

Article

Subgrid-Scale Parameterization in Burgers' Equation Using Structure-Preserving Neural Networks and Entropy Variables

Aijaz Nazir  and Ilya Timofeyev * 

Department of Mathematics, University of Houston, Houston, TX 77204, USA; anazir@cougarnet.uh.edu

* Correspondence: itimofey@cougarnet.uh.edu

Abstract

We present a machine learning approach for developing subgrid-scale (SGS) parameterizations in coarse simulations of partial differential equations. We utilize structure-preserving neural networks and entropy variables to learn subgrid fluxes in coarse simulations of the Burgers' equation. In particular, we employ a decoupled neural network architecture explicitly separating the subgrid corrections into two distinct components: a conservative Flux Potential network and an Eddy Viscosity network. We demonstrate that this reduced-order framework maintains high physical fidelity, accurately reproducing the energy spectrum, spatial and temporal correlation functions, and dynamical characteristics of the full-scale system. Furthermore, we show that our approach is robust and applicable to parameters outside the training regime.

Keywords: Burgers' equation; entropy variables; deep learning; sub-grid parameterization; structure-preserving neural networks

MSC: 65M99; 68T07

1. Introduction

Developing models for unresolved degrees of freedom in coarse-grained simulations of fluid models has been an active research topic for many decades. The literature on this subject is extensive; some examples from computational fluid dynamics (CFD) include Large Eddy Simulations (e.g., [1,2]), subgrid-scale viscosity models (e.g., [3,4]), the POD-ROM (Proper Orthogonal Decomposition Reduced-Order Models) approach (e.g., [5–7]), tensorial ROM for parameterized systems [8–11]. Various extensions of these techniques have also been proposed, which led to advances in computational methods for fluid dynamics. In addition, stochastic mode reduction [12] has been used to develop effective computational models for coarse variables for the 1D Burgers' [13] and the 1D shallow-water [14] equation. Recently, neural networks have been used to develop parameterizations of subgrid terms (e.g., [15–20]) with many examples demonstrating the validity and effectiveness of this approach. However, with a complete theory of neural networks still in the early stages of development, there is no guarantee of the mathematical properties of ML reduced models, such as stability, convergence, response to external perturbations, etc. Therefore, one of the most fruitful future research directions is to combine the advantages of rigorous mathematics and machine learning.

Hyperbolic systems have a rich mathematical structure that has been extensively explored in the literature. Analytical advancements for hyperbolic systems have led to a better understanding of nonlinear wave phenomena and advances in computational



Academic Editor: James M. Buick

Received: 28 May 2026

Revised: 16 July 2026

Accepted: 16 July 2026

Published: 19 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

methods. One important analytical concept in the theory of hyperbolic systems is the existence of the entropy function $\eta(u)$, where u is the vector of dependent variables in the system (see, e.g., [21]). This leads to the definition of entropy variables $v = \nabla_u \eta(u)$ and entropy fluxes that provide additional analytical structure and an alternative symmetric formulation in the space of entropy variables. This symmetric structure plays an important role in both the theoretical analysis of hyperbolic systems and the design of stable numerical methods. In particular, the entropy function and entropy variables provide a natural framework for constructing entropy-consistent and entropy-stable discretizations.

Recently, the entropy function and entropy variables have been used to develop neural network models to learn hyperbolic conservation laws from solution trajectories [22,23]. In this work, we use a similar approach where we learn the entropy function and construct a neural network model to learn subgrid fluxes in coarse simulations of hyperbolic equations. In contrast to [22,23], our focus is not on learning the underlying conservation law, but on adapting the methodology to learn the subgrid parameterization for unresolved physical processes in a coarse discretization of hyperbolic conservation laws. Moreover, we focus on long stationary simulations reproducing the energy cascade, the spatial and temporal correlation functions, and individual solutions of the hyperbolic system. Therefore, our work is related to CFD techniques mentioned at the beginning of this introduction, since our goal is to represent unresolved subgrid dynamics through effective neural network closures. The main contribution of this paper is the formulation and numerical validation of a structure-preserving neural-network closure framework for coarse simulations of hyperbolic conservation laws.

In this work, we utilize a structure-preserving neural network to model the subgrid processes in coarse simulations of the Burgers' equation. The main contribution of this paper is to introduce and systematically test a general structure-preserving framework for constructing data-driven subgrid parameterizations for coarse discretizations of hyperbolic conservation laws. The proposed framework combines machine learning with analytical properties of hyperbolic systems, particularly the entropy function, entropy variables, and flux-potential formulation. Although the present study focuses on the one-dimensional Burgers' equation, the objective is not merely to construct a model for this particular equation, but to establish a methodology that can subsequently be extended to more general hyperbolic systems and multidimensional fluid-dynamical problems.

In particular, we use three neural networks to model different components. First, we use an Input-Convex Neural Network (ICNN) [24] to learn the entropy function $\eta(u)$; entropy variables can then be obtained by automatic differentiation. Second, we use a feed-forward neural network to learn the flux potential. Third, we use a feed-forward neural network with feature variables to develop a Smagorinsky-type [3] Eddy Viscosity approximation. Details are presented in Section 4.2.

An important part of the contribution is the numerical evaluation of the complete framework rather than of its individual neural-network components in isolation. We test whether the resulting closure can simultaneously reproduce individual solution trajectories, long-time statistical properties, and the transfer of energy across resolved scales. We also assess the stability of the proposed coarse-grid discretization and compare its performance with standard numerical and subgrid-scale models.

We add large-scale forcing to perform long-time stationary simulations and show that our neural network reduced model accurately reproduces the energy spectra and the spatial and temporal correlation structure of the fully resolved simulations. We also demonstrate that the resulting coarse mesh discretization is stable, accurately reproduces individual solutions, and does not require any additional stabilization (e.g., flux limiters). Our results demonstrate that incorporating the mathematical properties of hyperbolic systems directly

into a machine learning approach yields an efficient, stable, and accurate subgrid modeling framework. Thus, the principal contribution of this work is both the formulation of the entropy-based neural closure framework and its validation across several complementary measures of coarse-model performance.

The rest of the paper is organized as follows. In Section 2, we introduce the Burgers' equation and numerical discretization. In Section 3, we discuss the problem formulation, including the definition of coarse variables, a brief overview of the entropy function and entropy variables, and a summary of the neural network modeling approach. Section 4 presents details about the data preparation, network architecture, and training. Numerical results are presented in Section 6. Conclusions are discussed in Section 7.

2. Problem Formulation

2.1. Burgers' Equation with Stochastic Forcing

The Burgers' equation [25] is one of the well-studied nonlinear partial differential equations, often used as a prototype model for turbulence. We consider the stochastically forced Burgers' equation over a periodic domain $x \in [0, L]$ with $L = 2\pi$ given by

$$\frac{\partial}{\partial t} u + \frac{\partial}{\partial x} f(u) = \rho(x, t) \quad (1)$$

with the nonlinear flux $f(u) = u^2/2$ and $u \equiv u(x, t)$. Here, ρ denotes a large-scale stochastic forcing term, defined as

$$\rho(x, t) = A \sum_{k \in K} [\alpha_k(t) \cos(kx) + \beta_k(t) \sin(kx)], \quad (2)$$

where A represents the forcing amplitude and L is the length of the spatial domain. The coefficients α_k and β_k are random variables evolving according to the AR(1) (autoregressive model of order 1) process

$$\alpha_k(t + \Delta t) = \psi \alpha_k(t) + \sigma \epsilon_{k,1}(t), \quad \beta_k(t + \Delta t) = \psi \beta_k(t) + \sigma \epsilon_{k,2}(t), \quad (3)$$

where $0 < \psi = 1 - \gamma \Delta t$ with $\gamma > 0$ and $\sigma > 0$ are AR(1) parameters, and $\epsilon_{k,i}(t)$, $i = 1, 2$ are i.i.d. Normal $N(0, \Delta t)$ random variables. In this paper, we use $\gamma = 1$ and $\sigma = 1.41$. The forcing magnitude is $A = 1$, unless otherwise specified. Coefficients α_k and β_k are time-correlated Normal $N(0, \sigma^2 / (1 - \psi^2))$ random variables. Equation (3) can be viewed as a temporal discretization of the Ornstein–Uhlenbeck process. In this paper, we use $K = \{1, 2, 3\}$, i.e., the first three wavenumbers are forced. The stochastic forcing $\rho(x, t)$ is introduced to counteract the numerical dissipation inherent in finite-volume schemes.

2.2. Space–Time Discretization

Next, we consider a finite-volume discretization of the PDE in (1). The spatial domain is discretized using a uniform finite-volume mesh with grid spacing $\Delta x = L/N_f$. The computational cells are defined as $C_i = [x_{i-1/2}, x_{i+1/2}]$, $i = 0, \dots, N_f - 1$, where the cell interfaces are located at $x_{i-1/2} = i\Delta x$ and $x_{i+1/2} = (i+1)\Delta x$. The cell centers are given by $x_i = (i+1/2)\Delta x$. This defines the fine-scale computational mesh $\mathcal{M}_f = \{x_i = i\Delta x \mid i = 0, 1, \dots, N_f - 1\}$. The semi-discrete system becomes

$$\frac{d}{dt} u_i = - \frac{f_{i+1/2} - f_{i-1/2}}{\Delta x} + \rho_i, \quad (4)$$

where $\Delta x = L/N_f$, u_i is the average over the cell C_i , and $f_{i+1/2}$ is a suitable discretization of the flux function. There are various options for the numerical flux $f_{i+1/2}$ with particular

details found in standard literature on finite volume and finite difference methods for conservation laws (e.g., [26,27]). In this paper, we adopt the Local Lax–Friedrichs flux (LLF) [28] given by

$$f_{i+1/2} = \frac{f(u_{i+1}) + f(u_i)}{2} - \frac{\lambda_{i+1/2}}{2}(u_{i+1} - u_i), \quad (5)$$

where $\lambda_{i+1/2}$ is the local bound for the maximum wave speed. In practice, $\lambda_{i+1/2}$ is often approximated using the Rusanov formula $\lambda_{i+1/2} = \max(|f'(u_i)|, |f'(u_{i+1})|) = \max(|u_i|, |u_{i+1}|)$.

For the temporal discretization, we employ Heun’s method, a second-order explicit strong stability preserving (SSP) [29] Runge–Kutta scheme, to advance the semi-discrete system in time. Let $\mathbf{u} = \{u_i(t), i = 0, \dots, N_f - 1\}$. Then, the semi-discrete system in (4) can be schematically written as $\dot{\mathbf{u}} = \mathbf{g}(\mathbf{u}) + \boldsymbol{\rho}(t)$. The Heun’s method can be written as

$$\tilde{\mathbf{u}}^{n+1} = \mathbf{u}^n + \mathbf{g}(\mathbf{u}^n)\Delta t, \quad \mathbf{u}^{n+1} = \mathbf{u}^n + \frac{1}{2}(\mathbf{g}(\mathbf{u}^n) + \mathbf{g}(\tilde{\mathbf{u}}^{n+1}))\Delta t + \boldsymbol{\rho}^n,$$

where $\mathbf{u}^n$ denotes the numerical solution at time t_n . We also include stochastic forcing $\boldsymbol{\rho}^n = \{\rho_i^n, i = 0, \dots, N_f - 1\}$ with $\rho_i^n \equiv \rho(x_i, t^n)$ into the second step.

3. Methodology

Fully resolved simulations are often prohibitively expensive, making it desirable to obtain a coarse description of the physical phenomena. To this end, it is necessary to develop an appropriate subgrid model that represents the interactions between the resolved and unresolved (subgrid) degrees of freedom. In this work, we formulate such a model by adapting an entropy-based, structure-preserving neural-network methodology to the problem of subgrid closure. The proposed framework combines three neural networks that learn the entropy function, the flux potential, and the Eddy Viscosity component, respectively, thus incorporating analytical structure directly into the subgrid parameterization.

3.1. Coarse Variables and Deviations

We introduce an averaging operator and define coarse variables as local averages over q fine grid points, i.e.,

$$U_I(t) = \frac{1}{q} \sum_{i=qI}^{q(I+1)-1} u_i(t) \quad (6)$$

for each cell I in a coarse mesh $\mathcal{M}_c = \{0, 1, 2, \dots, N_c - 1\}$ with $N_c = N_f/q$. The unresolved degrees of freedom are then defined as the residuals or deviations $y_i = u_i - U_{I(i)}$, where the index i for y_i and u_i refers to the coarse cell I where the fine cell i is located.

The dynamics of U_I can be obtained by averaging the fine-scale Equation (4):

$$\frac{d}{dt} U_I = -\frac{1}{q\Delta x} \sum_{i=qI}^{q(I+1)-1} (f_{i+1/2} - f_{i-1/2}) + \frac{1}{q} \sum_{i=qI}^{q(I+1)-1} \rho_i(t), \quad (7)$$

and using the telescoping property, the sum in the equation above can be rewritten as

$$\sum_{i=qI}^{q(I+1)-1} (f_{i+1/2} - f_{i-1/2}) = f_{q(I+1)-1/2} - f_{qI-1/2}.$$

Next, since our goal is to remain within the flux-difference framework, we define the “True” coarse fluxes as

$$F_{I+1/2}^T = f_{q(I+1)-1/2}, \quad F_{I-1/2}^T = f_{qI-1/2}, \quad (8)$$

where the superscript T stands for the “True” flux. In addition, we also define $\rho_I^U(t) \equiv \frac{1}{q} \sum_{i=qI}^{q(I+1)-1} \rho_i(t)$ as the average of the forcing term on the coarse mesh. The Equation (7) is exact, but not closed, because fluxes $f_{i\pm 1/2}$ depend on small-scale variables. Note that $f_{q(I+1)-1/2}$ represents the exact physical flux evaluated at the right boundary of the coarse cell I . Because it acts on the fine grid interface, it is a function of both the coarse variables and deviations. Substituting $u_i = y_i + U_{I(i)}$ into (5) and computing explicitly the coarse index $I(i)$, the “True” flux can be decomposed into a macroscopic component and a subgrid flux:

$$F_{I+1/2}^T = F_{I+1/2}(\mathbf{U}) + G_{I+1/2}(\mathbf{U}, \mathbf{y}), \quad (9)$$

where

$$F_{I+1/2} = \frac{F(U_I) + F(U_{I+1})}{2} = \frac{1}{4} (U_{I+1}^2 + U_I^2). \quad (10)$$

The subgrid flux $G_{I+1/2}(\mathbf{U}, \mathbf{y})$ depends on both the coarse variables and deviations. The goal of subgrid modeling is to develop a suitable approximation of the subgrid flux expressed only in terms of the coarse variables, thus eliminating explicit dependence on the deviations. In this paper, we use structure-preserving neural networks to approximate the subgrid flux $G_{I+1/2}$.

We assume that the forcing is slowly varying in space, and the averaged forcing $\rho_I^U(t)$ can be well approximated by the forcing evaluated at the mid-point of cell I , i.e., $\rho_I^U(t) \approx \rho(X_I, t)$. Thus, we denote $\rho_I(t) := \rho(X_I, t)$ and omit the subscript U for the rest of the paper.

3.2. Entropy Variables and Symmetrization

Data-driven subgrid models consistent with the underlying physical properties of the corresponding partial differential equations would have a better chance of reproducing the properties of fully resolved dynamics. Therefore, we design our neural network for estimating subgrid fluxes to be consistent with the theory of entropy-stable conservation laws (see, e.g., [21,30–32]), which we summarize briefly in this section.

Consider the conservation law

$$\frac{\partial}{\partial t} u + \frac{\partial}{\partial x} f(u) = 0. \quad (11)$$

Many conservation laws admit a strictly convex entropy function $\eta(u)$. The existence of such a function implies that the system is symmetrizable. Moreover, the strict convexity of η ensures a one-to-one mapping between the conservative variables u and the entropy variables, defined as

$$v = \eta'(u). \quad (12)$$

Since $\eta'(u)$ is strictly convex, then $v = \eta'(u)$ is invertible, and we can define $u(v)$. Transforming the governing equations into the dual space of entropy variables symmetrizes the system, since it defines a one-to-one change of variables (at least locally). The derivative $u'(v)$ is positive definite since the entropy function $\eta(u)$ is strictly convex. Next, we can write fluxes in terms of entropy variables $g(v) := f(u(v))$ and recast the conservation law in the symmetric form using entropy variables

$$u'(v) \frac{\partial v}{\partial t} + g'(v) \frac{\partial v}{\partial x} = 0.$$

We can also define the flux potential $\phi(u) = vf(u) - r(u)$ with $r'(u) = \eta'(u)f'(u)$. If we substitute the entropy variables into the flux potential, then we can rewrite the conservation law (11) as

$$\frac{\partial}{\partial t}u + \frac{\partial}{\partial x}\phi'(v) = 0.$$

3.3. Neural Network Subgrid Modeling

In this work, we propose a hybrid formulation for the discrete dynamics of the coarse variable U_I . Rather than utilizing a black-box neural network to approximate the right-hand side of the equation for coarse variables, we keep the flux formulation and decompose fluxes into three parts: a macroscopic component, a learned structurally conservative part, and a learned dissipative correction. We utilize the following function form for the reduced equations for coarse variables:

$$\frac{d}{dt}U_I = -\frac{1}{q\Delta x}(\mathcal{F}_{I+1/2} - \mathcal{F}_{I-1/2}) + \rho_I(t), \quad (13)$$

where the numerical flux at the interface $I + 1/2$ is constructed to preserve the underlying physical structure of the conservation law:

$$\mathcal{F}_{I+1/2} = F_{I+1/2} + F_{I+1/2}^{NN}(U_I, U_{I+1}) - \mathcal{D}_{I+1/2}^{NN}(U_I, U_{I+1}). \quad (14)$$

Here, $F_{I+1/2}$ is given by Equation (10) and the construction of $F_{I+1/2}^{NN}$ and $\mathcal{D}_{I+1/2}^{NN}$ is described below.

Neural Subgrid Flux $F_{I+1/2}^{NN}$ and Entropy Encoding. The structurally conservative part is estimated by learning two separate neural networks: the *Entropy Neural Network* η_θ and the *Flux Potential Neural Network* ϕ_θ . To ensure consistency, this term is calculated in the dual space of entropy variables $v = \eta'(U)$ [22,23]. In particular, the flux $F_{I+1/2}^{NN}$ is computed as

$$F_{I+1/2}^{NN} = \frac{1}{2}[\phi'_\theta(v_I) + \phi'_\theta(v_{I+1})], \quad v_I = \eta'_\theta(U_I),$$

where derivatives are computed using automated differentiation and θ denotes the network's parameters. To mathematically guarantee that $\eta_\theta(U)$ is a strictly convex entropy function, it is parameterized using an Input-Convex Neural Network (ICNN) [24]. Strict convexity (i.e., $\eta'' > 0$) is architecturally enforced by restricting the hidden-layer weights to the non-negative domain and employing convex, non-decreasing activation functions. Additionally, a quadratic skip-connection is incorporated to maintain a strictly positive lower bound on the second derivative. Particular details about the architecture of $\eta_\theta(u)$ and $\phi_\theta(v)$ are provided in Sections 4.2.1 and 4.2.2, respectively.

Dissipation $\mathcal{D}_{NN}$. The idea is similar to the Smagorinsky-type approximation [3]. Here, a neural network is used to learn the state-dependent viscosity coefficient. In particular, we use the following form for the dissipation term

$$\mathcal{D}_{I+1/2}^{NN} = C_\theta(\xi) \frac{\Lambda_{I+1/2}}{2} \times (U_{I+1} - U_I). \quad (15)$$

Here, $\Lambda_{I+1/2}$ denotes the local bound for the maximum wave speed, defined as $\Lambda_{I+1/2} = \max(|U_I|, |U_{I+1}|)$. The coefficient C_θ is predicted by an *Eddy Viscosity Neural Network*, which interrogates the local flow state through a feature vector ξ comprising the local average velocity and the amplified interfacial jump. We ensure that $C_\theta(\xi)$ is positive by applying the sigmoid function in the final layer. The sigmoid activation allows us to impose hard bounds on the viscosity coefficient, so that $C_\theta(\xi) \in [C_{\min}, C_{\max}]$.

Unlike traditional schemes where the dissipation coefficient is a fixed constant, C_θ is a state-dependent parameter learned through end-to-end backpropagation. This allows the model to perform *Residual Discovery*, where the neural branches ($F_{I+1/2}^{NN}$ and C_θ) collaborate

to resolve the high-frequency discrepancies between the coarse macroscopic discretization and the filtered high-fidelity Direct Numerical Simulation (DNS) reference. Details about the architecture of C_θ are provided in Section 4.2.3.

4. Data Preparation, Network Architecture, and Training

We develop a reduced model for the local averages of the stochastically forced Burgers' equation by introducing a structure-preserving hybrid "Gray-Box" framework. Rather than relying on standard feed-forward neural networks to approximate the entire nonlinear flux, we decompose the numerical flux into an analytical macroscopic component and learned structural and dissipative subgrid corrections. We note that standard mean-squared error formulations struggled to accurately capture shock dynamics without over-dissipating smooth waves. To address this limitation, we introduce a gradient-weighted loss function that emphasizes high-frequency interfacial jumps and train the model end-to-end using the total physical flux. Furthermore, the spurious oscillations that traditionally require flux limiters are natively suppressed through a state-dependent Eddy Viscosity Network, while consistency for nonlinear flux corrections is architecturally guaranteed via an Input-Convex Neural Network (ICNN).

The performance of the reduced model is assessed based on how well equilibrium statistical properties are recovered compared to those of the Direct Numerical Simulation (DNS). In particular, we illustrate that the reduced model is able to accurately reproduce the spectral energy decay and the spatial and temporal correlation functions of the large-scale variables. The response of the reduced model with respect to changes in the amplitude of the forcing is also analyzed.

In this section, we describe how the training dataset is generated, the structure of the three-branch neural architecture, and the latent learning procedure used to approximate the subgrid physics and resolve the dynamics on the coarse mesh.

4.1. Data Preparation

We conduct a high-resolution numerical simulation of the full model (4), sampling the solution at intervals of $100\Delta t$ with $\Delta t = 0.001$. Thus, each run generates 10000 snapshots. To ensure stability and remove transient effects, the initial 5000 model time units are discarded. The initial velocity field is defined as a superposition of the first two Fourier modes,

$$u(x, 0) = 0.1 \sin\left(\frac{2\pi x}{L} + \phi_1\right) + 0.1 \sin\left(\frac{4\pi x}{L} + \phi_2\right),$$

where $L = 2\pi$ and the phase shifts $\phi_1, \phi_2 \sim \mathcal{N}(0, 1)$ are independently sampled from a normal distribution. Time series of variables U_0 and U_1 , the corresponding "true" flux $F_{1/2}^T$, and the smoothness indicator $\beta_{1/2}$ (computed for coarse variables) are recorded. We generate three datasets for three different initial conditions, filter them as described below, and then combine them into one balanced dataset. The final dataset contains approximately 152 K samples. Details about training and validation samples are provided in Table 1.

Dataset Filtering. Smoothness indicators, introduced in WENO schemes [33], provide a quantitative measure of local solution regularity. For Burgers' dynamics, the indicator β identifies regions of nonlinear steepening and shock formation. Thus, it enables targeted selection of dynamically active zones for model training. In this work, local smoothness is quantified using a standard WENO-type smoothness indicator. For a discrete solution U_I , the indicator β_I is defined as

$$\beta_I = \frac{13}{12}(U_{I-1} - 2U_I + U_{I+1})^2 + \frac{1}{4}(U_{I-1} - U_{I+1})^2. \quad (16)$$

The first term approximates the local curvature of the solution through a second-order finite difference, while the second term measures the squared first-order gradient. In previous works (e.g., [34]), we determined that near-shock regions are most challenging for subgrid modeling. Therefore, to emphasize regions with significant subgrid activity, the dataset is filtered using the smoothness indicator β , which quantifies the local variability of the solution. A threshold of $\beta^* = 0.25$ is applied, separating the data into high- β (active subgrid regions) and low- β (smoother regions) subsets. Samples with $\beta \geq \beta^*$, which correspond to shocks, are retained in full. We remove 70% of the low- β samples from the dataset. This selective downsampling prevents the over-representation of low- β values, preserving diversity in the dataset while ensuring sufficient representation of higher- β values. This approach ensures that the training dataset effectively captures different behaviors, allowing the neural network to generalize well across different conditions.

4.2. Network Architecture and Training

We employ a structure-preserving hybrid neural architecture to learn the subgrid structural correction and dynamic dissipation as functions of the local state, which differs from the usual strategy of learning global solutions directly. The goal is to achieve a local, physics-informed approximation, where the total numerical flux at a specific interface $I + 1/2$ is determined only by the adjacent cell averages U_I and U_{I+1} . Thus, our flux representation relies on a compact two-point stencil; this is possible by using the entropy-consistent framework. Due to spatial homogeneity, it is sufficient to use data from a representative local interface (at $1/2$ in our case), allowing for a smaller dataset and efficient parallel training. We then utilize the same neural network to compute subgrid fluxes $\mathcal{F}_{I+1/2}$ for $I = 0, \dots, N_c - 1$ in simulations of the reduced dynamics (13).

The model architecture is implemented in JAX/Flax [35] and consists of three neural networks, as shown schematically in Figure 1. First, the *Entropy Neural Network* parameterizes the convex entropy function η using an Input-Convex Neural Network. It utilizes *Softplus* activations, non-negative weight constraints enforced via absolute value operations, and a quadratic skip-connection to guarantee a strictly positive-definite Hessian. Second, the *Flux Potential Neural Network* utilizes smoothly differentiable *Tanh* activations to map the dual entropy variables to the flux potential. Finally, the *Eddy Viscosity Neural Network* employs *Swish* activation [36] and processes an augmented feature vector containing the local average velocity and an amplified interfacial jump, i.e.,

$$\xi = \left[\frac{U_I + U_{I+1}}{2}, 10|U_I - U_{I+1}| \right].$$

Its final layer applies a scaled *Sigmoid* function to restrict the dissipation coefficient $C_\theta(\xi)$ strictly within $[C_{\min}, C_{\max}] = [0.35, 2.0]$. Analysis of the neural network numerical scheme in Section 5 provides analytical guidelines for selecting constants $C_{\min}$ and $C_{\max}$.

To train this hybrid architecture, the model is optimized directly on the high-fidelity DNS dataset. Rather than relying on standard mean squared error, which frequently struggles to balance shock capturing with smooth wave preservation, we introduce a *Gradient-Weighted Loss* function

$$\mathcal{L} = \sum_{\text{batch}} w_{I+1/2} \|F_{I+1/2}^T - \mathcal{F}_{I+1/2}\|_2^2 \quad \text{with} \quad w_{I+1/2} = 1.0 + \gamma |U_I - U_{I+1}|,$$

where the summation is taken over a batch of samples. The loss evaluates the Mean Squared Error (MSE) between the assembled neural flux prediction and the “True” coarse flux computed from fine-mesh simulations. To aggressively target high-frequency interfacial jumps without over-dissipating smooth regions, the per-sample squared error is scaled by a

weight factor $w_{I+1/2}$, where γ serves as a tunable hyperparameter controlling the sensitivity to local gradients; the value $\gamma = 0.2$ works well in this case. While offline validation metrics naturally favor $\gamma \rightarrow 0$ due to the reduced penalty magnitude, our a posteriori analysis demonstrates that under-regularizing the network removes critical physical constraints, causing under-damped high-frequency errors during dynamic integration. Conversely, excessive weighting ($\gamma \geq 0.4$) introduces severe optimization stiffness. Thus, $\gamma = 0.2$ provides the minimum effective regularization necessary for long-term shock stability.

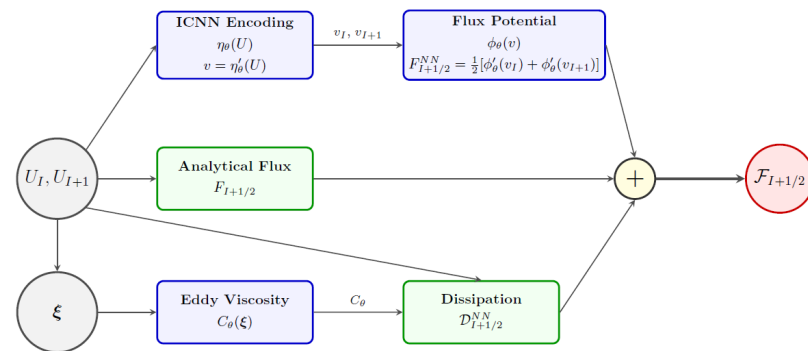


Figure 1. Schematic representation of the three branches for the structure-preserving “Gray-Box” computational pipeline. The total numerical flux $\mathcal{F}_{I+1/2}$ is the sum of three fluxes and is given by Equation (14). The flux $F_{I+1/2}$ (middle) is evaluated using Formula (10). Structural corrections (top) are restricted by learning a convex network $\eta_\theta(U)$ and the flux potential $\phi_\theta(v)$. State-dependent dissipation (bottom) is implemented by learning the viscosity coefficient $C_\theta(\xi)$ and multiplying it by the LLF-type term (Equation (15)). Blue boxes are modeled using different neural networks. Green boxes are given by explicit formulas.

Network parameters are optimized via latent-space learning using the Adam optimizer. We implement a cosine decay learning-rate schedule with an initial rate of 10^{-3} , allowing rapid initial convergence followed by fine-tuning of the network weights. The dynamic cosine decay schedule decreases the learning rate smoothly from an initial value to a prescribed minimum value according to a cosine curve, providing a gradual transition from aggressive parameter updates to smaller corrective updates during training. The model is trained for 200 epochs using mini-batches of size 128, iteratively minimizing the gradient-weighted loss on the training set. Validation is carried out by performing a long simulation of the reduced equation on the coarse grid and comparing spectra and individual solutions. In the next sections, we use a generic notation $z^{(\cdot)}$ and $\sigma(\cdot)$ to denote hidden layers and activation functions; these design features are different in each section. A comprehensive summary of the physical parameters, architectural capacity, and training configurations for the proposed model is provided in Table 1.

A sensitivity analysis on hidden neurons ($N \in \{16, 32, 64\}$) for the Eddy Viscosity NN revealed a flat plateau in validation error. This indicates that the underlying physical manifold of the subgrid flux is fully represented by narrower architectures. To maximize computational acceleration relative to the DNS baseline, we selected the parsimonious 16-neuron architecture, strictly minimizing matrix-multiplication overhead during the PDE time-stepping loop.

Table 1. Summary of physical and neural network hyperparameters for the proposed architecture.

Category	Parameter	Value
Physical Setup	Fine Grid Resolution (N_f)	512
	Coarse Grid Resolution (N_c)	64
	Domain Length (L)	2π
	Time Step (Δt)	0.001
Dataset Specifications	Training Samples (80%)	121,832
	Validation Samples (20%)	30,459
Entropy NN (η)	Activations	Softplus
	2 Hidden Layers	16, 16
Flux Potential NN (ϕ)	Activations	Tanh
	1 Hidden Layer	16
Eddy Viscosity NN (ν)	Activations	Swish, Sigmoid
	1 Hidden Layer	16
	Dynamic Viscosity Bounds	[0.35, 2.0]
Training & Optimization	Optimizer	Adam
	Learning Rate Schedule	Cosine Decay
	Initial Learning Rate	1×10^{-3}
	Gradient Regularization Weight (γ)	0.2
	Batch Size	128
	Epochs	200

4.2.1. Input Convex Neural Network (ICNN) for Entropy Formulation

To ensure that the learned discrete conservation law remains hyperbolic, the parameterized entropy function $\eta_\theta(u)$ must be strictly convex with respect to the state variable u . We achieve this by structuring the entropy network as an Input Convex Neural Network (ICNN) [24]. ICNNs enforce overall convexity by satisfying two structural constraints:

1. The nonlinear activation functions must be both convex and non-decreasing.
2. All weight matrices connecting the hidden layers must be strictly non-negative.

Since we consider a scalar equation, the input is $u \in \mathbb{R}^1$. We satisfy the first constraint by employing the smooth softplus activation function $\sigma(x) = \ln(1 + \exp(x))$. In the first hidden layer, the input undergoes a standard affine transformation followed by the activation:

$$z^{(1)} = \sigma(W_u^{(1)}u + b^{(1)}),$$

where $W_u^{(1)}$ and $b^{(1)}$ are the standard weights and biases.

For the second hidden layer, the network utilizes a skip-connection, directly re-injecting the input u alongside the features from the previous layer. To satisfy the second convexity constraint, the hidden-to-hidden weight matrix $W_z^{(1)}$ is restricted to the non-negative orthant. In our implementation, this is achieved by taking the element-wise absolute value of the trainable parameters, defining the effective weights as $\tilde{W}_z^{(1)} = |W_z^{(1)}|$. Thus, the second hidden layer evaluates as

$$z^{(2)} = \sigma(z^{(1)}\tilde{W}_z^{(1)} + W_u^{(2)}u + b^{(2)}).$$

The dimension of each hidden layer is 16.

To maximize the expressive power of the ICNN, the final output layer is augmented with both linear and quadratic transformations of the original input u . The quadratic weight coefficient, $\tilde{W}_S$, is also constrained to be non-negative via an absolute value projection ($\tilde{W}_S = |W_S|$). The final convex entropy function is computed as

$$\eta(u) = z^{(2)}\tilde{W}_z^{(2)} + b^{(3)} + W_u^{(3)}u + \tilde{W}_S u^2.$$

Since $\tilde{W}_S \geq 0$ and u^2 is a convex function, this quadratic augmentation strictly preserves the overall convexity of the network. This absolute-value bounding strategy provides a robust guarantee of hyperbolicity without the need for complex regularization penalties during training. Note that the standard entropy function for the Burgers' equation is quadratic, $\eta(u) = \frac{1}{2}u^2$. Since the goal of our framework is to learn an entropy function appropriate for the coarse discretization, the learned entropy function needs to not coincide exactly with the standard entropy of the Burgers' equation. However, including the quadratic term in the output layer provides the network with the exact Burgers' entropy as a baseline, allowing it to learn only the additional corrections required by the coarse model. Therefore, this augmentation, most likely, improves the efficiency of training the entropy network. For other hyperbolic systems, it may be advantageous to augment the output layer with a known convex entropy function of the underlying equations.

4.2.2. Flux Potential Network

While the entropy network $\eta_\theta(u)$ guarantees the stability of the system, the dispersive wave-speed corrections are governed by the conservative subgrid flux, defined as the gradient of a scalar potential: $\phi'_\theta(v)$.

To parameterize this potential function ϕ_θ , we use a standard Fully Connected Neural Network (FCNN) mapping the entropy variables v to a scalar output. For this specific implementation, the architecture is kept deliberately lightweight. The input v is passed through a single hidden layer of dimension 16 paired with a hyperbolic tangent tanh activation function. The hidden state is computed as

$$z^{(1)} = \tanh\left(W_v^{(1)}v + b^{(1)}\right).$$

This is followed by a linear output layer to predict the scalar potential:

$$\phi_\theta(v) = W_v^{(2)}z^{(1)} + b^{(2)}.$$

Taking the exact derivative of this compact network with respect to its inputs provides the conservative flux correction in the reduced equation.

4.2.3. Eddy Viscosity Network

To suppress spurious oscillations without over-damping the physical energy cascade, we introduce an independent Eddy Viscosity network, C_θ . Rather than replacing the baseline numerical dissipation (such as the Local Lax–Friedrichs scheme), this network outputs a dynamic scaling coefficient that modulates the baseline viscosity based on localized feature inputs ξ (e.g., cell averages, jumps, or curvature).

The network processes these input features through a single hidden layer of 16 neurons utilizing the swish activation function [36], which ensures smooth, non-vanishing gradient flow during training:

$$z^{(1)} = \text{swish}\left(W_f^{(1)}\xi + b_f^{(1)}\right).$$

A critical design element of this network is the strict physical bounding of its output. To prevent artificial energy pileup at the smallest grid scales due to insufficient dissipation, the final layer utilizes a sigmoid activation function $\sigma(x)$ which is explicitly scaled and shifted:

$$C_\theta = 1.65 \sigma \left(W_f^{(2)} z^{(1)} + b_f^{(2)} \right) + 0.35.$$

Since the sigmoid function is strictly bounded by $(0, 1)$, this formulation guarantees that the network output operates exclusively within the range $C_\theta \in [0.35, 2.00]$. This architecture ensures that at least 35% of the baseline numerical viscosity is always maintained to suppress spurious high-frequency oscillations.

5. Analytical Properties of the Reduced Model

In this section, we describe several analytical properties of the hybrid scheme with flux (14). In particular, we derive conditions under which the numerical scheme is total variation diminishing and discuss entropy stability.

The total variation diminishing (TVD) property is a key stability criterion for numerical schemes for hyperbolic conservation laws. A TVD scheme guarantees that the total variation of the numerical solution does not increase in time, thereby suppressing the growth of spurious oscillations near discontinuities.

Definition 1 (Discrete total variation). Let $\{u_j\}$, $j = 0, \dots, J$ be a discrete sequence. Its total variation is defined by

$$TV(u) = \sum_{j=0}^{J-1} |u_{j+1} - u_j|.$$

The following definition introduces the TVD property of a discrete numerical solution $u^n = \{u_0^n, \dots, u_J^n\}$, where n denotes the time-step index.

Definition 2 (Total variation diminishing scheme). Let u_j^n denote the numerical approximation at time-step n , i.e., $u_j^n \approx u(n\Delta t, j\Delta x)$. A numerical scheme is called total variation diminishing, or TVD, if the total variation of the numerical solution does not increase with time, i.e.,

$$TV(u^{n+1}) \leq TV(u^n)$$

for every n .

First, we need to make an assumption about the regularity of the Neural Subgrid Flux.

Assumption 1 (Lipschitz regularity of the Neural Subgrid Flux). Consider the Neural Subgrid Flux $F_{I+1/2}^{\text{NN}} = \mathcal{N}(U_I, U_{I+1})$, where $\mathcal{N} : \mathbb{R}^2 \rightarrow \mathbb{R}$ is a continuous function. Assume that for each admissible interval $\mathcal{B} = [u_{\min}, u_{\max}]$, $\mathcal{N}(a, b)$ is Lipschitz in each argument separately; that is, there is a constant L_{nn} (possibly $L_{\text{nn}} = L_{\text{nn}}(\mathcal{B})$) such that

$$|\mathcal{N}(a, b) - \mathcal{N}(c, b)| \leq L_{\text{nn}} |a - c|, \quad |\mathcal{N}(a, b) - \mathcal{N}(a, c)| \leq L_{\text{nn}} |b - c| \quad (17)$$

for all $a, b, c \in \mathcal{B}$.

Next, we derive TVD conditions for the effective NN scheme and discuss consequences for designing the neural network. The main idea is that the neural corrective flux may introduce additional variation into the numerical solution, while the Eddy Viscosity term acts to damp this variation. The TVD conditions derived below quantify the amount of

numerical dissipation required to compensate for the corrective flux and guarantee that the total variation does not increase from one time step to the next.

Proposition 1 (Sufficient TVD conditions for the neural network-augmented scheme). *Let the spatial domain be discretized with uniform mesh spacing Δx and time step Δt , and define $r = \Delta t / \Delta x$. Consider the conservative update*

$$U_I^{n+1} = U_I^n - r(\mathcal{F}_{I+1/2} - \mathcal{F}_{I-1/2}), \quad (18)$$

where the composite numerical flux at the interface $I + 1/2$ is

$$\mathcal{F}_{I+1/2} = F_{I+1/2} + F_{I+1/2}^{\text{NN}}(U_I, U_{I+1}) - \mathcal{D}_{I+1/2}^{\text{NN}}. \quad (19)$$

Here,

$$F_{I+1/2} = \frac{1}{2}(f(U_I) + f(U_{I+1})), \quad F_{I+1/2}^{\text{NN}} = \mathcal{N}(U_I, U_{I+1}),$$

and the dynamic Eddy Viscosity flux is

$$\mathcal{D}_{I+1/2}^{\text{NN}} = C_{\theta, I+1/2} \frac{\alpha_{I+1/2}}{2} (U_{I+1} - U_I), \quad (20)$$

where $C_{\theta, I+1/2} = C_{\theta}(\xi_{I+1/2})$ is evaluated from the known numerical solution at time level n and treated as an interface coefficient during the update. Assume that

$$0 < C_{\min} \leq C_{\theta, I+1/2} \leq C_{\max} < \infty.$$

Let

$$\Delta U_{I+1/2} = U_{I+1} - U_I$$

and let $\Lambda_{I+1/2} = \max\{|f'(U_I)|, |f'(U_{I+1})|\}$ be a local wave-speed bound. To avoid degeneracy of the diffusion (20) when $\Lambda_{I+1/2} = 0$, assume that the viscosity uses a positive speed floor, i.e.,

$$\alpha_{I+1/2} = \max(\Lambda_{I+1/2}, \varepsilon), \quad \varepsilon > 0.$$

Assume also that the neural corrective flux satisfies Assumption 1 on the admissible range, with Lipschitz constant L_{nn} . The Lipschitz constant L_{nn} measures the sensitivity of the neural corrective flux to perturbations in its inputs. Consequently, larger values of L_{nn} require stronger numerical dissipation in order to preserve the TVD property.

If, for every interface $I + 1/2$,

$$C_{\theta, I+1/2} \alpha_{I+1/2} \geq \Lambda_{I+1/2} + 2L_{\text{nn}}, \quad (21)$$

and the time step satisfies

$$r(C_{\theta, I+1/2} \alpha_{I+1/2} + 2L_{\text{nn}}) \leq 1 \quad \text{for every interface } I + 1/2, \quad (22)$$

then the scheme (18)–(19) satisfies Harten's TVD conditions.

The first condition in (21) requires that the adaptive Eddy Viscosity be large enough to dominate both the physical wave propagation and the variation introduced by the neural corrective flux. The second condition (22) is a CFL-type restriction that ensures the coefficients in Harten's incremental representation satisfy the bounds required for the TVD property.

The proof proceeds by rewriting the scheme in Harten's incremental form. The neural corrective flux is first expressed in terms of neighboring solution differences using its

Lipschitz continuity, after which the resulting coefficients are compared with Harten's sufficient conditions for TVD schemes.

Proof. The proof consists of expressing each contribution to the numerical flux in terms of neighboring solution differences so that the update can be written in Harten's incremental form. Once this representation is obtained, the TVD property follows by verifying Harten's coefficient conditions.

Define

$$v_{I+1/2} = C_{\theta, I+1/2} \alpha_{I+1/2}.$$

Then,

$$\mathcal{D}_{I+1/2}^{\text{NN}} = \frac{v_{I+1/2}}{2} \Delta U_{I+1/2}.$$

We now express each component of the numerical flux in terms of the neighboring solution differences $\Delta U_{I\pm 1/2}$. This allows the update to be written in Harten's incremental form.

First, for the macroscopic central flux,

$$F_{I+1/2} - F_{I-1/2} = \frac{1}{2} a_{I+1/2} \Delta U_{I+1/2} + \frac{1}{2} a_{I-1/2} \Delta U_{I-1/2}, \quad (23)$$

where $|a_{I+1/2}| \leq \Lambda_{I+1/2}$ from the definition of $\Lambda_{I+1/2}$.

Next, the neural corrective flux is treated similarly. Since it depends on two neighboring states, we use the Lipschitz regularity assumption to express its difference in terms of the same neighboring increments. We write

$$F_{I+1/2}^{\text{NN}} - F_{I-1/2}^{\text{NN}} = \mathcal{N}(U_I, U_{I+1}) - \mathcal{N}(U_{I-1}, U_I).$$

Adding and subtracting $\mathcal{N}(U_I, U_I)$ gives

$$F_{I+1/2}^{\text{NN}} - F_{I-1/2}^{\text{NN}} = [\mathcal{N}(U_I, U_{I+1}) - \mathcal{N}(U_I, U_I)] + [\mathcal{N}(U_I, U_I) - \mathcal{N}(U_{I-1}, U_I)].$$

Thus, there exist coefficients $q_{I+1/2}$ and $p_{I-1/2}$ such that

$$F_{I+1/2}^{\text{NN}} - F_{I-1/2}^{\text{NN}} = q_{I+1/2} \Delta U_{I+1/2} + p_{I-1/2} \Delta U_{I-1/2}, \quad (24)$$

with

$$q_{I+1/2} = \frac{\mathcal{N}(U_I, U_{I+1}) - \mathcal{N}(U_I, U_I)}{\Delta U_{I+1/2}}, \quad p_{I-1/2} = \frac{\mathcal{N}(U_I, U_I) - \mathcal{N}(U_{I-1}, U_I)}{\Delta U_{I-1/2}}, \quad (25)$$

so that

$$|q_{I+1/2}| \leq L_{\text{nn}}, \quad |p_{I-1/2}| \leq L_{\text{nn}} \quad (26)$$

using Assumption 1. The coefficients $q_{I+1/2}$ and $p_{I-1/2}$ are difference quotients of $\mathcal{N}$ with respect to its second and first arguments, respectively. When the denominator is zero, the associated numerator is also zero and the coefficient may be set equal to zero.

The Eddy Viscosity contribution satisfies

$$\mathcal{D}_{I+1/2}^{\text{NN}} - \mathcal{D}_{I-1/2}^{\text{NN}} = \frac{v_{I+1/2}}{2} \Delta U_{I+1/2} - \frac{v_{I-1/2}}{2} \Delta U_{I-1/2}. \quad (27)$$

Substituting (23), (24), and (27) into (18), we obtain

$$U_I^{n+1} = U_I^n + \Phi_{I+1/2}^+ \Delta U_{I+1/2} - \Phi_{I-1/2}^- \Delta U_{I-1/2}. \quad (28)$$

Equation (28) has the standard incremental form considered by Harten. The coefficients $\Phi^\pm$ determine whether the update is TVD. The coefficients are defined as

$$\Phi_{I+1/2}^+ = \frac{r}{2} (v_{I+1/2} - a_{I+1/2} - 2q_{I+1/2}), \quad (29)$$

$$\Phi_{I+1/2}^- = \frac{r}{2} (v_{I+1/2} + a_{I+1/2} + 2p_{I+1/2}). \quad (30)$$

By Harten's theorem, the incremental scheme (28) is TVD provided that, for every interface,

$$\Phi_{I+1/2}^+ \geq 0, \quad \Phi_{I+1/2}^- \geq 0, \quad \Phi_{I+1/2}^+ + \Phi_{I+1/2}^- \leq 1. \quad (31)$$

It therefore remains to verify that the coefficients $\Phi^\pm$ satisfy the three inequalities in (31). We estimate each coefficient using the wave-speed bound and the Lipschitz continuity of the neural corrective flux. Using $|a_{I+1/2}| \leq \Lambda_{I+1/2}$ and $|q_{I+1/2}| \leq L_{nn}$, condition (21) gives

$$v_{I+1/2} - a_{I+1/2} - 2q_{I+1/2} \geq v_{I+1/2} - \Lambda_{I+1/2} - 2L_{nn} \geq 0.$$

Therefore, $\Phi_{I+1/2}^+ \geq 0$. Similarly, using $|p_{I+1/2}| \leq L_{nn}$,

$$v_{I+1/2} + a_{I+1/2} + 2p_{I+1/2} \geq v_{I+1/2} - \Lambda_{I+1/2} - 2L_{nn} \geq 0,$$

and hence $\Phi_{I+1/2}^- \geq 0$.

The final TVD condition concerns the sum of the incremental coefficients. Adding (29) and (30), the macroscopic advective terms cancel:

$$\Phi_{I+1/2}^+ + \Phi_{I+1/2}^- = r(v_{I+1/2} + p_{I+1/2} - q_{I+1/2}).$$

Using (26), we obtain

$$\Phi_{I+1/2}^+ + \Phi_{I+1/2}^- \leq r(v_{I+1/2} + 2L_{nn}).$$

Therefore, condition (22) implies

$$\Phi_{I+1/2}^+ + \Phi_{I+1/2}^- \leq 1.$$

All three Harten conditions (31) hold, and the scheme is TVD. $\square$

When $\Lambda_{I+1/2} > \varepsilon$, condition (21) can be rewritten as

$$C_{\theta, I+1/2} \geq 1 + \frac{2L_{nn}}{\Lambda_{I+1/2}}.$$

Inequalities (21) and (22) have direct consequences for the architecture of the network. Recall that we use a rescaled sigmoid function in the final layer of the dissipation network to ensure that $C_{\min} \leq C_{\theta, I+1/2} \leq C_{\max}$. Thus, the inequality above indicates that it is important to select $C_{\max} > 1$ to make the TVD property feasible. At the same time, conditions (21) and (22) are sufficient, not necessary, and hence they should be interpreted as conservative analytical guidelines. The role of the Eddy Viscosity term is primarily to stabilize the scheme near shocks and steep gradients; in smooth regions, excessive artificial diffusion may reduce accuracy. In our numerical experiments, we found that imposing $C_{\min} = 1$ makes the scheme too diffusive. Therefore, we choose $C_{\min} < 1$ and allow the neural network to increase $C_{\theta, I+1/2}$ adaptively in regions where additional viscosity is needed. The lower bound $C_{\min}$ is also controlled by the next proposition. Condition (22) can also be checked a posteriori during the numerical simulation to verify that the time step is sufficiently small.

We note that strict TVD enforcement is not a necessary condition for a high-resolution scheme to perform well in practice. High-order methods such as WENO [37] and Runge–Kutta Discontinuous Galerkin [38] are not strictly TVD [39,40], which is consistent with the theoretical barrier established by Godunov’s theorem [41]. The conditions derived in Proposition 1 are sufficient but not necessary: the scheme may satisfy the TVD criterion locally depending on the network output C_θ , but is not required to do so globally in order to remain stable and accurate.

The TVD property controls the growth of oscillations in the numerical solution, whereas entropy stability guarantees consistency with the physical entropy inequality satisfied by weak solutions. We next derive a sufficient condition under which the neural corrective flux and the adaptive Eddy Viscosity term together produce a discretely entropy-stable scheme.

Proposition 2 (Discrete entropy stability of the neural network-augmented scheme). *Consider the semi-discrete conservative scheme for the Burgers’ equation*

$$u_t + \left(\frac{1}{2} u^2 \right)_x = 0$$

with entropy pair

$$\eta(u) = \frac{1}{2} u^2, \quad q(u) = \frac{1}{3} u^3,$$

entropy variable $v = \eta'(u) = u$, and entropy potential

$$\psi(u) = v f(u) - q(u) = \frac{1}{6} u^3.$$

The semi-discrete scheme is

$$\frac{dU_I}{dt} = -\frac{1}{\Delta x} (\mathcal{F}_{I+1/2} - \mathcal{F}_{I-1/2}), \quad (32)$$

where

$$\mathcal{F}_{I+1/2} = F_{I+1/2} + F_{I+1/2}^{\text{NN}} - \mathcal{D}_{I+1/2}^{\text{NN}}. \quad (33)$$

Here,

$$F_{I+1/2} = \frac{U_I^2 + U_{I+1}^2}{4} \quad (34)$$

is the central macroscopic flux, and

$$\mathcal{D}_{I+1/2}^{\text{NN}} = C_{\theta, I+1/2} \frac{\alpha_{I+1/2}}{2} (U_{I+1} - U_I), \quad (35)$$

where

$$\Lambda_{I+1/2} = \max\{|U_I|, |U_{I+1}|\}, \quad \alpha_{I+1/2} = \max\{\Lambda_{I+1/2}, \varepsilon\}, \quad \varepsilon > 0.$$

Also, assume that the neural corrective flux is constructed using the convex Entropy Neural Network η_θ and the Flux Potential Neural Network ϕ_θ as follows:

$$\mathcal{N}(U_{I+1}, U_I) := F_{I+1/2}^{\text{NN}} = \frac{1}{2} [\phi'_\theta(Z_I) + \phi'_\theta(Z_{I+1})], \quad Z_I = \eta'_\theta(U_I). \quad (36)$$

Note that here we define the “internal” entropy variables Z_I that, in general, are not the same as the entropy variables for the Burgers’ equation.

If, for every interface,

$$C_{\theta, I+1/2} \alpha_{I+1/2} \geq \frac{|U_{I+1} - U_I|}{6} + 2L_{\text{nn}}, \quad (37)$$

then the semi-discrete scheme satisfies a discrete entropy inequality

$$\frac{d}{dt}\eta(U_I) + \frac{Q_{I+1/2} - Q_{I-1/2}}{\Delta x} \leq 0, \quad (38)$$

for a suitable numerical entropy flux $Q_{I+1/2}$.

The proof follows Tadmor's entropy stability framework. We first identify the entropy-conservative reference flux and then estimate the entropy production associated with each component of the numerical flux. The Lipschitz regularity assumption is used to bound the entropy production contributed by the neural corrective flux. The adaptive Eddy Viscosity is finally shown to dominate the remaining positive entropy production.

Proof. Following Tadmor's framework, we compare the composite numerical flux with the entropy-conservative flux for Burgers' equation. The entropy production associated with each component of the numerical flux is then estimated separately.

For the Burgers' equation, the physical flux, entropy function, and entropy flux are given by

$$f(u) = \frac{1}{2}u^2, \quad \eta(u) = \frac{1}{2}u^2, \quad q(u) = \frac{1}{3}u^3,$$

respectively. The entropy variable is $v = u$, and the entropy potential is

$$\psi(u) = vf(u) - q(u) = \frac{1}{6}u^3.$$

By Tadmor's condition [31], a two-point numerical flux $F_{I+1/2}^{\text{EC}}$ is entropy-conservative if and only if it satisfies

$$(V_{I+1} - V_I) F_{I+1/2}^{\text{EC}} = \psi_{I+1} - \psi_I. \quad (39)$$

Therefore, using entropy variables and the entropy potential for the Burgers' equation, $V_I = U_I$, and

$$F_{I+1/2}^{\text{EC}} = \frac{\psi(U_{I+1}) - \psi(U_I)}{U_{I+1} - U_I} = \frac{U_I^2 + U_I U_{I+1} + U_{I+1}^2}{6}.$$

A scheme satisfies (38) if and only if the total entropy production at each interface,

$$\mathcal{E}_{I+1/2} := (V_{I+1} - V_I) (\mathcal{F}_{I+1/2} - F_{I+1/2}^{\text{EC}}), \quad (40)$$

satisfies $\mathcal{E}_{I+1/2} \leq 0$. We evaluate the contribution of each component of (33) to $\mathcal{E}_{I+1/2}$ next.

Macroscopic Flux. We first consider the contribution of the central macroscopic flux. Since this flux is not entropy conservative, it generates positive entropy production that must later be compensated by numerical dissipation.

The difference between the central macroscopic flux and the entropy-conservative flux is

$$F_{I+1/2} - F_{I+1/2}^{\text{EC}} = \frac{U_I^2 + U_{I+1}^2}{4} - \frac{U_I^2 + U_I U_{I+1} + U_{I+1}^2}{6} = \frac{(U_{I+1} - U_I)^2}{12}.$$

Thus, the entropy production associated with the central macroscopic flux is

$$(U_{I+1} - U_I) (F_{I+1/2} - F_{I+1/2}^{\text{EC}}) = \frac{(U_{I+1} - U_I)^3}{12}. \quad (41)$$

Neural Corrective Flux. Next, we estimate the entropy contribution of the neural corrective flux. Unlike the macroscopic flux, this contribution depends on the learned neural approximation and is controlled using the Lipschitz regularity assumption.

From (36), the Neural Corrective Flux satisfies the neural entropy production equation

$$\mathcal{R}_{I+1/2} := (V_{I+1} - V_I) F_{I+1/2}^{\text{NN}} = (U_{I+1} - U_I) \frac{\phi'_\theta(Z_I) + \phi'_\theta(Z_{I+1})}{2}. \quad (42)$$

Next, using the definition of the neural flux, we can write $\phi'_\theta(Z(U)) = \mathcal{N}(U, U)$, and using Assumption 1, we can bound the difference $|\phi'_\theta(Z(U_{I+1})) - \phi'_\theta(Z(U_I))| = |\mathcal{N}(U_{I+1}, U_{I+1}) - \mathcal{N}(U_I, U_I)| \leq 2L_{\text{nn}}|U_{I+1} - U_I|$. Therefore,

$$|\mathcal{R}_{I+1/2}| \leq L_{\text{nn}}(U_{I+1} - U_I)^2. \quad (43)$$

Dissipative Flux and Entropy Production. Combining the previous estimates, the total entropy production of the composite numerical flux is bounded by

$$\mathcal{R}_{I+1/2} + \frac{(U_{I+1} - U_I)^3}{12} - (U_{I+1} - U_I) \mathcal{D}_{I+1/2}^{\text{NN}}.$$

Treating the worst-case scenario when the neural entropy production is $\mathcal{R}_{I+1/2} > 0$, the Tadmor entropy stability condition is therefore

$$(U_{I+1} - U_I) \mathcal{D}_{I+1/2}^{\text{NN}} \geq \frac{(U_{I+1} - U_I)^3}{12} + |\mathcal{R}_{I+1/2}|. \quad (44)$$

Using (35), we obtain $(U_{I+1} - U_I) \mathcal{D}_{I+1/2}^{\text{NN}} = \frac{1}{2} C_{\theta, I+1/2} \alpha_{I+1/2} (U_{I+1} - U_I)^2$. The Eddy Viscosity term therefore compensates both the entropy production of the central flux and the worst-case positive contribution of the neural corrective flux. Thus, a sufficient pointwise condition for (44) is

$$C_{\theta, I+1/2} \frac{\alpha_{I+1/2}}{2} (U_{I+1} - U_I)^2 \geq \frac{|U_{I+1} - U_I|^3}{12} + L_{\text{nn}}(U_{I+1} - U_I)^2,$$

which is equivalent, for $U_{I+1} \neq U_I$, to

$$C_{\theta, I+1/2} \alpha_{I+1/2} \geq \frac{|U_{I+1} - U_I|}{6} + 2L_{\text{nn}}. \quad (45)$$

If $U_{I+1} = U_I$, the inequality is trivially satisfied using the definition of $\mathcal{R}_{I+1/2}$ in (42).

Hence, the adaptive dissipation dominates all possible positive entropy production, implying the discrete entropy inequality (38). $\square$

Condition (37) requires the adaptive Eddy Viscosity to provide enough dissipation to offset both the entropy generated by the central discretization and the additional entropy production that may arise from the neural corrective flux.

Consider the lower bound with $\mathcal{R}_{I+1/2} \leq 0$. Then, for $\Lambda_{I+1/2} \geq \varepsilon$, the condition (37) is equivalent to

$$C_{\theta, I+1/2} \geq \frac{|U_{I+1} - U_I|}{6 \Lambda_{I+1/2}}$$

with $\Lambda_{I+1/2} = \max(|U_{I+1}|, |U_I|)$. The triangle inequality gives $|U_{I+1} - U_I| \leq |U_I| + |U_{I+1}| \leq 2\Lambda_{I+1/2}$. Thus, the standard sufficient condition for discrete entropy stability of the central flux approximation becomes

$$C_{\theta, I+1/2} \geq \frac{2\Lambda_{I+1/2}}{6\Lambda_{I+1/2}} = \frac{1}{3}.$$

Thus, condition (37) (same as eq. (45)) is essentially equivalent to

$$C_{\theta, I+1/2} \geq \frac{1}{3} + \frac{2L_{nn}}{\Lambda_{I+1/2}}.$$

Therefore, we select $C_{\min} = 0.35$ as the lower bound for the Eddy Viscosity network. Numerically, as shown in the next section, this bound is sufficient to ensure that no oscillations develop near shocks.

6. Numerical Results

In this section, the performance of the neural network-based reduced model is evaluated by comparing its stationary statistical properties with those of the full model. We perform long stationary simulations for the full and reduced models and perform a detailed comparison of several statistical properties. The time step for the simulations of the reduced models is consistent with the time step used in the numerical integration of the full model.

For our numerical experiments with the Burgers' equation, we used the fine-mesh discretization $N_f = 512$ in simulations of the full model. This high-resolution grid ensures that all relevant physical features are captured. The numerical diffusion in this fine-mesh setup is negligible, meaning further refinement would yield minimal improvements. We generate coarse-grid representations with $q = 8$. Simulations on the fine-mesh grid are referred to as DNS (Direct Numerical Simulation), serving as a reference for comparison. Reduced model simulations are denoted as LLF-64 for the Local Lax–Friedrichs method with 64 points, and as NN-64 when the neural network subgrid model is used. Since the neural network framework is structure-preserving, there is no risk that the NN-reduced model may generate non-physical behaviors.

Energy Spectra. To quantify the model's statistical accuracy, we evaluate the distribution of kinetic energy across different spatial scales. The energy spectrum is computed in Fourier space by applying the discrete Fourier transform (DFT) to the state variable $u(x, t)$ at each recorded time step, yielding the Fourier coefficients $\hat{u}_k(t)$ for each wavenumber k . The time-averaged spectral energy is calculated as

$$E_k = \frac{1}{T} \int_0^T |\hat{u}_k(t)|^2 dt \approx \frac{1}{N_T} \sum_{i=1}^{N_T} |\hat{u}_k(t_i)|^2$$

We generate a long, stationary time series to ensure statistical convergence. The system is integrated for $T = 1000$ using a time step of $\Delta t = 0.001$. To compute the time-averaged spectra, we sample data with $\Delta t^{\text{sample}} = 0.1$, yielding $N_T = 10,000$ discrete snapshots. The total macroscopic energy of the system is the sum of the spectral energy over all wavenumbers: $E_{\text{total}} = \sum_k E_k$.

Figure 2 compares the time-averaged energy spectra of the high-resolution DNS, the low-resolution LLF-64 simulation, the TVD-64 scheme by van Leer [42], and the simulation of the reduced NN-64 model. The LLF-64 scheme is clearly too diffusive, artificially draining energy from wavenumbers starting with $|k| \geq 6$ and causing a premature spectral decay. The TVD-64 scheme is less diffusive than the LLF-64 simulation; nevertheless, it still underestimates the spectra and the total energy at the resolution $N_c = 64$. This clearly demonstrates the need for subgrid modeling if the reduced resolution $N = 64$ is the target. The NN-64 reduced model accurately resolves the inertial range, tracking the DNS energy cascade precisely up to the grid cut-off without exhibiting spurious energy accumulation. We present the relative error for the total energy in each system in Table 2. The NN-64 subgrid modeling framework yields a low relative error compared to the LLF-64 model. This confirms that the localized Eddy Viscosity network correctly balances energy dissipation, ensuring physical conservation principles are maintained at the macroscopic level.

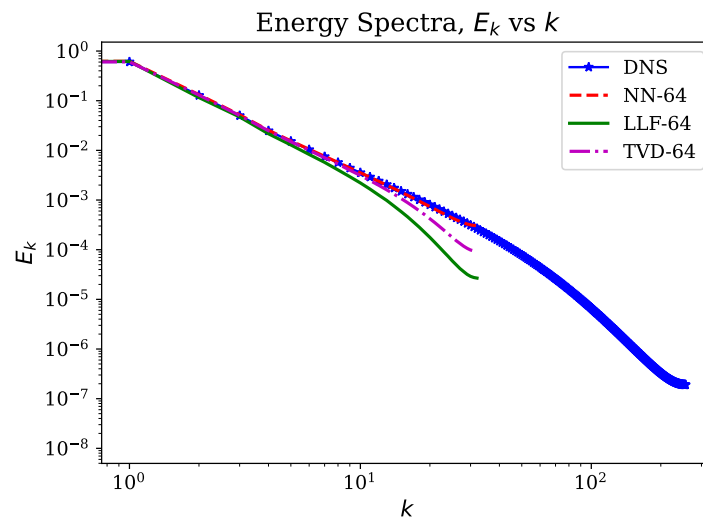


Figure 2. Energy spectra E_k in simulations of the fully resolved model with $N_f = 512$, reduced NN-64 model, TVD-64 scheme, and coarse LLF-64 simulation. The NN-64 parameterization effectively corrects the excessive numerical dissipation inherent to the coarse LLF-64 and TVD-64 schemes, perfectly recovering the energy cascade.

Table 2. Comparison of total energy and the relative error for the different numerical models.

Model	Total Energy (E_{total})	Relative Error with DNS
DNS	0.8830	
NN-64 Model	0.8746	0.95%
LLF-64	0.8466	4.12 %
TVD-64	0.8696	1.5 %

The TVD-64 scheme represents the Monotonic Upstream-Centered Scheme for Conservation Laws (MUSCL), originally formulated by van Leer [42]. The MUSCL scheme extends the first-order Godunov approach to achieve second-order spatial accuracy via a piecewise linear reconstruction of the state variables at the cell interfaces. In our implementation, we employ the classical van Leer limiter [43], which dynamically enforces the Total Variation Diminishing (TVD) condition by reverting to first-order accuracy in regions of steep gradients or local extrema. The reconstructed interface states are subsequently evaluated using the Local Lax–Friedrichs (LLF) solver to compute the numerical fluxes, ensuring a robust and entropy-stable baseline for performance comparison.

6.1. Comparison with Other Subgrid-Scale Models

To explicitly demonstrate the limitations of traditional closures, we compare the proposed neural architecture against classical static and dynamic Smagorinsky models [3,4,44]. Both models are applied as explicit corrective fluxes on top of the Local Lax–Friedrichs (LLF) [28] baseline scheme on the coarse $N = 64$ grid.

In addition, the Wasserstein Generative Adversarial Network (WGAN) [45] and Stochastic Mode Reduction (SMR) [13] have been used to model subgrid processes for the same coarse variables (6) in the finite-volume discretization of the forced 1D Burgers' equation. Overall, the WGAN and SMR parameterizations perform well in reproducing the spectra and temporal correlations. However, training the WGAN deep learning model was particularly challenging. The SMR approach also performed well, but it slightly overestimated the energy of higher wavenumbers in the spectra. Moreover, we found that when coarse variables are defined as local spatial averages, it may be challenging to compute SMR interaction coefficients for systems of equations and multi-dimensional problems since the

SMR approach is a semi-analytical technique that requires knowledge of the statistics of fluctuations $y_i = u_i - U_{I(i)}$. Finally, an analytical investigation of both the WGAN and SMR parameterizations is extremely challenging.

Static Smagorinsky. The classical static Smagorinsky model [3] computes artificial viscosity directly proportional to the absolute local velocity gradient, given by the formulation $\nu_{sgs} = (C_s \Delta x)^2 \left| \frac{\partial u}{\partial x} \right|$. We first evaluate the static Smagorinsky model across a range of fixed coefficients. Figure 3 presents the energy spectra for $C_s \in \{0.01, 0.05, 0.15, 0.20\}$. For the inviscid Burgers' equation, this spatial gradient is naturally maximized at the shock front. Consequently, the static model applies its heaviest numerical damping exactly where the wave is steepest. For artificially low values ($C_s = 0.01$), the subgrid contribution is negligible, leaving the overly dissipative LLF baseline unchanged. However, as C_s approaches standard theoretical values ($C_s = 0.15$ and 0.20), the excess viscosity severely degrades the wave amplitude.

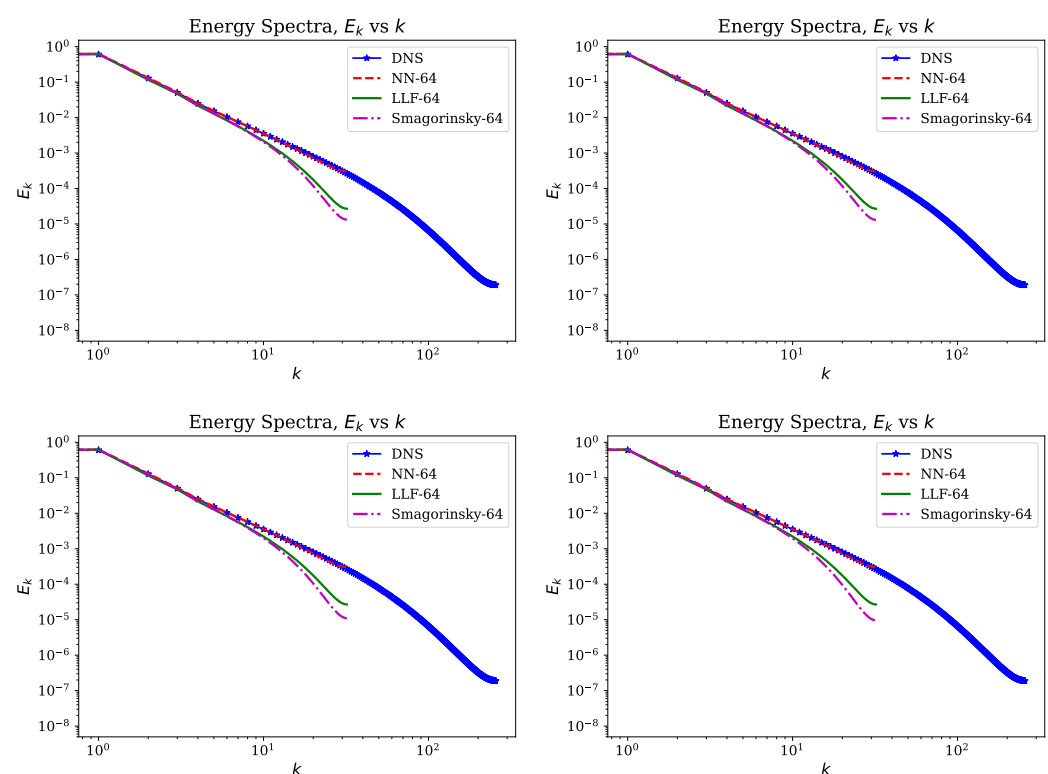


Figure 3. Energy spectra E_k in simulations of the fully resolved model with $N_f = 512$ (blue), reduced NN-64 model (red; nearly overlaps with blue), coarse LLF-64 simulation (green), and Static Smagorinsky model (magenta) with $C_s = 0.01$ (top left), $C_s = 0.05$ (top right), $C_s = 0.15$ (bottom left), and $C_s = 0.2$ (bottom right). The NN parameterization effectively corrects the excessive numerical dissipation inherent to the coarse LLF-64 scheme, perfectly recovering the energy cascade.

Dynamic Smagorinsky Model. We next evaluate the Dynamic Smagorinsky Model (DSM) [4,44]. Unlike the static variant, DSM dynamically computes the coefficient $C_s(x, t)$ utilizing a dual-filtering procedure. By defining a grid-filter scale Δ and a test-filter scale $\hat{\Delta}$ (where typically $\hat{\Delta} = 2\Delta$), the resolved turbulent stresses at the test-filter level, T , and the grid-filter level, τ , are related exactly by Germano's identity: $\mathcal{L} = T - \hat{\tau} = \widehat{uu} - \widehat{\widehat{uu}}$, where $\mathcal{L}$ denotes the resolved Leonard stress.

Applying the Smagorinsky closure at the grid-filter and test-filter levels gives

$$\mathcal{L} \approx C_s^2 M, \quad \text{where} \quad M = -2\hat{\Delta}^2 \left| \partial_x \widehat{u} \right| \partial_x \widehat{u} + 2\Delta^2 \left| \partial_x \widehat{\widehat{u}} \right| \partial_x \widehat{\widehat{u}}.$$

Following the least-squares argument of Lilly [44], the dynamic coefficient is obtained by minimizing the Germano-identity residual, $(\mathcal{L} - C_s^2 M)^2$. In the present one-dimensional setting, this yields the local scalar estimate

$$C_s^2(x, t) = \frac{\mathcal{L}(x, t)M(x, t)}{M^2(x, t)},$$

whenever $M \neq 0$. Equivalently, in strictly scalar form, this reduces to $C_s^2 = \mathcal{L}/M$. In practice, one may introduce spatial, temporal, or ensemble averaging in the numerator and denominator to regularize this estimate.

Figure 4 illustrates the failure of this local DSM formulation for the 1D Burgers' test case considered here. The dynamic procedure assumes scale similarity of the modeled SGS stress between the grid-filter and test-filter levels. This assumption is reasonable in inertial-range turbulence but becomes questionable near under-resolved Burgers' shocks, where the filtered gradient is dominated by the shock thickness rather than by a self-similar cascade. Across such sharp interfaces, the Leonard stress $\mathcal{L} = \widehat{u}^2 - \widehat{\widehat{u}}^2$ measures the finite resolved variance introduced by the test filter, while the modeled tensor M is constructed from nonlinear functions of the filtered gradient. The resulting local ratio $C_s^2 = \mathcal{L}M/M^2$ can therefore become poorly conditioned or change sign near shocks.

When $\mathcal{L}M < 0$, the dynamic procedure yields a negative Eddy Viscosity, corresponding to local backscatter or anti-diffusion. In a shock-dominated Burgers' solution, such anti-diffusive contributions are numerically dangerous because they can oppose the entropy-dissipative regularization needed to obtain the physically admissible solution. Applying ad hoc clipping removes these negative-viscosity regions, but it does not control large positive values of C_s^2 . Consequently, localized peaks in the dynamic coefficient near the shock can produce excessive artificial viscosity, leading to over-damping and distortion of the resolved wave.

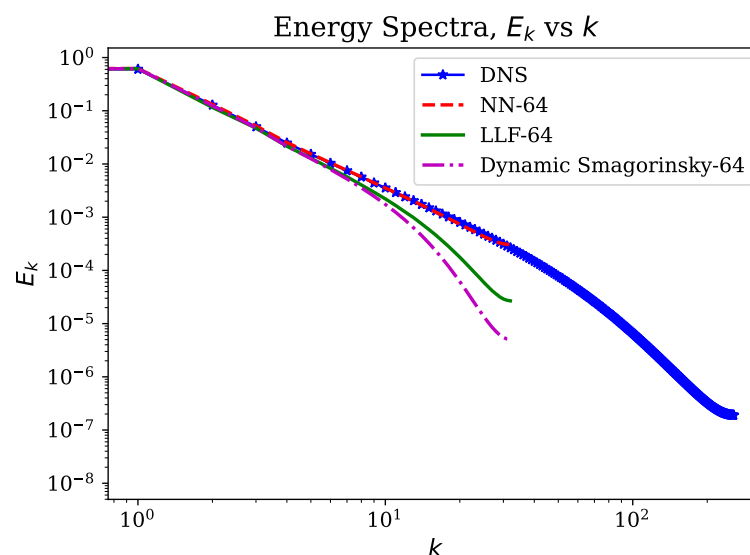


Figure 4. Energy spectra E_K in simulations of the fully resolved model with $N_f = 512$ (blue), reduced NN-64 model (red; nearly overlaps with blue), coarse LLF-64 simulation (green), and Dynamic Smagorinsky model (magenta). The failure of the Dynamic Smagorinsky Model (DSM) is likely caused by applying the test filter across sharp discontinuities, where the scale-similarity assumption breaks down; the resulting Leonard stress is dominated by the finite shock jump, leading to poorly conditioned dynamic-coefficient estimates and localized excess viscosity that distorts the resolved wave structure.

Spatial Correlation Function. To validate how well spatial structures are captured by the reduced model, we analyze the two-point spatial correlation function. This statistical metric quantifies the dependence and spatial coherence between different locations across the domain.

For the scalar state variable $U(x, t)$, the time-averaged spatial correlation evaluated at a discrete grid point X_I is defined as

$$C(X_I) = \frac{1}{N_T} \sum_{k=1}^{N_T} U(X_0, t_k) U(X_I, t_k), \quad (46)$$

where N_T represents the total number of sampled time snapshots. Because the stochastically forced system is spatially homogeneous and operates within a periodic domain, its statistical properties are translationally invariant. Therefore, computing the correlation with the first spatial grid point X_0 is sufficient to characterize the entire domain.

Figure 5 (left) presents the normalized spatial correlation function for the resolved flow field. The curve exhibits the characteristic symmetric profile expected in a periodic domain, reflecting the spatial decorrelation length and the recovery of coherency across the periodic boundary. Both the reduced NN-64 and coarse LLF-64 models show agreement with the high-resolution DNS reference. This implies that the NN corrections are not affecting the spatial correlation significantly, and the reduced model accurately reproduces the spatial correlation scales. This confirms that our structure-preserving architecture successfully captures the macroscopic spatial scales and physical coherence of the flow field, despite operating on a truncated grid resolution.

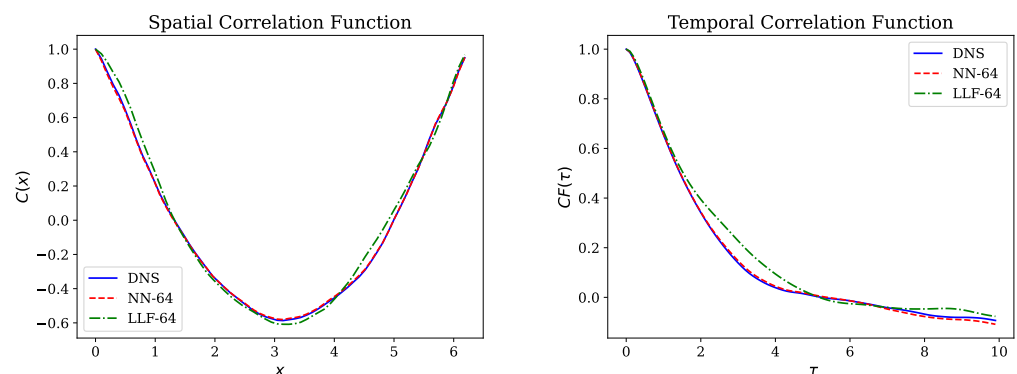


Figure 5. (Left): Spatial correlation function for coarse variables U_I given by (46), (Right): Temporal correlation function of coarse variables U_0 given by (47). Solid blue line: high-resolution DNS simulations, dashed green line: simulations of the LLF-64 model, dashed red line: simulations of the NN-64 reduced model.

Temporal Correlation Function. To evaluate the dynamical consistency of the reduced model, we compute the temporal correlation function. This statistical metric quantifies the system's memory. It indicates how rapidly dynamical variables decorrelate from their prior states. Since the Burgers' equation is spatially homogeneous, we consider only the spatial location $U(X_0, t)$. The time correlation function is defined as

$$CF(\tau) = \frac{1}{T} \int_0^T U(X_0, t) U(X_0, t + \tau) dt. \quad (47)$$

Computationally, the time-correlation function is computed using a moving-window approach.

The normalized temporal correlation functions for the coarse variables in NN-64, LLF-64, and DNS are presented in the right part of Figure 5. The NN-64 reduced model demonstrates very good agreement with the high-resolution DNS simulation. For the

temporal correlation function, a slight discrepancy is observed between the LLF-64 and DNS curves, indicating that the NN correction terms improve the temporal correlation statistics. The NN parameterization accurately captures the decorrelation timescale and the decay profile of the correlation function. Numerical results with spatial and temporal correlation functions confirm that the reduced-order framework preserves not only the spatial energy cascade but also the accurate statistical evolution and temporal dynamics of the full-scale system.

Solution Snapshots. Neural network approximations are often treated as data-driven “black-box” models without incorporating physical constraints. As a consequence, neural network approximations, when left uncontrolled, can produce unbounded oscillations or unrealistic solutions. One possible approach is to employ flux limiters [34,46] to suppress non-physical behavior and spurious oscillations. In this paper, we demonstrate that our approach does not require flux-limiting stabilization techniques. This is a direct consequence of using structure-preserving neural networks, which enforce stability at the architectural level. The structure-preserving neural network design used in this paper guarantees that the subgrid parameterization generates robust, physically consistent solutions, replacing the need for traditional flux-limiting procedures with rigorous architectural constraints.

Figure 6 depicts typical solution profiles for four selected times. For comparison, we also include the LLF-64 coarse-model results, which bear little resemblance to the DNS solution and fail to reproduce the relevant dynamics. We can see that the NN model demonstrates excellent agreement with the DNS, successfully capturing shock formations and nonlinear wave propagation. The embedded Eddy Viscosity network is essential in naturally suppressing spurious oscillations near steep gradients, eliminating the need for external flux limiters.

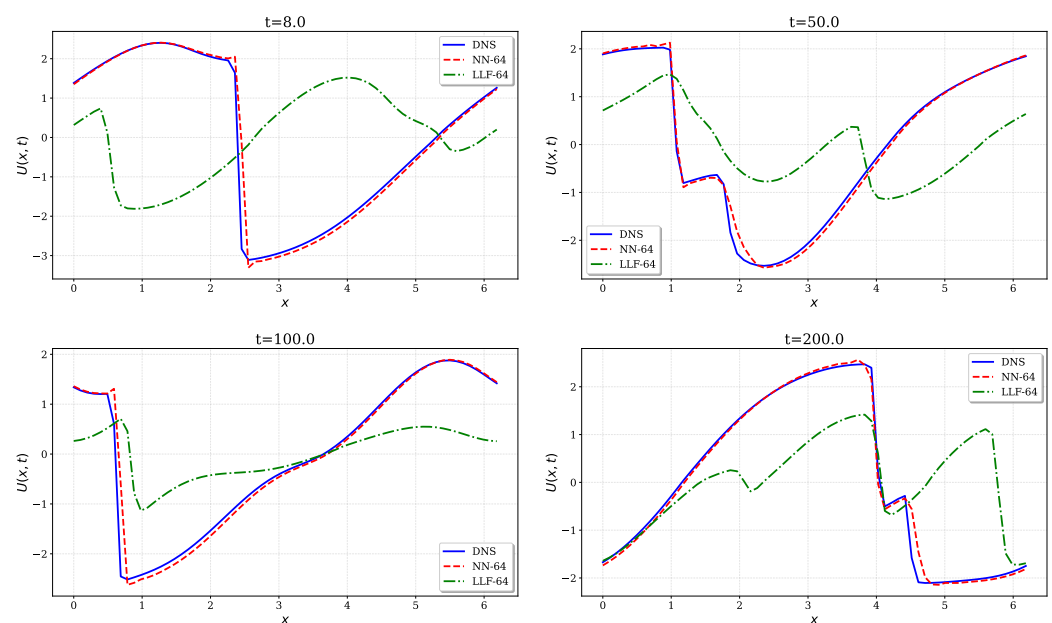


Figure 6. Comparison between velocity snapshots $U(x, t)$ for the Burgers' equation in simulations of the NN-64 reduced model (red dashed), LLF-64 (green dash-dot), and DNS-512 (blue solid) at times $t = 8, 50, 100, 200$.

6.2. Extrapolation Results

One important practical concern regarding neural networks is their ability to generalize beyond the training regime. To evaluate this, we apply the NN reduced model without retraining to different forcing regimes and initial conditions, analyzing its robustness and adaptability.

Larger Forcing Regimes. To assess the robustness and physical fidelity of our framework, we perform simulations of the reduced model with out-of-distribution forcing. In particular, the forcing amplitude is increased from the baseline training value of $A = 1.0$ to $A = 1.2$ and $A = 1.4$, representing increases of 20% and 40% in the energy injected into the system, respectively. Recall that forcing is given by (2). The forcing is injected into the larger scales and cascades down to the unresolved subgrid scales through nonlinear interactions. Accurate subgrid parameterization must dynamically adapt to this increased energy throughput to maintain stability and spectral accuracy.

The energy spectra for simulations with larger forcing are presented in Figure 7. The NN-64 reduced model accurately tracks the DNS energy cascade across the entire resolved wavenumber range. The model achieves this consistency across all forcing regimes without retraining. The localized Eddy Viscosity network, bounded by physical constraints, successfully adjusts the subgrid dissipation dynamically. It applies the precise amount of dissipation required to prevent spurious oscillations while preserving the high-wavenumber tail. We also verified that solution snapshots are reproduced very well in regimes with increased forcing. Figure 8 depicts snapshots in simulations with $A = 1.4$. The same initial condition and forcing realization are used in simulations with $A = 1$ depicted in Figure 6.

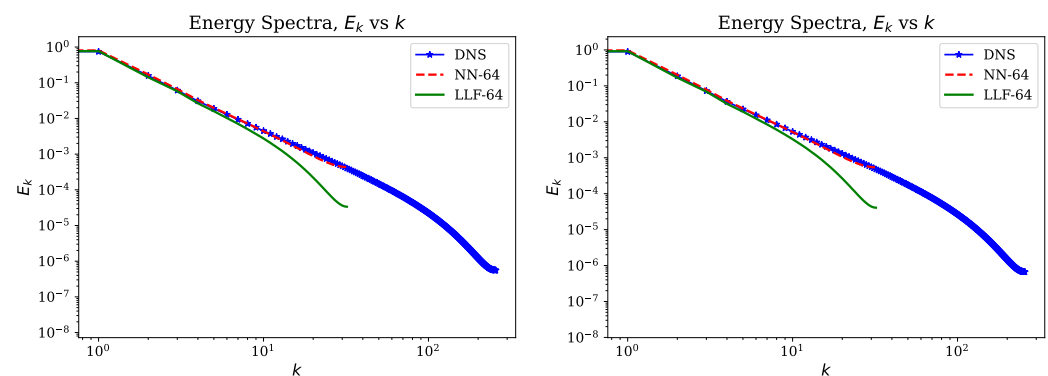


Figure 7. Energy spectra of DNS, NN-64, and LLF-64 models with increased forcing $A = 1.2$ (left) and $A = 1.4$ (right). The NN-64 reduced model is simulated without retraining the NN parameterization.

As expected, the overall magnitude of the solution increases with larger forcing, but all networks generalize well in this regime. In particular, the Eddy Viscosity network suppresses spurious oscillations near shocks, while the structure-preserving network models the nonlinear wave interactions. Our numerical results confirm that all three networks successfully learned the physics of the nonlinear energy cascade, allowing them to generalize robustly beyond the original training regime.

Step Initial Conditions. To further evaluate how well our framework generalizes to out-of-sample regimes, we consider discontinuous initial conditions. While the neural network parameterization was trained exclusively on smooth initial wave profiles, we evaluate its performance here on a discontinuous step function (a Riemann-type problem) without any retraining. The specific initial condition considered here is

$$u(x, 0) = \begin{cases} 0.5, & \frac{2\pi}{3} \leq x < \frac{4\pi}{3} \\ -0.5, & \text{otherwise} \end{cases}$$

Handling sharp discontinuities can be particularly challenging for machine learning parameterizations. Black-box models trained exclusively on smooth data typically suffer from severe, non-physical Gibbs oscillations or instability when encountering shocks (see,

e.g., [34,46–49]). In contrast, our structure-preserving NN parameterization remains remarkably stable and does not require any adjustments for discontinuous data.

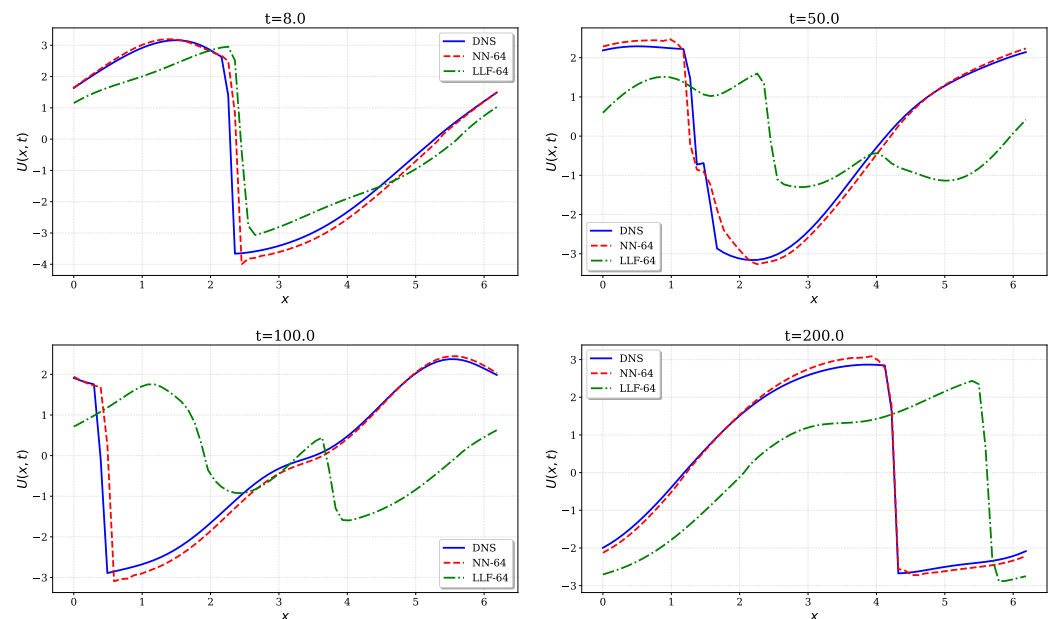


Figure 8. Comparison between velocity snapshots $U(x, t)$ for the Burgers' equation in simulations of the NN-64 reduced model (red dashed), LLF-64 (green dash-dot), and DNS-512 (blue solid) with increased forcing $A = 1.4$ at times $t = 8, 50, 100, 200$. Compare with simulations with $A = 1$ in Figure 6. Note that the NN parameterization was trained on the baseline regime $A = 1$ and is evaluated here without retraining.

Figure 9 compares the DNS, NN-64, and LLF-64 solution snapshots for the discontinuous initial condition at $t = 0.1, 10, 40, 50$. At early simulation times (e.g., $t = 0.1$), the reduced-order model accurately captures the sharp, near-vertical gradients of the initial step without overshoots. As the simulation evolves ($t = 10, 40, 50$), the discontinuities propagate, interact, and decay. Throughout this evolution, the NN-64 reduced model consistently reproduces the correct shock speed and amplitude of the high-resolution DNS reference.

This successful extrapolation highlights the robustness of our structure-preserving NN parameterization approach and the embedded Eddy Viscosity network in particular. Because the architecture enforces physical bounds and localized dissipation dynamically, this network successfully recognizes and stabilizes steep gradients even in a physical regime absent from the training data. Overall, the NN-64 parameterization provides a substantial improvement not only for averaged quantities but also for individual solution trajectories.

The training time for the neural network model is approximately 12 min. This fast training time reflects the design (relatively small depth and width) of all three neural networks.

Table 3 reports the inference cost for the fully resolved DNS model with $N_f = 512$, the coarse LLF-64 and NN-64 models using the same time step as the DNS, and the NN-64 model using a time step four times larger, $4\Delta t$. The inference time of the coarse NN-64 model using the smaller time step is greater than that of the DNS because evaluating the neural networks requires additional operations. However, the coarser spatial resolution allows the use of a larger time step, which substantially reduces the inference cost of the NN-64 model. We verified that simulations performed with the larger time step reproduce the same statistical quantities and trajectory predictions. We expect the relative computational savings to become more significant for multidimensional problems, where the cost of fully resolved simulations increases rapidly with spatial resolution.

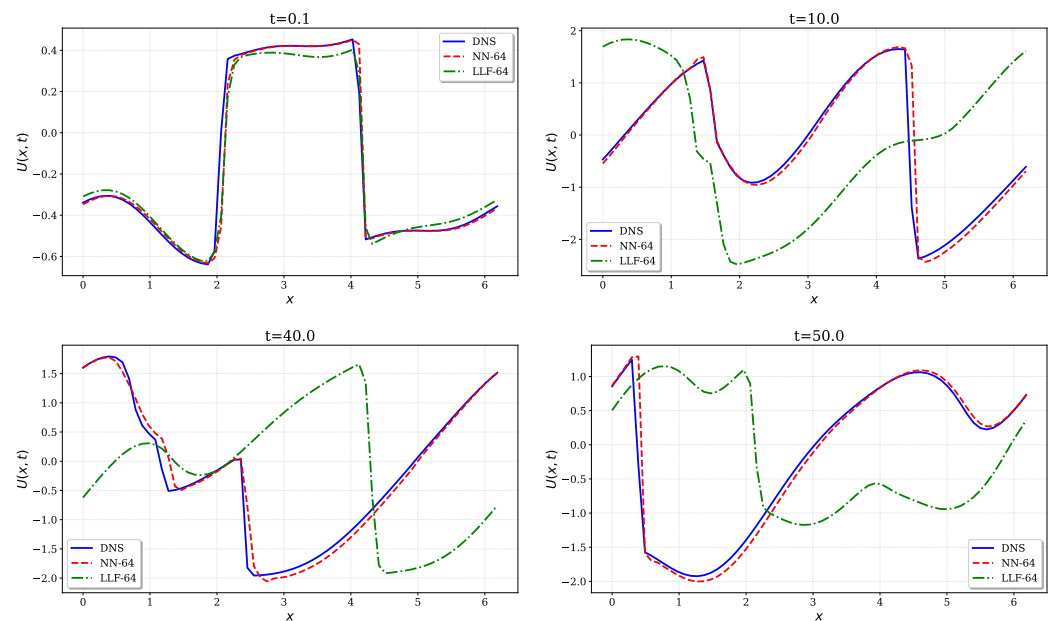


Figure 9. Comparison between velocity snapshots $U(x, t)$ for the Burgers' equation in simulations of the NN-64 reduced model (red dashed), LLF-64 (green dash-dot), and DNS-512 (blue solid) with step initial conditions at times $t = 0.1, 10, 40, 50$. Note that the NN parameterization was trained on solutions with smooth initial conditions and is evaluated here without retraining.

Table 3. Inference wall-clock time for simulation with $T = 1000$ for the DNS ($N_f = 512$) and simulations on a uniform coarse grid ($N_c = 64$).

Scheme	Inference Time (s)
DNS	243.46
LLF-64	165.59
NN-64	329.90
NN-64 with $4\Delta t$	80.87

7. Conclusions

In this paper, we develop a structure-preserving neural network (NN) approach for parameterizing subgrid physical processes in coarse simulations of hyperbolic conservation laws. This structure-preserving approach relies on the underlying analytical properties of hyperbolic systems, but does not require explicit knowledge of the entropy function. Instead, the entropy function is learned using an input-convex neural network. Subgrid fluxes are then estimated using another network to learn the flux potential. A third neural network approximates the Eddy Viscosity and has design features similar to traditional Eddy Viscosity models. The main contribution of this work is the formulation and numerical validation of a structure-preserving neural-network framework for subgrid closure. The novelty lies in adapting this methodology to represent unresolved fluxes in a prescribed coarse discretization and in combining the learned entropy function and flux potential with a neural-network Eddy Viscosity closure.

We demonstrate that our NN reduced model performs very well and reproduces the statistical properties of solutions with high accuracy. In particular, there is no over-damping at higher wavenumbers in the energy spectra of coarse variables U_I (see Figure 2), which is usually difficult to achieve. We also demonstrate that the NN parameterization is structurally stable and that the solutions of the NN reduced model do not develop spurious oscillations. Consequently, the proposed NN parameterization approach does not require additional stabilization techniques.

Numerical results in Section 6.2 show that the NN reduced model generalizes well outside the training regime. In simulations with larger forcing, both the total energy and the magnitude of solutions increase, and the NN parameterization operates outside of the training regime. The NN reduced model reproduces the energy spectra and individual solutions in simulations with increased forcing very well. Numerical simulations with discontinuous initial conditions also demonstrate that the NN reduced model generalizes well to regimes with discontinuous initial data. Results with larger forcing are particularly relevant for potential applications in atmospheric fluid dynamics, where it is important to consider increased atmospheric forcing on the ocean due to climate change scenarios.

The numerical comparisons with the static and dynamic Smagorinsky models further demonstrate the advantages of the proposed formulation. For the Burgers' equation test cases considered here, which include solutions with shocks, the static model introduces excessive dissipation, whereas the dynamic model can produce excessively large stresses because its scale-invariance assumption is not well suited to these cases. The comparison with the second-order TVD method also provides a conventional shock-capturing baseline. These results show that the proposed closure improves the representation of unresolved dynamics relative to the classical models considered in this study.

The analytical results in Section 5 demonstrate that the resulting NN parameterization is entropy stable and provide useful guidelines for selecting the range for the diffusion constant in the Eddy Viscosity network. This highlights the advantage of a structure-preserving approach to data-driven flux estimation in the reduced model. In contrast to black-box closure strategies, the proposed formulation preserves key stability properties while retaining the flexibility of neural-network-based parameterizations. This combination of analytical tractability, numerical robustness, and improved predictive accuracy makes the approach particularly attractive for reduced-order modeling of hyperbolic conservation laws.

Overall, the NN parameterization yields a considerable improvement in the accuracy of both the energy spectra and individual solution trajectories. The model also extrapolates well to regimes with stronger forcing, suggesting that the learned closure captures robust coarse-scale dynamics rather than merely interpolating within the training distribution. A further advantage is that accurate performance is obtained using relatively small neural networks, making the approach computationally lightweight and straightforward to train compared, for instance, with WGAN-based closure models. Finally, because the parameterization is formulated at the level of the fluxes, it remains amenable to analytical investigation, providing a useful bridge between data-driven modeling and structure-preserving numerical analysis.

The structure-preserving approach allows the use of a small number of hidden layers and neurons in each layer. For instance, the Eddy Viscosity network has only one hidden layer. In addition, we expect that our approach can be extended to other hyperbolic problems, but applications to more complex equations, and especially to systems of equations, introduce additional network-design considerations. For instance, it is important to properly design the Eddy Viscosity network to avoid over-damping at higher wavenumbers of the coarse variables, which is a common problem in many closure models. Therefore, it is necessary to select an appropriate vector of feature variables ξ in (15), and possibly increase the number of layers or split the Eddy Viscosity network into several networks with scalar outputs to avoid inaccurate estimation of the vector $C(\xi)$. This requires a separate investigation and will be addressed in future work.

We do not foresee conceptual issues in extending our approach to systems of equations and multidimensional problems. Therefore, extending the structure-preserving framework to systems of conservation laws and multidimensional problems represents a natural and highly promising direction for future work. We expect that the overall architecture of the

Entropy Neural Network and the Flux Potential Neural Network would remain similar. However, it might be necessary to increase the number of layers and/or the number of neurons to incorporate multidimensional spatial stencils. Developing an efficient and accurate Eddy Viscosity network might require a separate investigation, as discussed in the previous paragraph. For multidimensional systems, the entropy function would depend on the full vector of state variables, and the neural-network architecture would need to account for fluxes in multiple spatial directions. The Eddy Viscosity network would also need to incorporate multidimensional spatial derivatives and interactions between different directions. The resulting increase in the number of state variables, spatial stencils, and model parameters may also lead to more demanding and potentially less well-conditioned optimization problems. Determining suitable architectures and training strategies that preserve entropy stability and provide adequate dissipation without introducing excessive numerical diffusion will therefore require a dedicated investigation. Moreover, extending the entropy-stability analysis would likely require additional structural assumptions on the neural networks, potentially imposing further constraints on their architecture.

Author Contributions: Conceptualization, I.T. and A.N.; Methodology, A.N. and I.T.; Software, A.N.; Investigation, A.N.; Data curation, A.N.; Writing—original draft, A.N.; Writing—review and editing, I.T.; Visualization, A.N.; Project administration, I.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: The authors thank Lu Zhang for helpful discussions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sagaut, P. *Large Eddy Simulation for Incompressible Flows: An Introduction*; Springer: Berlin/Heidelberg, Germany, 2006. <https://doi.org/10.1007/b137536>.
2. Lesieur, M.; Métais, O.; Comte, P. *Large-Eddy Simulations of Turbulence*; Cambridge University Press: Cambridge, UK, 2005. <https://doi.org/10.1017/cbo9780511755507.008>.
3. Smagorinsky, J. General circulation experiments with the primitive equations: I. The basic experiment. *Mon. Weather. Rev.* **1963**, *91*, 99–164. [https://doi.org/10.1175/1520-0493\(1963\)091<0099:gcewtp>2.3.co;2](https://doi.org/10.1175/1520-0493(1963)091<0099:gcewtp>2.3.co;2).
4. Germano, M.; Piomelli, U.; Moin, P.; Cabot, W.H. A dynamic subgrid-scale eddy viscosity model. *Phys. Fluids A Fluid Dyn.* **1991**, *3*, 1760–1765. <https://doi.org/10.1063/1.2739419>.
5. Quarteroni, A.; Manzoni, A.; Negri, F. *Reduced Basis Methods for Partial Differential Equations: An Introduction*; Springer: Berlin/Heidelberg, Germany, 2015. <https://doi.org/10.1007/978-3-319-15431-2>.
6. Benner, P.; Gugercin, S.; Willcox, K. Survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev.* **2015**, *57*, 483–531. <https://doi.org/10.1137/130932715>.
7. Rowley, C.W.; Dawson, S.T. Model reduction for flow analysis and control. *Annu. Rev. Fluid Mech.* **2017**, *49*, 387–417. <https://doi.org/10.1146/annurev-fluid-010816-060042>.
8. Mamonov, A.V.; Olshanskii, M.A. A priori analysis of a tensor ROM for parameter dependent parabolic problems. *SIAM J. Numer. Anal.* **2025**, *63*, 239–261. <https://doi.org/10.1137/23m1616844>.
9. Mamonov, A.V.; Olshanskii, M.A. Tensorial parametric model order reduction of nonlinear dynamical systems. *SIAM J. Sci. Comput.* **2024**, *46*, A1850–A1878. <https://doi.org/10.1137/23m1553789>.
10. Mamonov, A.V.; Olshanskii, M.A. Interpolatory tensorial reduced order models for parametric dynamical systems. *Comput. Methods Appl. Mech. Eng.* **2022**, *397*, 115122. <https://doi.org/10.1016/j.cma.2022.115122>.
11. Bin Mizan, M.R.; Olshanskii, M.; Timofeyev, I. A parametric tensor ROM for the shallow water dam break problem. *Comput. Fluids* **2026**, *309*, 107005. <https://doi.org/10.1016/j.compfluid.2026.107005>.
12. Majda, A.J.; Timofeyev, I.; Vanden-Eijnden, E. A Mathematics Framework for Stochastic Climate Models. *Comm. Pure Appl. Math.* **2001**, *54*, 891–974. <https://doi.org/10.1002/cpa.1014>.

13. Dolaptchiev, S.I.; Achatz, U.; Timofeyev, I. Stochastic closure for local averages in the finite-difference discretization of the forced Burgers equation. *Theor. Comput. Fluid Dyn.* **2013**, *27*, 297–317. <https://doi.org/10.1007/s00162-012-0270-1>.
14. Zacharuk, M.; Dolaptchiev, S.I.; Achatz, U.; Timofeyev, I. Stochastic subgrid-scale parametrization for one-dimensional shallow-water dynamics using stochastic mode reduction. *Q. J. R. Meteorol. Soc.* **2018**, *144*, 1975–1990. <https://doi.org/10.1002/qj.3396>.
15. Lino, M.; Fotiadis, S.; Bharath, A.A.; Cantwell, C.D. Current and emerging deep-learning methods for the simulation of fluid dynamics. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **2023**, *479*, 20230058. <https://doi.org/10.1098/rspa.2023.0058>.
16. Rasp, S.; Pritchard, M.S.; Gentine, P. Deep learning to represent subgrid processes in climate models. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 9684–9689. <https://doi.org/10.1073/pnas.1810286115>.
17. Bolton, T.; Zanna, L. Applications of deep learning to ocean data inference and subgrid parameterization. *J. Adv. Model. Earth Syst.* **2019**, *11*, 376–399. <https://doi.org/10.1029/2018MS001472>.
18. Chantry, M.; Christensen, H.; Dueben, P.; Palmer, T. Opportunities and challenges for machine learning in weather and climate modelling: Hard, medium and soft AI. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200083. <https://doi.org/10.1098/rsta.2020.0083>.
19. Brenowitz, N.D.; Bretherton, C.S. Spatially extended tests of a neural network parametrization trained by coarse-graining. *J. Adv. Model. Earth Syst.* **2019**, *11*, 2728–2744. <https://doi.org/10.1029/2019ms001711>.
20. Krasnopolsky, V.M.; Fox-Rabinovitz, M.S.; Chalikov, D.V. New approach to calculation of atmospheric model physics: Accurate and fast neural network emulation of longwave radiation in a climate model. *Mon. Weather. Rev.* **2005**, *133*, 1370–1383. <https://doi.org/10.1175/mwr2923.1>.
21. Dafermos, C.M. *Hyperbolic Conservation Laws in Continuum Physics*; Springer: Berlin/Heidelberg, Germany, 2005. https://doi.org/10.1007/3-540-29089-3_11.
22. Chen, Z.; Gelb, A.; Lee, Y. Learning the dynamics for unknown hyperbolic conservation laws using deep neural networks. *SIAM J. Sci. Comput.* **2024**, *46*, A825–A850. <https://doi.org/10.1137/22m1537333>.
23. Liu, L.; Zhang, L.; Gelb, A. Neural entropy-stable conservative flux form neural networks for learning hyperbolic conservation laws. *J. Comput. Phys.* **2026**, *553*, 114719. <https://doi.org/10.1016/j.jcp.2026.114719>.
24. Amos, B.; Xu, L.; Kolter, J.Z. Input convex neural networks. In *Proceedings of the International Conference on Machine Learning*; PMLR: Sydney, Australia, 2017; pp. 146–155. <https://doi.org/10.48550/arXiv.1609.07152>.
25. Burgers, J.M. A mathematical model illustrating the theory of turbulence. *Adv. Appl. Mech.* **1948**, *1*, 171–199. [https://doi.org/10.1016/s0065-2156\(08\)70100-5](https://doi.org/10.1016/s0065-2156(08)70100-5).
26. LeVeque, R.J. *Numerical Methods for Conservation Laws*; Springer: Berlin/Heidelberg, Germany, 1992; Volume 132. https://doi.org/10.1007/978-3-0348-8629-1_3.
27. Hesthaven, J.S. *Numerical Methods for Conservation Laws: From Analysis to Algorithms*; SIAM: Philadelphia, PA, USA, 2017. <https://doi.org/10.1137/1.9781611975109.ch2>.
28. Rusanov, V.V. Calculation of interaction of non-steady shock waves with obstacles. *J. Comput. Math. Math. Phys.* **1961**, *1*, 267–279. [https://doi.org/10.1016/0041-5553\(62\)90062-9](https://doi.org/10.1016/0041-5553(62)90062-9).
29. Gottlieb, S.; Ketcheson, D.I.; Shu, C.W. *Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations*; World Scientific: Singapore, 2011. https://doi.org/10.1142/9789814289276_0002.
30. Godlewski, E.; Raviart, P.A. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*; Springer: Berlin/Heidelberg, Germany, 2021; Volume 118. <https://doi.org/10.1007/978-1-0716-1344-3>.
31. Tadmor, E. The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Math. Comput.* **1987**, *49*, 91–103. <https://doi.org/10.1090/s0025-5718-1987-0890255-3>.
32. Tadmor, E. Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Acta Numer.* **2003**, *12*, 451–512. <https://doi.org/10.1017/s0962492902000156>.
33. Jiang, G.S.; Shu, C.W. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.* **1996**, *126*, 202–228. <https://doi.org/10.1006/jcph.1996.0130>.
34. Mojamder, M.A.H.; Xu, Z.; Wang, M.; Timofeyev, I. Parametrization of Subgrid Scales in Long-Term Simulations of the Shallow-Water Equations Using Machine Learning and Convex Limiting. *Fluids* **2026**, *11*, 75. <https://doi.org/10.3390/fluids11030076>.
35. Bradbury, J.; Frostig, R.; Hawkins, P.; Johnson, M.J.; Katariya, Y.; Leary, C.; Maclaurin, D.; Necula, G.; Paszke, A.; VanderPlas, J.; et al. JAX: Composable Transformations of Python+NumPy Programs; 2018. Available online: <https://github.com/google/jax> (accessed on 8 July 2026).
36. Ramachandran, P.; Zoph, B.; Le, Q.V. Searching for activation functions. *arXiv* **2017**, arXiv:1710.05941. <https://doi.org/10.48550/arXiv.1710.05941>.
37. Shu, C.W. Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws. *Lect. Notes Math.* **1998**, *1697*, 325–432. <https://doi.org/10.1007/BFb0096355>.
38. Cockburn, B.; Shu, C.W. Runge–Kutta discontinuous Galerkin Methods for Convection-Dominated Problems. *J. Sci. Comput.* **2001**, *16*, 173–261. <https://doi.org/10.1023/A:1012873910884>.

39. Sweby, P.K. High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws. *SIAM J. Numer. Anal.* **1984**, *21*, 995–1011. <https://doi.org/10.1137/0721062>.
40. Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2009. <https://doi.org/10.1007/b79761>.
41. Godunov, S.K. A Difference Scheme for Numerical Solution of Discontinuous Solution of Hydrodynamic Equations. *Mat. Sb.* **1959**, *47*, 271–306.
42. Van Leer, B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comput. Phys.* **1979**, *32*, 101–136. [https://doi.org/10.1016/0021-9991\(79\)90145-1](https://doi.org/10.1016/0021-9991(79)90145-1).
43. Van Leer, B. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. *J. Comput. Phys.* **1974**, *14*, 361–370. [https://doi.org/10.1016/0021-9991\(74\)90019-9](https://doi.org/10.1016/0021-9991(74)90019-9).
44. Lilly, D.K. A proposed modification of the Germano subgrid-scale closure method. *Phys. Fluids A Fluid Dyn.* **1992**, *4*, 633–635. <https://doi.org/10.1063/1.858280>.
45. Alcala, J.; Timofeyev, I. Subgrid-scale parametrization of unresolved scales in forced Burgers equation using generative adversarial networks (GAN). *Theor. Comput. Fluid Dyn.* **2021**, *35*, 875–894. <https://doi.org/10.1007/s00162-021-00581-z>.
46. Timofeyev, I.; Schwarzmann, A.; Kuzmin, D. Application of machine learning and convex limiting to subgrid flux modeling in the shallow-water equations. *Math. Comput. Simul.* **2025**, *238*, 163–178. <https://doi.org/10.1016/j.matcom.2025.04.031>.
47. Coutinho, E.J.; Zhao, H.; Cunha, A.M.; Afonso, S.; Mukherjee, S. Physics-informed neural networks with adaptive localized artificial viscosity. *J. Comput. Phys.* **2023**, *489*, 112265. <https://doi.org/10.1016/j.jcp.2023.112265>.
48. Fuks, O.; Tchepeli, H.A. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. *J. Mach. Learn. Model. Comput.* **2020**, *1*, 19–37. <https://doi.org/10.1615/jmachlearnmodelcomput.2020033905>.
49. Krishnapriyan, A.S.; Gholami, A.; Zhe, S.; Kirby, R.; Mahoney, M.W. Characterizing possible failure modes in physics-informed neural networks. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 26548–26560. <https://doi.org/10.48550/arXiv.2109.01050>.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.