta, Giovanni Russo, and Chi-Wang Shu. *Numerical solutions of partial differential equations*, pages 89–95. Springer Science & Business Media, 2009. ISBN 978-3-7643-8940-6. doi: 10.1007/978-3-7643-8940-6_7. URL https://doi.org/10.1007/978-3-7643-8940-6_7.

Johannes Brandstetter, Daniel Worrall, and Max Welling. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Joan Bruna, Benjamin Peherstorfer, and Eric Vanden-Eijnden. Neural galerkin schemes with active learning for high-dimensional evolution equations. *Journal of Computational Physics*, 496: 112588, 2024.

Christian G. Claudel and Alexandre M. Bayen. Lax–Hopf Based Incorporation of Internal Boundary Conditions Into Hamilton-Jacobi Equation. Part II: Computational Methods. *IEEE Transactions on Automatic Control*, 55(5):1158–1174, 2010.

Bernardo Cockburn and Chi-Wang Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 1998.

Bernardo Cockburn, George E Karniadakis, and Chi-Wang Shu. *Discontinuous Galerkin methods: theory, computation and applications*, volume 11. Springer Science & Business Media, 2012.

Albert Cohen and Ronald DeVore. Approximation of high-dimensional parametric pdes. *Acta Numerica*, 24:1–159, 2015.

Tim De Ryck, Ameya D Jagtap, and Siddhartha Mishra. Error estimates for physics-informed neural networks approximating the navier–stokes equations. *IMA Journal of Numerical Analysis*, 44(1): 83–119, 2024.

Björn Engquist and Stanley Osher. One-sided difference approximations for nonlinear conservation laws. *Mathematics of Computation*, 36(154):321–351, 1981.

Lawrence C Evans. *Partial differential equations*, volume 19. American Mathematical Society, 2022.

Tiffany Fan, Kailai Xu, Jay Pathak, and Eric Darve. Solving inverse problems in steady-state navier-stokes equations using deep neural networks. *arXiv preprint arXiv:2008.13074*, 2020.

SK Godunov. A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. *Sbornik: Mathematics*, 47(8-9):357–393, 1959.

Yanan Guo, Xiaoqun Cao, Bainian Liu, and Mei Gao. Solving partial differential equations using deep learning and physical constraints. *Applied Sciences*, 10(17):5917, 2020.

Changqing Hu and Chi-Wang Shu. A discontinuous galerkin finite element method for hamilton–jacobi equations. *SIAM Journal on Scientific computing*, 21(2):666–690, 1999.

Stanislav N Kružkov. First order quasilinear equations in several independent variables. *Mathematics of the USSR-Sbornik*, 10(2):217, 1970.

Gitta Kutyniok, Philipp Petersen, Mones Raslan, and Reinhold Schneider. A theoretical analysis of deep neural networks and parametric pdes. *Constructive Approximation*, 55(1):73–125, 2022.

PD Lax. The initial value problem for nonlinear hyperbolic equations in two independent variables. *Ann. Math. Studies*, 33(21):1–229, 1954.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Randall J LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence*, 3(3):218–229, 2021.

Siddhartha Mishra and Roberto Molinaro. Estimates on the generalization error of physics-informed neural networks for approximating pdes. *IMA Journal of Numerical Analysis*, 43(1):1–43, 2023.

Victor Morand, Nils Müller, Ryan Weightman, Benedetto Piccoli, Alexander Keimer, and Alexandre M Bayen. Deep learning of first-order nonlinear hyperbolic conservation law solvers. *Journal of Computational Physics*, 511:113114, 2024.

Sinan Muftu. *Finite Element Method: Physics and Solution Methods*. Academic Press, 2022. ISBN 978-0128211274.

Yesom Park, Changhoon Song, and Myungjoo Kang. Beyond derivative pathology of pinns: Variable splitting strategy with convergence analysis, 2024. URL https://arxiv.org/abs/2409.20383.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

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.

Lars Ruthotto and Eldad Haber. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 62(3):352–364, 2020.

Yong Shang, Fei Wang, and Jingbo Sun. Randomized neural network with petrov–galerkin methods for solving linear and nonlinear partial differential equations. *Communications in Nonlinear Science and Numerical Simulation*, 127:107518, 2023. ISSN 1007-5704. doi: https://doi.org/10.1016/j.cnsns.2023.107518. URL https://www.sciencedirect.com/science/article/pii/S1007570423004392.

William Sharpless, Zeyuan Feng, Somil Bansal, and Sylvia Herbert. Linear supervision for nonlinear, high-dimensional neural control and differential games, 2025. URL https://arxiv.org/abs/2412.02033.

Rongye Shi, Zhaobin Mo, and Xuan Di. Physics-informed deep learning for traffic state estimation: A hybrid paradigm informed by second-order traffic models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 540–547, 2021.

Rongye Shi, Zhaobin Mo, Kuang Huang, Xuan Di, and Qiang Du. A physics-informed deep learning paradigm for traffic state and fundamental diagram estimation. *IEEE Transactions on Intelligent Transportation Systems*, 23(8):11688–11698, 2022. doi: 10.1109/TITS.2021.3106259.

Yeonjong Shin, Jerome Darbon, and George Em Karniadakis. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type pdes. *arXiv preprint arXiv:2004.01806*, 2020.

Chi-Wang Shu. High order eno and weno schemes for computational fluid dynamics. In *High-order methods for computational physics*, pages 439–582. Springer, 1999.

Michele D. Simoni and Christian G. Claudel. A fast simulation algorithm for multiple moving bottlenecks and applications in urban freight traffic management. *Transportation Research Part B: Methodological*, 104:238–255, 2017.

Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

Justin Sirignano, Jonathan F MacArt, and Jonathan B Freund. Dpm: A deep learning pde augmentation method with application to large-eddy simulation. *Journal of Computational Physics*, 423: 109811, 2020.

Bram Van Leer, Wen-Tzong Lee, and Kenneth G Powell. Sonic-point capturing. 1989.

Fangyu Wu, Dequan Wang, Minjune Hwang, Chenhui Hao, Jiawei Lu, Jiamu Zhang, Christopher Chou, Trevor Darrell, and Alexandre Byen. Decentralized vehicle coordination: The berkeley deepdrive drone dataset. *arXiv*, 2022.

Mingyuan Yang and John T. Foster. hp-variational physics-informed neural networks for nonlinear two-phase transport in porous media. *Journal of Machine Learning for Modeling and Computing*, 2(2):15–32, 2021. ISSN 2689-3967.

Yinhao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366:415–447, 2018.