Journal of Computational Physics 437 (2021) 110324

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A data-driven physics-informed finite-volume scheme for nonclassical undercompressive shocks

Deniz A. Bezgin [*], Steffen J. Schmidt, Nikolaus A. Adams

Department of Mechanical Engineering, Chair of Aerodynamics and Fluid Mechanics, Technical University of Munich, Boltzmannstraße 15, Garching 85748, Germany

ARTICLE INFO

Article history:
Available online 29 March 2021

Keywords:
Cubic flux
Machine learning
Undercompressive shocks
Truncation error
Convolutional neural network

ABSTRACT

We propose a data-driven physics-informed finite-volume scheme for the approximation of small-scale dependent shocks. Nonlinear hyperbolic conservation laws with non-convex fluxes allow nonclassical shock wave solutions. In this work, we consider the cubic scalar conservation law as representative of such systems. As standard numerical schemes fail to approximate nonclassical shocks, schemes with controlled dissipation and schemes with well-controlled dissipation have been introduced by LeFloch and Mohammadian and by Ernest and coworkers, respectively. Emphasis has been placed on matching the truncation error of the numerical scheme with physically relevant small-scale mechanisms. However, aforementioned schemes can introduce oscillations as well as excessive dissipation around shocks. In our approach, a convolutional neural network is used for an adaptive nonlinear flux reconstruction. Based on the local flow field, the network combines local interpolation polynomials with a regularization term to form the numerical flux. This allows to modify the discretization error by nonlinear terms. In a supervised learning task, the model is trained to predict the time evolution of exact solutions to Riemann problems. The model is physics-informed as it respects the underlying conservation law. Numerical experiments for the cubic scalar conservation law show that the resulting method is able to approximate nonclassical shocks very well. The adaptive reconstruction suppresses oscillations and enables sharp shock capturing. Generalization to unseen shock configurations, smooth initial value problems, and shock interactions is robust and shows very good results.

1. Introduction

Nonlinear hyperbolic partial differential equations (PDEs) describe many physical systems of interest, e.g. compressible fluid dynamics. Solutions to these equations admit discontinuous solutions, i.e. shock waves, which can develop from smooth initial conditions over time. Weak solutions to the conservation laws are sought which are no longer uniquely defined. It is well known that hyperbolic conservation laws may admit unphysical solutions. Incorporating small-scale mechanisms, such as diffusion or dispersion, leads to regularized systems of conservations laws [1,2]. Physically consistent solutions to the hyperbolic conservation laws result from vanishing small-scale limits of the corresponding regularized systems.

For a wide class of conservation laws, a suitable entropy condition can be found which is consistent with the underlying small-scale effects. These entropy conditions are sufficient to single out the physically consistent solution [3]. Many numer-

[*] Corresponding author.
E-mail addresses: deniz.bezgin@tum.de (D.A. Bezgin), steffen.schmidt@tum.de (S.J. Schmidt), nikolaus.adams@tum.de (N.A. Adams).

https://doi.org/10.1016/j.jcp.2021.110324

ical methods are available for approximating entropy-satisfying solutions of hyperbolic conservation laws [4]. Amongst the most popular methods are finite-volume methods which evolve cell averages in time by computing cell interface fluxes. Interface fluxes can be obtained from exact or approximate Riemann solvers. High-order spatial accuracy can be achieved by nonlinear adaptive reconstruction methods, e.g. essentially non-oscillatory (ENO) [5], weighted-essentially non-oscillatory (WENO) [6], and targeted ENO (TENO) [7,8] schemes.

However, there are many physical systems whose shock solutions explicitly depend on the underlying small-scale mechanisms [2]. Examples include conservative hyperbolic systems with dispersive phenomena, nonconservative hyperbolic systems, and boundary layer problems. A comprehensive overview of the literature dealing with the modeling and approximation of small-scale dependent shock solutions to nonlinear hyperbolic systems is given in [2]. For these small-scale dependent shock solutions a standard entropy condition is insufficient to single out the physically consistent solution. In fact, aforementioned standard numerical approaches are inadequate for the solution of such systems [2,9]. Hou and LeFloch [10] and Hayes and LeFloch [11] explained this failure in terms of the equivalent equation (also referred to as modified differential equation) of aforementioned schemes. As the leading-order truncation error terms need not match the small-scale mechanisms of these regularized hyperbolic equations, convergence towards the physically consistent solution is not guaranteed. An additional admissibility criterium has to be prescribed. Note that the importance of the equivalent equation or the truncation error, respectively, of a numerical scheme has also been emphasized in implicit turbulence modeling [12].

In this study, we focus on nonclassical entropy solutions to nonlinear hyperbolic systems with diffusive-dispersive regularization. The linear diffusion-dispersion model (also referred to as modified Korteweg-deVries-Burgers equation) was first studied by Jacobs, McKinney and Shearer [13], and later by Hayes and LeFloch [14,15] and Bedjaoui and LeFloch [16]. More involved systems include models of thin liquid films [17–20], the generalized Camassa-Holm model [21,22,20], and models of viscous-capillary materials [23], to name a few. The interested reader will find a thorough overview on systems with undercompressive shocks in [2]. In all of the above systems, nonclassical shock waves, i.e. shock waves which violate the Lax entropy condition [1,3], may arise [1,2]. In [14], the authors introduced a kinetic relation as an admissibility condition for selecting the correct nonclassical weak solution and assessed the capabilities of numerical schemes to produce nonclassical shocks. In [15], the same authors investigated the capability of conservative finite difference schemes to approximate nonclassical shocks by analyzing equivalent equations and determining the corresponding kinetic functions numerically. Following these preliminary works, LeFloch and coworkers designed a family of high-order linear finite difference schemes - so called schemes with controlled dissipation - whose equivalent equations match the regularizing equations to leading order [20]. Schemes with controlled dissipation explicitly discretize diffusive and dispersive regularization terms. Recognizing that the approximation quality of schemes with controlled dissipation deteriorates with increasing shock strength, an extension of said schemes to arbitrary shock strengths was proposed by Ernest et al. [9]. These so called schemes with well-controlled dissipation (WCD) balance the higher order terms in the equivalent equation with the small-scale dependent terms of the underlying physical model. Beljadid et al. have recently applied WCD schemes to hyperbolic systems in nonconservative form [24]. Boutin et al. combined finite difference schemes with an interface tracking to ensure sharp interfaces and essentially exact calculation of nonclassical shocks [25].

In recent years, machine learning has been successfully applied to various problems in fluid mechanics. In [26], Brunton et al. discuss applications of machine learning in the field of fluid mechanics, especially in flow modeling and optimization. Duraisamy et al. [27] review data-driven techniques for turbulence modeling. Significant emphasis has been placed on the identification of underlying physical systems, so called hidden physics, and the recovery of unknown dynamics [28] from data. Raissi, Karniadakis and collaborators introduced the idea of physics-informed learning machines, i.e. learning machines that adhere to any given law of physics. Numerical Gaussian Processes [29] use the time-discretized PDE as covariance functions while Physics-Informed Neural Networks (PINNs) [30] explicitly include the PDE into their objective function. Making use of sparse regression algorithms, Brunton and coworkers introduced a framework for the identification of ordinary [31] and partial differential equations [32]. Sparse identification has recently been applied to the identification of truncation error terms [33]. Related to this, PDE-Net [34] identifies underlying hidden PDE models by learning convolutional kernels from data. At the same time, PDE-Net is capable to make long-time predictions of the dynamics of the learned system. A series of work has investigated the approximation of flow dynamics in a lower dimensional space [35]. Machine learning has also been successfully applied in designing numerical methods. Related to this work, the data-driven discretization by Bar-Sinai et al. [36] learns optimal spatial discretizations to PDEs from fine-resolved simulations. Stevens et al. [37,38] have trained neural networks to perturb the original smoothness measures of WENO5-JS to find an improved and equation independent discretization scheme. In [39], the authors applied neural networks as Riemann solvers in a finite volume scheme to the cubic flux function.

Recognizing that schemes with controlled as well as schemes with well-controlled dissipation show strong dispersive oscillations around shock discontinuities and struggle with sharp shock capturing, and building upon recent advances in machine learning, we develop in this paper a data-driven physics-informed finite-volume method for undercompressive shocks. In particular, we study the cubic scalar conservation law as a prototype for a nonlinear hyperbolic conservation law with non-convex flux. Solutions of this equation allow for nonclassical shocks. We train a neural network to find an optimal interface reconstruction. The neural network outputs a probability distribution over a fixed set of WENO reconstruction stencils and locally introduces a dispersive regularization term. This allows us to match the underlying small-scale mechanisms by a combination of nonlinear truncation error terms. Our scheme is able to sharply resolve shock discontinuities without introducing spurious oscillations, generalizes well to configurations with unseen data, and delivers excellent results for a wide range of shock strengths. Especially, it also approximates nonclassical shocks from unseen smooth initial conditions and shock interactions very well.

The remainder of this paper is organized as follows. In the following section, we consider the scalar cubic conservation law with linear diffusion and dispersion. In section 3, we discuss the proposed data-driven physics-informed finite-volume scheme for undercompressive shocks. In section 4, we explain the network architecture and training routine in detail. In section 5, we show results for standard Riemann problems with nonclassical shocks. We analyze the convergence properties of said neural network scheme and test it on configurations with smooth initial conditions and combinations of Riemann problems. Finally, section 6 summarizes the work and gives concluding remarks.

2. Cubic scalar conservation law

We consider the scalar hyperbolic conservation law

$$u_t + f(u)_x = 0, \quad u = u(t,x) \in \mathbb{R}, t \geq 0. \tag{1}$$

Equation (1) is supplemented by initial and boundary conditions. Additionally, an entropy inequality of the form

$$\eta(u)_t + \psi(u)_x \leq 0 \tag{2}$$

is provided, where (η, ψ) is any strictly convex entropy pair. $\psi(u)$ is determined by $\psi'(u) = \eta'(u) f'(u)$, where the prime denotes the functional derivative.

Solutions to Eq. (1) may contain discontinuities even if initial conditions are smooth. Hence, solutions are considered in the sense of distributions. It is well known from the theory of nonclassical solutions [1], that Eq. (1) may generate nonclassical behaviour when f is a non-convex flux function. Nonclassical shocks violate standard entropy inequalities. In contrast to classical shocks, on which characteristics converge, characteristics pass through nonclassical shocks. That is,

$$s_{NC} < \lambda(u_{-,+}), \tag{3}$$

where s_{NC} is the propagation velocity of a nonclassical shock, $\lambda = f'(u)$ is the speed of the characteristics, and $u_{-,+}$ denote the values left and right of the shock. In that sense, nonclassical shocks are undercompressive.

Undercompressive shocks introduce non-uniqueness. A single entropy inequality like Eq. (2) does not suffice to determine a unique solution. As shown in [1], a nonclassical Riemann problem of the form of Eq. (1) allows for a one-parameter family of solutions. The non-uniqueness can be resolved by supplementing Eq. (1) by regularization terms. In this work, we focus on linear diffusive-dispersive regularizations of the hyperbolic conservation law in (1),

$$u^\epsilon_t + f(u^\epsilon)_x = \epsilon u^\epsilon_{xx} + \delta \epsilon^2 u^\epsilon_{xxx}, \tag{4}$$

where $\delta \in \mathbb{R}$ is a parameter that gives the ratio of the dispersive to the diffusive terms. Thus, the diffusive and dispersive terms are balanced. ϵ is a small-scale parameter. Taking the limit

$$u := \lim_{\epsilon \to 0} u^\epsilon, \tag{5}$$

generates a unique solution u of the underlying hyperbolic conservation law (1). Note that these solutions explicitly depend on the dynamics of the small-scales, i.e. the coefficient δ.

The same small-scale dependent solution can also be attained by providing an additional algebraic condition to the hyperbolic conservation law (1). The so called kinetic function φ^b is imposed at each nonclassical shock and connects the states left u_- and right u_+ of the nonclassical shock,

$$u_+ = \varphi^b\left(u_-\right). \tag{6}$$

φ^{-b} denotes the inverse of the kinetic function, i.e. $u_- = \varphi^{-b}(u_+)$. Naturally, each kinetic function is directly associated with the specific diffusive-dispersive model (4).

For the remainder of this work, we define the flux function as $f(u) = u^3$. The resulting scalar cubic conservation law is a prototype for nongenuinely nonlinear equations. Equation (1) becomes

$$u_t + (u^3)_x = 0, \tag{7}$$

and the regularized system reads

$$u^\epsilon_t + ((u^\epsilon)^3)_x = \epsilon u^\epsilon_{xx} + \delta \epsilon^2 u^\epsilon_{xxx}. \tag{8}$$

The corresponding kinetic function and nonclassical Riemann solver are presented in Appendix A.

3. Numerical method

The finite volume discretization of Eq. (1) corresponds to a convolution with a top-hat filter.

$$\frac{d\bar{u}_i}{dt} = -\frac{F(u_{i+1/2}) - F(u_{i-1/2})}{\Delta x_i}, \tag{9}$$

where $\bar{u}_i$ denotes the cell average value at the cell center x_i, $u_{i\pm1/2}$ are the unfiltered solution at the cell faces $x_{i\pm1/2}$, Δx_i is the width of cell i, and $F(u)$ is the physical flux in the (semi-)discrete setting, the cell index $i = 1, \cdots, n_x$. The cell face values of the unfiltered solution $u(x)$ have to be approximated by $\tilde{u}(x)$. Such approximations to the unfiltered solution can be achieved by primitive-function reconstruction [5]. Essentially non-oscillatory (ENO) and weighted essentially non-oscillatory (WENO) schemes are prominent examples of nonlinear approximation methods.

Since the classical methods are restricted to stencils of fixed size, we apply an extension of the ENO/WENO methodology introduced by Adams et al. [12]. That is, we allow a set of interpolation polynomials up to a given cumulative order K and construct a combination to approximate the cell face value according to

$$\tilde{u}^{\mp}_{i\pm1/2} = \sum_{k=1}^{K}\sum_{r=0}^{k-1} w^{\mp}_{k,r} p^{\mp}_{k,r}(x_{i\pm1/2}). \tag{10}$$

$w_{k,r}$ is the non-negative weight for the k-th order interpolation polynomial with left shift r, i.e. $p_{k,r}$. The left-face interpolant $p^{-}_{k,r}$ and the right-face interpolant $p^{+}_{k,r}$ are given by

$$p^{-}_{k,r}(x_{i+1/2}) = \sum_{l=0}^{k-1} c^{(k)}_{r,l}(i)\bar{u}_{i-r+l}, \quad p^{+}_{k,r}(x_{i-1/2}) = \sum_{l=0}^{k-1} c^{(k)}_{r-1,l}(i)\bar{u}_{i-r+l}. \tag{11}$$

The coefficients $c^{(k)}_{r,l}(i)$ are computed according to Eq. (9) of [12]

$$c^{(k)}_{r,l}(i) = \Delta x_{i-r+l} \sum_{\mu=l+1}^{k} \frac{\sum\limits_{\substack{p=0\\ p\neq\mu}}^{k} \sum\limits_{\substack{\nu=0\\ \nu\neq\mu,p}}^{k} x_{i+1/2} - x_{i-r+\nu-1/2}}{\prod\limits_{\substack{\nu=0\\ \nu\neq\mu}}^{k} x_{i-r+\mu-1/2} - x_{i-r+\nu-1/2}} \tag{12}$$

The derivation of the computation of the coefficients as well as a simplification for equally spaced meshes can be found in [40]. The sum of all weights $w^{\pm}_{k,r}$ over k and r has to be unity,

$$\sum_{k=1}^{K}\sum_{r=0}^{k-1} w^{\pm}_{k,r} = 1.$$

In general, the computation of the weights is based on a heuristic smoothness measure, e.g. the WENO smoothness measure [6] or the total variation within the considered stencil [12]. In the present work, the weights are outputs of the neural network which inherently provides a discrete smoothness measure.

Finally, to complete the numerical scheme, we have to introduce a numerical flux $\tilde{F}_N$ which approximates the physical flux F. We choose the numerical flux as

$$\tilde{F}_{N,i+1/2} = F(\tilde{u}^{-}_{i+1/2}) - \tilde{\delta}_{i+1/2}(c\Delta x_i)^2 u_{xx}(x_{i+1/2}) \tag{13}$$

which consists of the physical flux enhanced by a dispersive regularization term. c is a scaling parameter. $\tilde{\delta}_{i+1/2}$ is the dispersion coefficient at the cell face $x_{i+1/2}$ and is an output of a neural network. In general, a numerical flux function is a two argument function of left and right state reconstruction, i.e. $\tilde{F}_{N,i+1/2} = \tilde{F}_N(\tilde{u}^{-}_{i+1/2}, \tilde{u}^{+}_{i+1/2})$. Due to the fact that all characteristics for the cubic scalar conservation law are right-travelling waves (see Sec. 2), we are only interested in the downstream reconstruction polynomials $p^{-}_{k,r}(x_{i+1/2})$ of cell i. The numerical flux function reduces to $\tilde{F}_N(\tilde{u}^{-}_{i+1/2})$. This corresponds to an upwind numerical flux. Note that the reconstructed state $\tilde{u}^{-}_{i+1/2}$ itself may have upwind-biased, central, and down-biased contributions according to the local WENO coefficients. In Eq. (13), $u_{xx,i+1/2}$ is the second derivative evaluated at $x_{i+1/2}$ and is computed by centered finite differences. Assuming an equidistant mesh, $\Delta x_i = \Delta x$, $u_{xx}(x_{i+1/2})$ becomes

$$u_{xx}(x_{i+1/2}) = \frac{\bar{u}_{i-1} - \bar{u}_i - \bar{u}_{i+1} + \bar{u}_{i+2}}{2\Delta x^2}. \tag{14}$$

The numerical flux, Eq. (13), allows us to tune the truncation error of the numerical scheme by adaptive reconstruction and adaptive regularization. We consider reconstruction polynomials up to order $K = 3$. The underlying small-scale mechanisms can be matched by a combination of nonlinear truncation error terms. While a first-order reconstruction ($w^-_{1,0}$) introduces dissipation, second-order contributions ($w^-_{2,0}$ and $w^-_{2,1}$) manage the amount of dispersion. The third order reconstruction introduces hyper-dissipation. During numerical experimentation, we found that the amount of numerical dispersion introduced by second-order reconstruction was insufficient to balance the numerical dissipation in such a way that undercompressive shocks could be approximated, thus motivating the use of an explicit dispersive regularization term in the flux function (13). We expect our scheme - upon proper training - to be able to ensure a discrete diffusion-dispersion balance around nonclassical shocks, delivering physically consistent approximations of nonclassical shocks.

4. Machine learning

4.1. Convolutional neural networks

Neural networks are parameterizable nonlinear compound functions that map any input x to an output $y = f(x, \theta)$, where θ are free and learnable parameters. Deep neural networks (DNNs) consist of multiple layers of units (so called neurons) in between in- and output. They perform successive elementary nonlinear transformations to map x to y. The numerical values in each layer are called activations.

In multilayer perceptrons (MLPs), neurons in adjacent layers are densely connected. The vector of activations a^l in layer l is computed from the activations of the previous layer a^{l-1} by first applying an affine linear transformation, followed by an element-wise nonlinearity $\sigma(\cdot)$. The activation of the i-th neuron in layer l denoted as a^l_i is calculated as

$$a^l_i = \sigma\left(\sum_{j=1}^{N^{l-1}} W^{l-1}_{i,j} a^{l-1}_j + b^{l-1}_i\right), \tag{15}$$

where $W^{l-1}_{i,j}$ indicates the weight matrix linking layers $(l-1)$ and l, b^{l-1}_i is the bias vector, and N^{l-1} indicates the number of neurons in layer $(l-1)$. The error between the network prediction y and the true output $\hat{y}$ is calculated by a suitable loss function $C(y, \hat{y})$. Training a neural network means finding a set of parameters θ that approximately minimizes said loss function. Typically, the loss function is minimized via mini-batch gradient descent or the popular Adam optimizer [41].

Convolutional neural networks (CNNs) are a special type of deep neural network for structured input data. CNNs exploit sparsity and weight sharing to increase learning efficiency. First, neuron activations are computed only by local sparse interactions with neurons from the previous layer. The local weight matrix is called the filter kernel. Second, CNNs make use of parameter sharing, i.e. the same weight matrix is used for each neuron in a given layer, and, therefore, are translational invariant. Each CNN layer is typically made out of multiple feature maps, each of which is computed by a corresponding filter. The activation of the i-th neuron in the j-th feature map in layer l is denoted as $a^l_{i,j}$ and is calculated as

$$a^l_{i,j} = \sigma\left(\sum_{m=1}^{n^{l-1}_W} \sum_{n=1}^{n^{l-1}_C} W^{l-1}_{m,n,j} a^{l-1}_{i+m-1,n} + b^{l-1}_{i,j}\right). \tag{16}$$

$W^{l-1}_{m,n,j}$ is the filter kernel connecting the n-th feature map in layer $(l-1)$ to the j-th feature map in layer l. n^{l-1}_W is the filter width of $W^{l-1}_{m,n,j}$, n^{l-1}_C is the number of feature maps in layer $(l-1)$.

4.2. Network architecture

In the design of the neural network scheme, we make use of the methods of lines idea. Equation (9) is the semi-discrete approximation of the underlying hyperbolic PDE. For the integration of Eq. (9), cell interface fluxes have to be calculated. We use an artificial neural network for the cell interface flux calculation. Fig. 1 shows the integration of the neural network into the flow solver. The detailed interaction of the network and the flow field is shown in Fig. 2.

The network takes as input the cell-averaged solution at a given time step $\bar{u}^n$. The CNN has two outputs: It gives reconstruction weights $w^-_{k,r}$ for each reconstruction polynomial and the dispersion coefficient $\tilde{\delta}_{i+1/2}$. With these quantities, we can calculate the cell interface fluxes via Eq. (13) and finally the temporal derivatives $d\bar{u}_i/dt$ for each cell average. Depending on the time integration scheme, multiple substeps have to be calculated before an integration step can be performed. Each substep requires an evaluation of the CNN. Fig. 1 depicts the training framework for the forward Euler time integration (Runge-Kutta 1). During training, the network parameters are optimized over multiple time integration steps. This process can be seen as the optimization of a recursive process. Conceptually, this approach is similar to Bar-Sinai et al. [36]. However, we further restrict the neural network to weighting given reconstruction polynomials and a regularization term. Using WENO polynomials compared to a more liberal reconstruction, e.g. the pseudo-linear filter in [36], has several

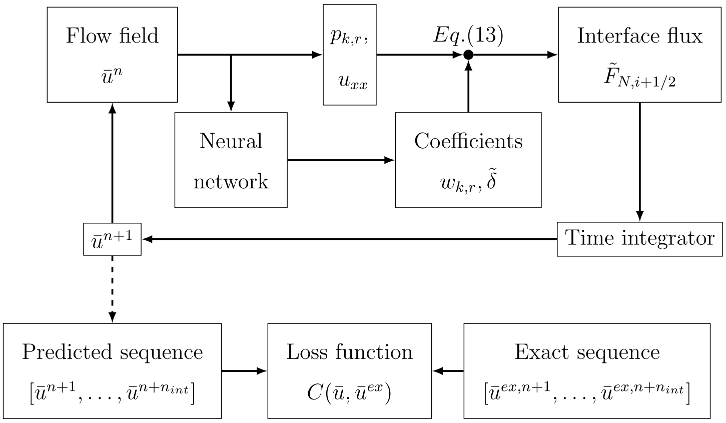

Fig. 1. Interaction between neural network and flow solver. The neural network is integrated in the flow solver. The WENO weights $w_{k,r}$ and the dispersion coefficient $\tilde{\delta}$ predicted by the network are used for the cell interface flux calculation. During training, updated flow states $\bar{u}^{n+1}$ up to $\bar{u}^{n+n_{int}}$ are recorded for the loss calculation. The finite volume solver has to be differentiated using automatic differentiation to compute gradients w.r.t. network parameters. Here, the training routine is depicted for a forward Euler integration scheme. For a general Runge-Kutta time integrator, multiple substeps have to be calculated before an updated flow state is recorded.

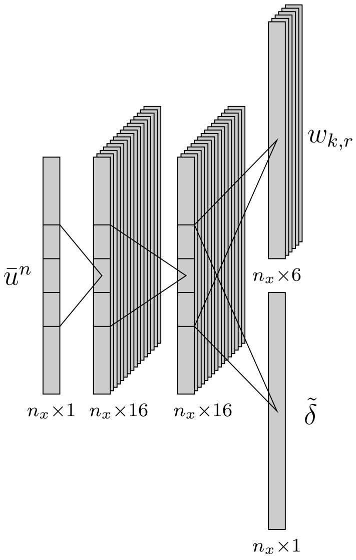

Fig. 2. Detailed view of the neural network architecture. The flow field $\bar{u}^n$ is the input to the network. The input is passed through 1-D convolutional layers with ReLU activation functions before reaching the output. The output consists of two parts: the WENO weights $w_{k,r}$ and the dispersion coefficient $\tilde{\delta}$. The softmax activation is used for $w_{k,r}$ to ensure convexity of the WENO weights.

advantages: (i) we benefit from the theoretic background of a well-established numerical method; (ii) the machine learning intrusion into the numerical scheme is kept to a minimum; (iii) it provides a physics-informed scheme which adheres to any given law of physics, and provides comprehensible and interpretable results.

In this work, we use a three layer convolutional neural network (CNN) [42] with 16 kernels in each hidden layer. The kernel size is 3 and the stride is 1. The total number of trainable parameters is 1191. CNNs are a natural choice in the design of numerical schemes, as they provide sparse interactions and parameter sharing [42]. This enables us to train the neural network scheme, at a certain resolution and then, at test time, use the same network parameters for predictions at different resolutions. We add mild l_2 weight regularization with a regularization factor of 1×10^{-5} to prevent overfitting. In the hidden layers rectified linear units (ReLU) defined as $ReLU(x) = max(0, x)$ are used. The output layer for the reconstruction weights uses a softmax activation function. The softmax activation can be interpreted as a discrete probability distribution over the different stencil types. Naturally, this ensures that the reconstruction stays consistent as $\sum w_{k,r} = 1$ and $0 \leq w_{k,r} \leq 1 \ \forall \ r, k$. In the output layer for the $\tilde{\delta}$-coefficient a ReLU activation is used, since we want to enforce the physical constraint $\tilde{\delta}_{i+1/2} \geq 0$.

The loss functional is defined as the absolute difference between predicted and analytical cell-averaged values over n_{int} time steps

$$C = \sum_{j=1}^{n_{int}} \sum_{i=1}^{n_x} |\bar{u}^{ex}(x_i, t_j) - \bar{u}(x_i, t_j)|. \tag{17}$$

During numerical experimentation, we found that including successive predictions into the loss function greatly increases stability of the generated numerical scheme.

Our model was implemented with the TensorFlow library [43]. The model is trained with the Adam optimizer [41] on one Nvidia GTX 2080 GPU with a mini-batch size of 128. Weights are initialized according to the Glorot normal distribution [44]. We have trained two models for weak nonclassical shocks, one model for moderate and one for strong nonclassical shocks. For the weak shock models, we start with a learning rate of $\eta = 10^{-2}$, divide it by 10 after 100 and 200 iterations, and terminate the training process at 300 iterations. For the moderate and strong model, we start with a learning rate of $\eta = 10^{-3}$, divide it by 3 after 200 and 400 iterations, and terminate the training process at 600 iterations.

4.3. Time integration

Throughout this work, we use the classical fourth-order Runge Kutta scheme (RK4).

4.4. Training data

In a supervised learning task, we have to provide features and labels to train the network. In this subsection, we will explain the reasoning behind the training data generation.

Weak shocks have a shock strength less than 10, moderate shocks have shock strengths between 10 and 100. The maximum shock strength for strong shocks is greater than 100. We generate four distinct datasets. For weak, moderate, and strong shocks we generate one dataset each where we are only interested in Riemann problems with $u_L > 0$. The corresponding NN schemes are referred to as weak, moderate, or strong model. In section 5.5 however, we test the NN scheme on a smooth initial condition which not only develops nonclassical shocks from $u_- > 0$ to $u_+ < 0$ but also $u_- < 0$ to $u_+ > 0$. To correctly approximate such a test case, we generate another training dataset with weak shocks in which we also include Riemann problems with $u_L < 0$. We denote this model as the weak$^\pm$ model.

The training data is generated from exact solutions to Riemann problems which feature a nonclassical shock/classical shock (NCC) or nonclassical shock/rarefaction (NCR) wave structure. We use u_L and u_R to denote the constant states left and right of the initial discontinuity in the Riemann problem. u_M denotes the state right of the nonclassical shock. The exact Riemann solver for the cubic scalar conservation law can be found in the Appendix A. Appendix B provides the specific Riemann problems used in each training dataset. Generally, we choose Riemann problems equidistantly spaced in the Riemann problem space. For moderate and strong shocks, we add additional Riemann problems with larger values of u_R to the training sets. Such configurations are particularly challenging due to the similarity of shock speeds of nonclassical and classical shocks. For each Riemann problem, we sample snapshots of the exact solution in time between $t_0 = 0$ and $t_{final} = \frac{L - x_0}{3max(u_L^2, u_R^2)}$. L is the domain length and x_0 is the position of the initial jump of the Riemann problem. For the training sets we have chosen the domain $[0, 2]$, i.e. $L = 2$, and $x_0 = 0.4$. t_{final} is chosen such that all shocks remain within the computational domain.

Fig. 3 shows several training examples. The top row of Fig. 3 depicts training snapshots of the model for weak shocks. The first and third configuration display configurations with nonclassical shock and rarefaction wave, while the other configurations are nonclassical shock/classical shock configurations. The bottom row of Fig. 3 shows snapshots from the training set of the weak$^\pm$ model. This training set includes configurations with $u_L < 0$ and $u_R > 0$, e.g. the third, fourth, and fifth plot in the bottom row.

We use these solutions as starting points for the integration and advance them in time by making use of the neural network prediction. That is, to each initial distribution at time t^n corresponds a sequence of exact solutions at n_{int} successive time steps, i.e. at $t^n + \Delta t, t^n + 2\Delta t, ..., t^n + n_{int}\Delta t$. Δt is determined by a suitable CFL-condition. We use $CFL = 0.6$ in this work. Throughout this work we optimize the network based on $n_{int} = 7$ integration steps.

During numerical experimentation, it was found that it is crucial to provide enough training samples until the intermediate state (between nonclassical and classical shock or between nonclassical shock and rarefaction wave) is fully developed. The information gain for the network training process from samples with fully resolved two-shock or shock-rarefaction structure is marginal. We introduce a characteristic time scale $t_{separation}$ for the separation between the nonclassical shock and the classical shock and the nonclassical shock and the rarefaction wave, respectively,

$$t_{separation} = \frac{\Delta x}{\Delta v}, \tag{18}$$

where Δv is the speed with which the two shock waves separate. For NCC configurations

$$\Delta v = s_C - s_{NC} = \frac{u_M^3 - u_R^3}{u_M - u_R} - \frac{u_L^3 - u_M^3}{u_L - u_M}, \tag{19}$$

while for NCR configurations

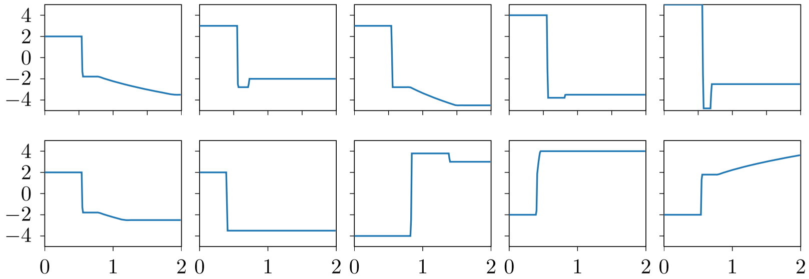

Fig. 3. Samples from the training dataset. Top row: training samples for the model for weak shocks. Bottom row: The training data for the weak$^{\pm}$ model is augmented by samples with $u_L < 0$.

Table 1
Number of samples for each model. c is the scaling used in Eq. (13).

Model	Δx	c	Training	Validation	Testing
Weak	1×10^{-2}	5	2800	900	100
Moderate	1×10^{-3}	5	7200	900	100
Strong	1×10^{-3}	10	3600	1000	100
Weak$^{\pm}$	1×10^{-2}	5	5600	1800	100

Table 2
Overview on model performance. The numbers in the brackets denote the standard deviation over the different initial conditions.

Model	Mean Absolute Error		Mean Absolute Percentage Error	
	Training	Validation	Testing	Testing Ref.
Weak	7.816×10^{-4}	7.548×10^{-4}	0.67 (0.77)	4.86 (2.12)
Moderate	1.077×10^{-3}	1.138×10^{-3}	0.48 (0.80)	52.19 (37.37)
Strong	2.123×10^{-3}	1.964×10^{-3}	1.68 (0.59)	179.85 (49.42)
Weak$^{\pm}$	8.223×10^{-4}	7.993×10^{-4}	0.79 (2.03)	4.89 (2.70)

$$\Delta v = s_R - s_{NC} = 3u_L^2 - \frac{u_L^3 - u_M^3}{u_L - u_M}. \tag{20}$$

s_C is the speed of the classical shock, s_{NC} is the speed of the nonclassical shock, and s_R is the speed of the rarefaction wave. $u_M = -u_L + \frac{\sqrt{2}}{3\sqrt{\delta}}$ is the value of the intermediate state. We sample up to 150 snapshots for each Riemann problem. 50 snapshots are uniformly sampled from $t_0 \leq t < t_{separation}$, the rest of the configurations is uniformly sampled from $t_{separation} \leq t \leq t_{final}$. Note that if the two shocks separate very quickly, such that it takes less than 50 time steps to reach the critical separation, all time steps until separation are sampled and the remaining samples are simply taken from $t_{separation} \leq t \leq t_{final}$.

The validation set is generated analogously. For the test set, we choose random initial conditions from a uniform distribution over the corresponding Riemann problem space. These initial conditions are then integrated until the final time t_{final}. We train one NN scheme on each dataset and thus obtain four models (weak, moderate, strong, and weak$^{\pm}$ model). Table 1 provides an overview over the four trained models. Table 2 shows an overview over the performance metrics of the different models. Wall-clock times are given in Appendix D. Training and validation error are calculated over n_{int} successive integration steps, while for the testing error we simulate the testing configurations until t_{final} - possibly over thousands of integration steps - and calculate the mean absolute percentage error (MAPE) compared to the exact solution at t_{final}. To provide a reference metric for the testing error, we use an 8-th-order WCD scheme by LeFloch [9]. See Sec. 5 for details of the WCD scheme. Note that the MAPE of the WCD scheme increases significantly with the shock strength due to the dispersive oscillations while the order of the MAPE for the NN scheme stays roughly constant.

5. Results

In this section, we test the neural network scheme (NN scheme) on a series of relevant Riemann problems unseen by the network during training, either in terms of extended time range or in terms of an entirely different setup. We are interested in Riemann problems which develop weak, moderate, and strong nonclassical shocks. The Riemann initial data are

$$u(x, t = 0) = \begin{cases} u_L & \text{for } x \leq 0.4, \\ u_R & \text{for } x > 0.4. \end{cases} \tag{21}$$

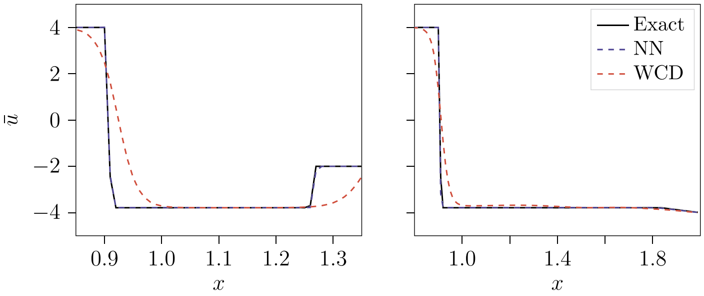

Fig. 4. Comparison of the NN scheme solution (13) with an 8-th-order WCD scheme. Solutions are shown at $t = 3.333 \times 10^{-2}$. Left: Solution of the Riemann problem for $u_L = 4$ and $u_R = -2$. Right: Solution of the Riemann problem for $u_L = 4$ and $u_R = -5$.

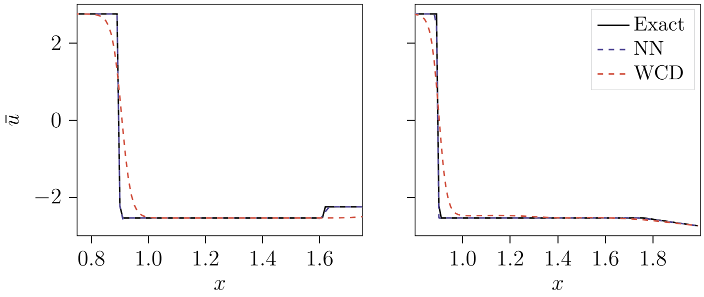

Fig. 5. Comparison of the NN scheme solution (13) with an 8-th-order WCD scheme. Solutions are shown at $t = 7.052 \times 10^{-2}$. Left: Solution of the Riemann problem for $u_L = 2.75$ and $u_R = -2.25$. Right: Solution of the Riemann problem for $u_L = 2.75$ and $u_R = -3.25$.

For most of the aforementioned test cases, we will set $u_R = -2$. By varying u_L, we determine the resulting shock strengths and solution structure. The domain of interest is $[0, 2]$. For all Riemann problems, we use transmissive boundaries. The initial distribution is integrated until $T_{final} = (L - x_0)/(3u_L^2)$, where $L = 2$ is the domain length and $x_0 = 0.4$ is the position of the initial jump. For definiteness, we use the physical diffusion-dispersion ratio $\delta = 5$. The CFL-number used for the NN scheme is 0.6.

Throughout this section we will use the scheme with well-controlled dissipation (WCD) [9] as a reference. We have adapted an 8-th-order WCD scheme to the finite volume formulation. In the WCD scheme, we have to choose the parameter τ which determines the ratio of high-order terms in the modified equation to leading-order diffusion and dispersion terms. We set $\tau = 0.04$ in all our test cases. The CFL-number used for the WCD scheme is 0.4.

Having tested our scheme on Riemann problems, we analyze the convergence. Finally, we are interested whether the NN scheme is able to predict the formation of nonclassical shocks from smooth initial conditions and how shock interactions are handled. Additional test cases are provided in Appendix E.

5.1. Weak shocks

We consider two standard Riemann problems from literature with $(u_L, u_R) = (4, -2)$ and $(u_L, u_R) = (4, -5)$. The former one has a nonclassical-classical structure while the latter one develops a nonclassical-rarefaction structure. The domain is resolved by 200 points which corresponds to $\Delta x = 1 \times 10^{-2}$. Fig. 4 compares the solution of the neural network scheme (13) with the finite volume scheme with well-controlled dissipation. The solution of our NN scheme is visually almost indistinguishable from the exact solution. The proposed scheme has very good shock capturing capabilities. For $u_R = -2$, both shock discontinuities appear sharp. Especially, the nonclassical shock is well approximated. For $u_R = -5$, the NN scheme is capable to approximate the nonclassical shock and the rarefaction wave. Since the dispersion is applied in a locally adaptive fashion, we can suppress dispersive oscillations around the leading shock which are present in the finite volume scheme with well-controlled dissipation. The shock position and the intermediate state are approximated well.

Although the results in Fig. 4 were integrated from initial conditions present in the training set, we want to emphasize that during training only a relatively small number of integration steps were performed. At test time, the proposed scheme provides very good results for integration far beyond the scope of the training process. Fig. 5 provides results for initial conditions which were neither part of the training nor the validation set. The approximation quality of the NN scheme for unseen configurations is very good. Nonclassical as well as classical shock discontinuities are captured sharply. Shock speeds and intermediate states are almost perfectly captured.

We are interested in how the neural network handles the cell face reconstruction. Fig. 6 shows a detailed view of the nonclassical and classical shock from Fig. 4, where $(u_L, u_R) = (4, -2)$. Depicted are the flow state and the outputs of the network, i.e. the dispersion coefficient $\tilde{\delta}$ and the reconstruction weights $w_{k,r}^-$. There are several main findings: Firstly, we

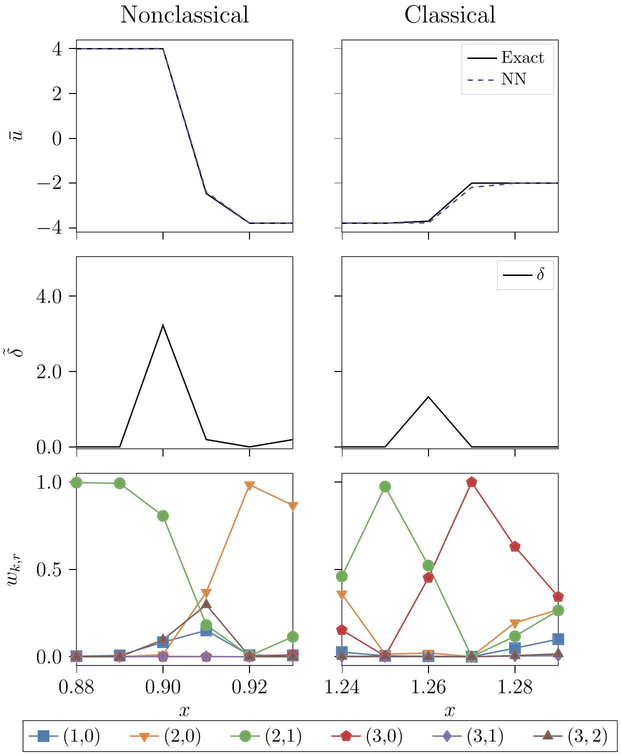

Fig. 6. Close-up of the discontinuities in Fig. 4 (left). Top: $\bar{u}$. Middle: $\tilde{\delta}$. Bottom: Reconstruction weights. $\tilde{\delta}$ and $w_{k,r}$ are direct outputs of the neural network.

observe that the network detects shock discontinuities and introduces dispersion only locally. Note that around both, nonclassical and classical, discontinuities the $\tilde{\delta}$ coefficient shows a peak. The local dispersion is preventing oscillatory behaviour away from the discontinuities. Secondly, the cell face reconstruction shows a systematic behaviour. We see that the network chooses the upwind reconstruction stencil $w_{2,1}$ in flow regions behind the shock, while downwind reconstruction stencils $w_{2,0}$ and $w_{3,0}$ are picked in front of the shock. Note that diffusion is only locally introduced at the nonclassical shock, i.e. $w_{1,0}$. This is in line with the understanding that in order to approximate a nonclassical shock diffusion and dispersion have to be balanced. At the classical shock, the network does not use diffusion $(w_{1,0})$ but rather uses dispersion to stabilize the shock. Slight amounts of diffusion are introduced at locations right of the classical shock. The central stencil with upwind bias $(w_{3,1})$ is not activated by the network in this example. The reconstruction away from discontinuities where the flow state is constant is determined by the last point affected by the shock discontinuity. Points in constant flow regions do not contribute to the loss function.

5.2. Moderate shocks

We consider two different nonclassical shocks of moderate strength and simulate them with the NN scheme for moderate shocks. The first Riemann problem consists of $u_L = 25$ and $u_R = -2$. The resulting exact solution is composed of a leading classical shock and a trailing nonclassical shock of strength around 50. The domain is relatively coarsely resolved with 2000 mesh points which corresponds to a cell width of $\Delta x = 10^{-3}$. Fig. 7 compares our scheme with the 8-th-order WCD scheme. The NN scheme approximates both shock discontinuities very well. The shock position is almost identical with the exact solution. A slight oscillation around the middle state u_M is visible. However, the oscillation decreases upon mesh refinement (see the highly resolved middle state for the next test case in Fig. 8). Compared to the WCD scheme, the NN scheme is able to capture both shocks while introducing very mild spurious oscillations.

Tests on other configurations also showed very satisfactory results. In Fig. 8, a different moderate shock with $u_L = 30$ and $u_R = -2$ is shown at an increased mesh resolution of $n_x = 10000$ points $(\Delta x = 2 \times 10^{-4})$. Intermediate state and shock speeds are approximated well. The spurious oscillations of the middle state are considerably smaller compared to the WCD scheme. Dispersive oscillations at the leading shock are suppressed completely.

5.3. Strong shocks

To simulate a nonclassical shock of strength larger than 100, we choose $u_L = 55$ and $u_R = -2$. The exact solution consists of a leading classical shock of moderate strength and a trailing nonclassical shock of strength around 110. The shock speeds

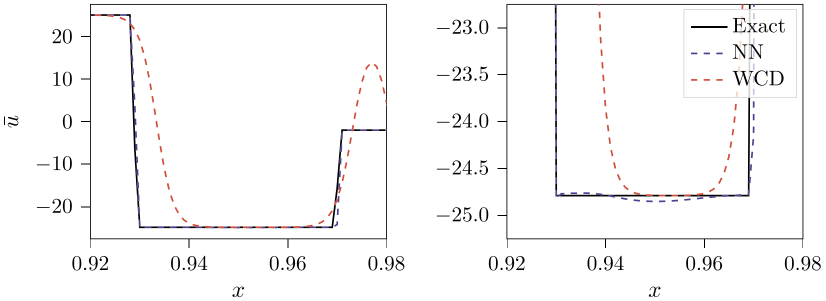

Fig. 7. Moderate shock with $u_L = 25$ and $u_R = -2$ at $t = 8.533 \times 10^{-4}$. Left: Comparison of the Neural Network approximation with an 8-th-order WCD scheme. Right: Detailed view of the middle state u_M.

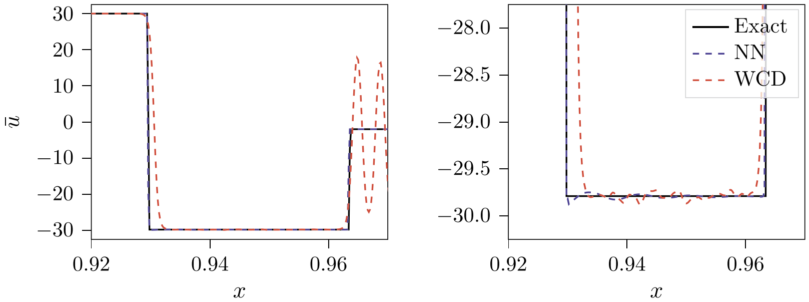

Fig. 8. Moderate shock with $u_L = 30$ and $u_R = -2$ at $t = 5.926 \times 10^{-4}$. Left: Comparison of the Neural Network approximation with an 8-th-order WCD scheme. Right: Detailed view of the middle state u_M.

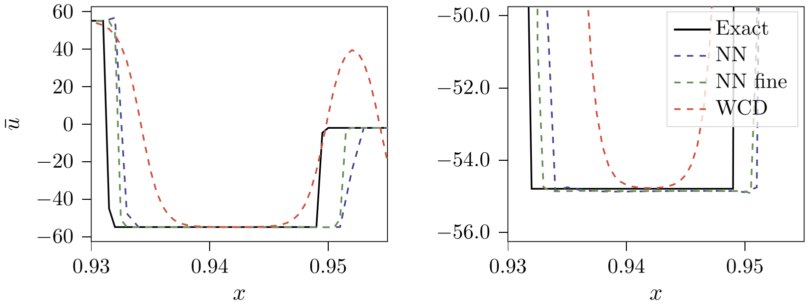

Fig. 9. Large shock with $u_L = 55$ and $u_R = -2$ at $t = 1.763 \times 10^{-4}$. Left: Comparison of the Neural Network approximation with an 8-th-order WCD scheme. Right: Detailed view of the middle state u_M.

are very similar, and resolving the intermediate state becomes more challenging. We compare the NN scheme solution at a resolution of 2000 (NN) and 10000 (NN fine) points to the WCD solution at 10000 points. For the NN scheme, we increase the multiplicative factor of the dispersive term to $c = 10$. Fig. 9 shows that the NN scheme can also handle strong shocks quite well. Compared to the WCD approximation, the NN scheme approximates the intermediate state very well and is able to capture the shocks sharply. We observe a slight offset between the exact shock positions and the shock positions predicted by the NN scheme. When compared to the coarse approximation, nonclassical and classical shock locations on the finer resolution are closer to the exact shock positions. This offset seems to be an artifact of the early development stages of the double shock structure. The difference in shock speeds for the given test case is very small which makes resolving the intermediate state very challenging.

5.4. Convergence

We conduct a mesh convergence study for the aforementioned models for weak, moderate, and strong shocks. The local interaction property of convolutional neural networks allows us to train a model at a given resolution and use the same network parameters for an increased or decreased spatial resolution without having to train a new model. On a sequence of grids, we compute solutions to the nonclassical Riemann problem with double shock structure. We compute the time averaged L^1 error between the approximate finite volume solution $\bar{u}^{n_x}$ at resolution n_x and the exact solution u^{ex} as follows,

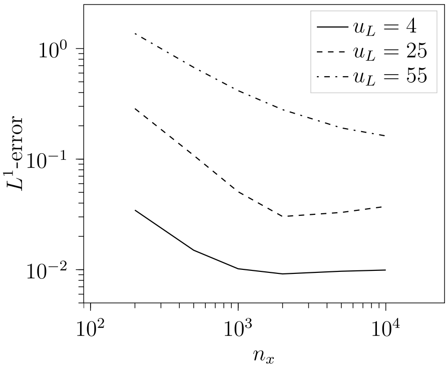

Fig. 10. Convergence in L^1 for the NN scheme across different regimes. Displayed are the L^1 error for weak, moderate, and strong shocks.

$$E := \frac{1}{n_t}\sum_{i=1}^{n_t}\int_{x=0}^{L} |\bar{u}^{n_x}(x, i\Delta t) - u^{ex}(x, i\Delta t)|dx.$$

We set $n_t = 100$ and increase the resolution from $n_x = 200$ to $n_x = 10000$ mesh points. The exact nonclassical solution for the Riemann problem is given in analytical form. Fig. 10 shows the resulting error with respect to the number of grid points. We show convergence for the three different schemes for weak, moderate, and strong shocks, respectively. Generally, we observe a convergence behaviour in the L^1 norm to the exact nonclassical solution as the resolution is increased. The model for weak shocks reaches a plateau at around $n_x = 2000$, after which the L^1 norm does not further decrease. For moderate shocks, we see that the minimal L^1 error is achieved at the training resolution of $n_x = 2000$ points. Further increasing the resolution does not further decrease the L^1 norm. We want to emphasize that we can train a separate model for each resolution. During numerical experimentation, we found that these models generally perform better than a model for which the resolution was adapted after training.

5.5. Smooth initial conditions

Similarly to [25], we consider a smooth initial condition

$$u_0(x) = 4sin(\pi x),$$

on the domain $L = 2$ with periodic boundary conditions. We expect the formation of nonclassical and classical shocks as time progresses. We regard this initial boundary value problem as a challenging test case since the network has neither seen any smooth initial conditions nor the dynamic development of shock discontinuities in the training process. As the exact solution is not known, we use a fine-resolved solution of the finite volume scheme with well-controlled dissipation as a reference solution. The increasing flanks form rarefactions, while the decreasing parts develop nonclassical-classical shock structures. As is known from [25], the sinusoidal initial condition will create nonclassical shocks with $u_- > 0$ and $u_+ < 0$ as well as nonclassical shocks with $u_- < 0$ and $u_+ > 0$. Therefore, we have supplemented the training data for the weak$^\pm$ model with solutions of the Riemann problem with $u_L < 0$, see Fig. B.14. During numerical experimentation, we found that when only nonclassical Riemann problems with $u_L > 0$ were included in the training process, the inferred model was not able to correctly resolve nonclassical shocks from $u_- < 0$ to $u_+ > 0$. Instead, the model would wrongly choose the classical Riemann solution which consists of a classical shock and a rarefaction wave.

Fig. 11 compares the solution of the NN scheme (weak$^\pm$ model) with the reference scheme from LeFloch and coworkers. The left column shows flow state and reconstruction variables at $t = 0.01$ while the right column shows the later time $t = 0.05$. Two WCD schemes are run at a resolution of 200 and 2000 mesh points, respectively. The NN scheme is simulated on the coarser grid of 200 mesh points. First, we note that our scheme is able to develop the discontinuous solution from smooth initial conditions. At $t = 0.01$, we see that the steepened flanks begin to form shock discontinuities. The neural network detects this and adapts reconstruction and local dispersion accordingly. At $t = 0.05$, two NCC structures are fully developed. The WCD scheme on the coarser mesh barely manages to resolve the shock discontinuities. The NN scheme shows sharp shock capturing. The shock strengths and shock positions are captured very well. The shock positions of the NN scheme are in very good agreement with the fine WCD solution, only the leading classical shock is slightly trailing behind. The post shock state (which is not constant any longer) is approximated very well when compared with the fine-resolved WCD solution. The quality of the solution is very satisfactory keeping in mind that during training only Riemann problems were used.

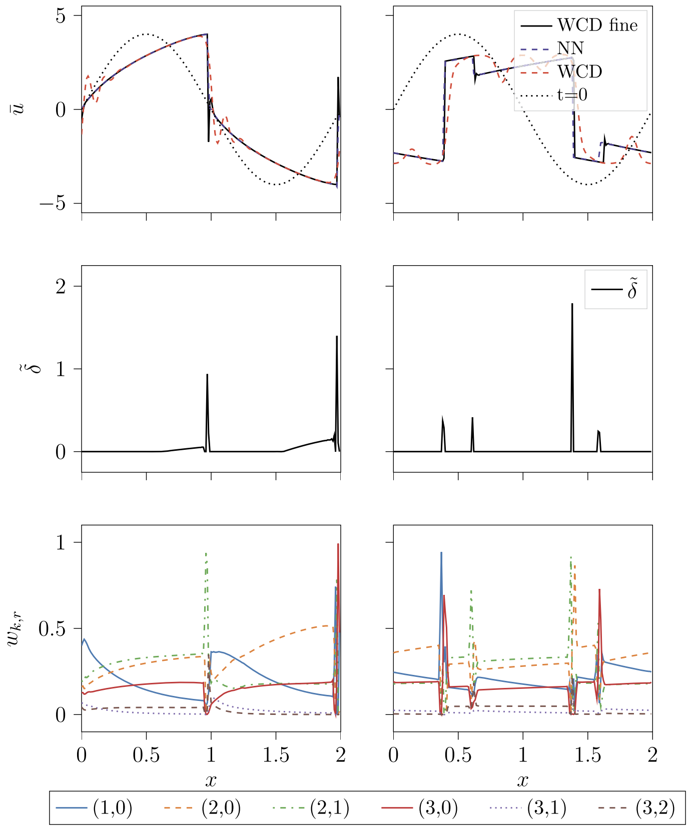

Fig. 11. Periodic initial conditions - comparison of the WCD scheme and NN scheme. Left: Solution at $t = 0.01$. Right: Solution at $t = 0.05$.

5.6. Shock interaction

We assess whether the NN scheme can approximate shock interactions, even though it has not been trained on such data. We test weak shock interactions with the weak$^\pm$ model and strong shock interactions with the strong shock model. For both test cases, we start with three well-separated Riemann problems which will develop shock interactions over time.

We test the weak$^\pm$ model on the domain $[0, 6]$ with a resolution of $\Delta x = 1 \times 10^{-2}$. As a comparison, we use the 8-th-order WCD scheme on $\Delta x = 1 \times 10^{-2}$ and on $\Delta x = 2 \times 10^{-3}$. The three Riemann problems are determined by the following initial condition,

$$u(x, t = 0) = \begin{cases} 4 & \text{for } x < 0.25, \\ -2 & \text{for } 0.25 \geq x < 0.75, \\ 3 & \text{for } 0.75 \geq x < 1.25, \\ -2 & \text{for } 1.25 \geq x. \end{cases} \tag{22}$$

The first Riemann problem results in a fast nonclassical shock/classical shock (NCC), the second Riemann problem generates a nonclassical shock/rarefaction wave (NCR) structure, and the final Riemann problem results in a second NCC. Fig. 12 compares the solution of the NN scheme with the solutions of the WCD scheme. At $t = 1.25 \times 10^{-2}$, the nonclassical solutions to each of the three Riemann problems have developed. At $t = 4.50 \times 10^{-2}$, the leftmost classical shock has interacted with the neighbouring nonclassical shock and the rarefaction wave has run into the rightmost nonclassical shock. The NN scheme approximation agrees well with the WCD solution on the finer mesh. We want to emphasize that the network has not seen such interactions during the training process. The NN scheme manages to approximate small-scale structures when compared to the WCD solution on the same resolution. The bottom row of Fig. 12 shows the solution at later times. The NN scheme shows very good agreement with the fine-resolved WCD result.

For the test case with strong shock interactions, the initial distribution is given by Eq. (23). The computational domain is $[0, 2]$. We choose a resolution of $\Delta x = 1 \times 10^{-3}$. The NN scheme is compared with the 8-th-order WCD scheme on the same resolution. The first Riemann problem results in a fast travelling classical shock (55 to 50), the second Riemann problem

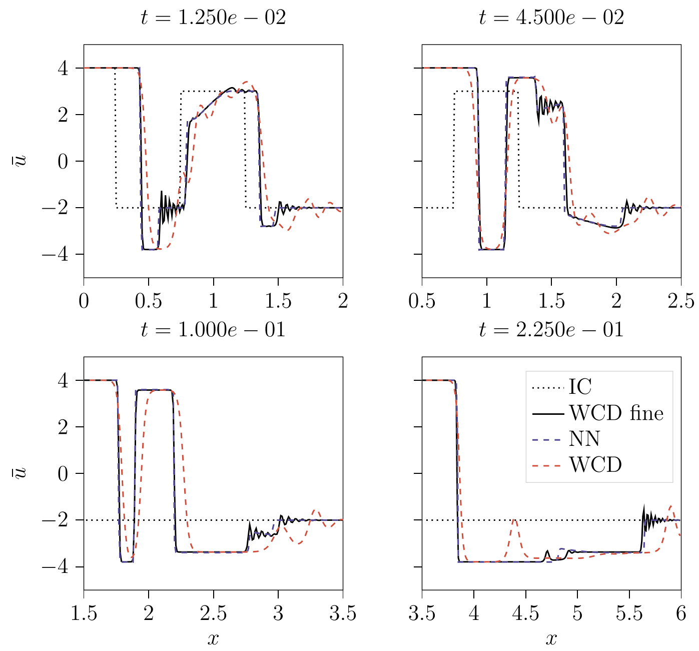

Fig. 12. Weak shock interaction - comparison between the NN scheme and the WCD scheme.

generates a strong nonclassical shock (50 to −50) and a classical shock (−50 to −5), and the final Riemann problem results in a slow travelling classical shock (−5 to 2).

$$u(x, t = 0) = \begin{cases} 55 & \text{for } x < 0.25, \\ 50 & \text{for } 0.25 \geq x < 0.75, \\ -5 & \text{for } 0.75 \geq x < 1.25, \\ 2 & \text{for } 1.25 \geq x. \end{cases} \tag{23}$$

Fig. 13 compares the NN scheme with the WCD scheme at three characteristic points in time. At $t = 5.00 \times 10^{-5}$, the solution structure of each Riemann problem is well developed. At $t = 1.50 \times 10^{-4}$, the fast travelling classical shock has interacted with the strong nonclassical shock. The resulting nonclassical shock from 55 to approximately −55 is well predicted by the NN scheme. The final interaction between both remaining classical shocks is complete at $t = 2.00 \times 10^{-4}$. The predicted NCC structure of the NN scheme is in good agreement with the WCD results.

6. Conclusion

The design of numerical methods for complex physics remains an important topic of research. Solutions to non-convex hyperbolic systems may include nonclassical shocks which depend on underlying small-scale physics. As standard numerical schemes fail to approximate such shock waves correctly, a plethora of work has been devoted to devise numerical schemes which explicitly discretize diffusive and dispersive regularization terms. Machine learning offers a new approach for designing numerical methods. Training data can be used to optimize established methods or come up with new schemes altogether.

In this work, we have put forward a data-driven physics-informed finite-volume scheme for the computation of undercompressive shocks. We have trained a convolutional neural network to combine an optimal cell interface reconstruction based on WENO polynomials with a local regularization term. This allowed us to devise a adaptive scheme which is able to capture nonclassical shocks sharply while suppressing spurious oscillations. The neural network recognizes nonclassical as well as classical shocks and chooses a reconstruction so that diffusion and dispersion are balanced. Our scheme has shown very satisfactory results for the approximation of shock speeds and post shock states for weak, moderate, and strong shocks. The generalizability of the neural network scheme is very good, the intermediate state for unseen Riemann configurations showed very good results. Even smooth initial conditions and shock interactions, which are not part of the training set, are handled very well. Restricting the neural network to the reconstruction task, provides not only a method which adheres to any given physical conservation law but also a scheme which yields interpretable results. We want to emphasize that the reconstruction systematics chosen by the neural network give detailed insight as to how a heuristic smoothness measures or numerical schemes for small-scale dependent shocks might look like. In this way, the scheme also serves as an analysis tool

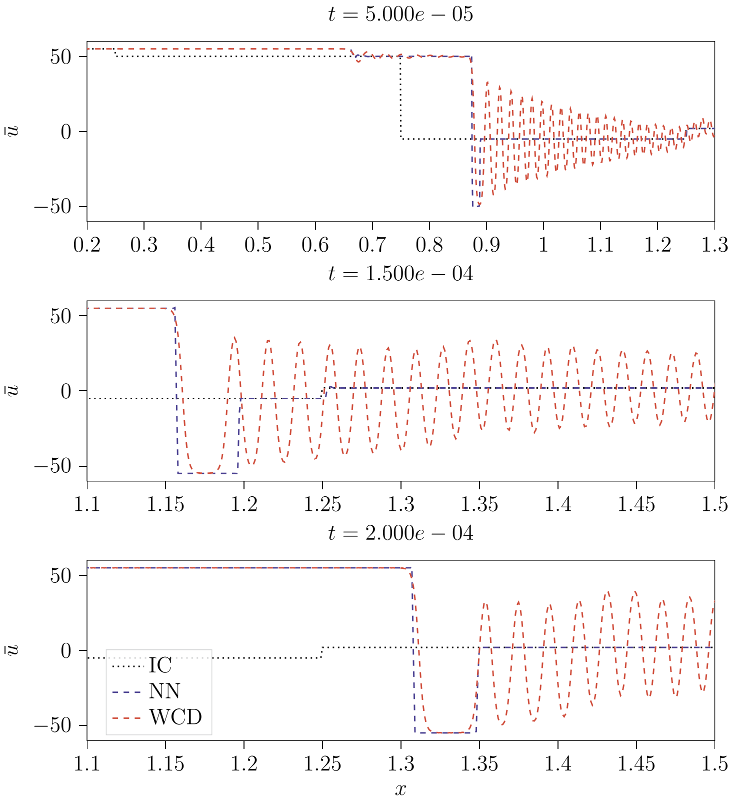

Fig. 13. Strong shock interaction - comparison between the NN scheme and the WCD scheme.

for the future design of numerical schemes. To the knowledge of the authors, this is the first data-driven shock-capturing scheme for undercompressive shocks.

There are points which need further investigation. Although, our scheme has shown good results for smooth distributions, it did not achieve maximum accuracy order in smooth regions. Incorporating smooth training examples might enable the neural network also to learn an optimal reconstruction in smooth flow regions. So far, we have trained different models according to the expected shock strength. A dictionary of models for different shock strengths might be a straightforward way to tackle problems with shocks of various strengths. Making the form of the regularization term adaptive may also lead to increased performance, especially for strong shocks.

The proposed NN scheme is trained with exact solutions of the underlying conservation law. Clearly, a set of exact solutions as rich as the one used in this work can only be generated for 1-D scalar problems. While the multidimensional case might be covered by a straightforward dimension-by-dimension application of the NN scheme, high fidelity training sets have to be assembled for more complicated systems of equations. Smoothed out high resolution WCD results, numerical solutions of interface tracking schemes, experimental data sets, and combinations of the aforementioned might offer the possibility to create high fidelity training sets for such systems. Therefore, an extension of the NN scheme to systems of conservation laws and to multidimensional problems remains a challenge and is the topic of ongoing research.

CRediT authorship contribution statement

Deniz A. Bezgin: Conceptualization, Investigation, Methodology, Software, Writing – original draft. **Steffen J. Schmidt:** Methodology, Supervision, Writing – review & editing. **Nikolaus A. Adams:** Funding acquisition, Project administration, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 667483).

Appendix A. Riemann solver for the scalar cubic flux model

We want to solve the Riemann problem of the cubic flux model

$$u_t + (u^3)_x = 0 \tag{A.1}$$

with initial data

$$u(x,0) = \begin{cases} u_L & \text{for } x < 0, \\ u_R & \text{for } x > 0, \end{cases} \tag{A.2}$$

and $u_L > 0$. We restrict the solutions to the vanishing small-scale limit of the diffusive-dispersive regularization of Eq. (A.1) given by

$$u_t^\epsilon + ((u^\epsilon)^3)_x = \epsilon u_{xx}^\epsilon + \delta \epsilon^2 u_{xxx}^\epsilon. \tag{A.3}$$

The constant δ is the ratio of dispersion to dissipation and describes the small-scale physics. The limiting solution of Eq. (A.3), i.e. $\epsilon \to 0$, is equivalent to the solution of Eq. (A.1) when prescribing the kinetic function $\varphi_\delta^b(u_-)$ at each nonclassical shock,

$$\varphi_\delta^b(u_-) = \begin{cases} -u_- - \frac{\sqrt{2}}{3\sqrt{\delta}} & \text{for } u_- \leq -\frac{2\sqrt{2}}{3\sqrt{\delta}}, \\ -u_-/2 & \text{for } |u_-| \leq \frac{2\sqrt{2}}{3\sqrt{\delta}}, \\ -u_- + \frac{\sqrt{2}}{3\sqrt{\delta}} & \text{for } u_- \geq \frac{2\sqrt{2}}{3\sqrt{\delta}}. \end{cases} \tag{A.4}$$

Then, the Riemann problem has the following solution structure [14]:

(a) For $u_L \leq \frac{2\sqrt{2}}{3\sqrt{\delta}}$ the solution is the classical Riemann solution:
  (i) A rarefaction wave, if $u_R \geq u_L$,
  (ii) a classical shock wave, if $u_R \in [-u_L/2, u_L)$,
  (iii) and a classical shock followed by a rarefaction wave, if $u_R < -\frac{1}{2}u_L$.

(b) For $u_L > \frac{2\sqrt{2}}{3\sqrt{\delta}}$ the solution is the nonclassical Riemann solution:
  (i) A rarefaction wave, if $u_R \geq u_L$,
  (ii) a classical shock wave, if $u_R \in [-\frac{\sqrt{2}}{3\sqrt{\delta}}, u_L)$,
  (iii) a nonclassical shock wave from u_L to $-u_L + \frac{\sqrt{2}}{3\sqrt{\delta}}$, followed by a classical shock wave to u_R, if $u_R \in (-u_L + \frac{\sqrt{2}}{3\sqrt{\delta}}, -\frac{\sqrt{2}}{3\sqrt{\delta}})$,
  (iv) and a nonclassical shock wave from u_L to $-u_L + \frac{\sqrt{2}}{3\sqrt{\delta}}$, followed by a rarefaction wave to u_R, if $u_R \leq -u_L + \frac{\sqrt{2}}{3\sqrt{\delta}}$.

The Riemann solver for $u_L < 0$ is structured analogously.

Appendix B. Training data

Fig. B.14 provides an overview of the training set for weak, moderate, strong, and weak$^\pm$ models. Note that the blank spaces in the training set for the weak$^\pm$ model could be filled with classical solutions to the Riemann problem.

Appendix C. Training history

Fig. B.15 shows the loss history of the weak, moderate, strong, and weak$^\pm$ models. We use the Adam optimizer for all models. For the weak and weak$^\pm$ model, the initial learning rate is $\eta = 10^{-2}$. The learning rate is divided by 10 after 100 and 200 iterations. The training process is terminated after 300 iterations. The moderate and strong models are trained for 600 iterations. The initial learning rate is $\eta = 10^{-3}$ and we divide it by 3 after 200 and 400 iterations.

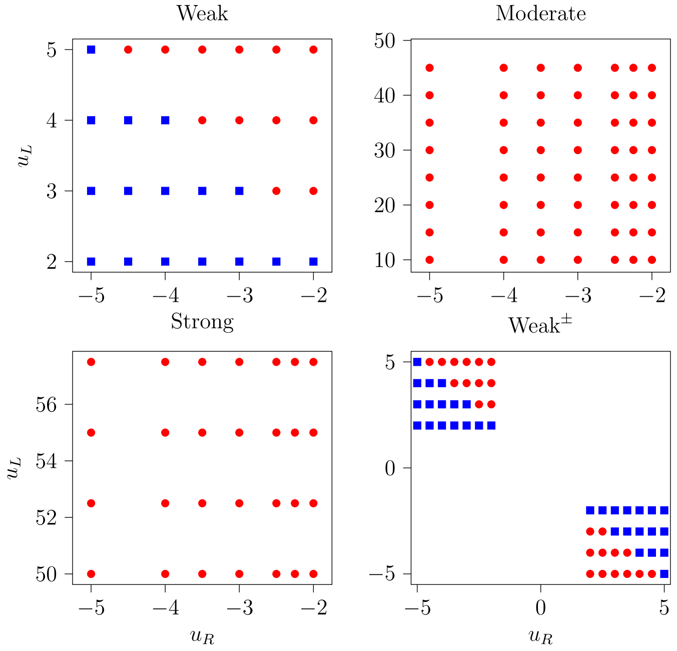

Fig. B.14. Riemann initial configurations in training sets. (●) Nonclassical shock + Classical shock, (■) Nonclassical shock + rarefaction.

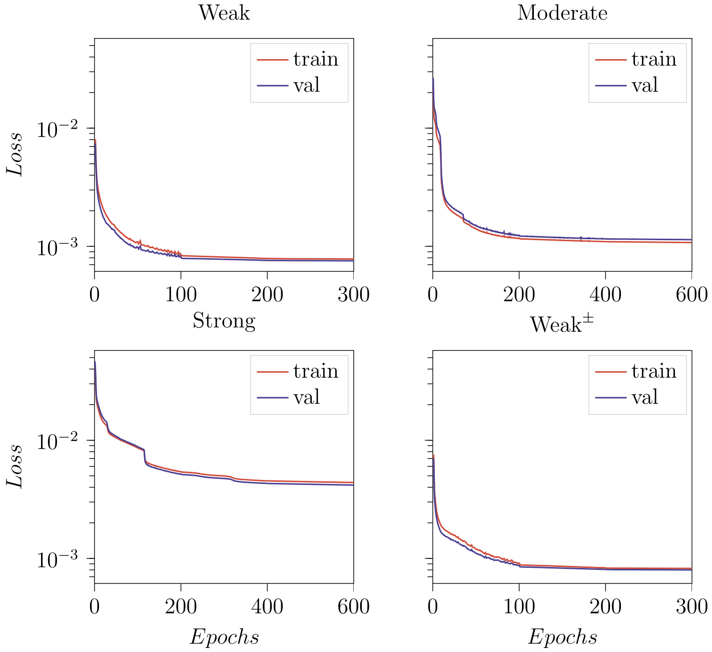

Fig. B.15. Loss history of the NN scheme models. From top left to bottom right: Weak, moderate, strong, and weak$^\pm$ model.

Appendix D. CPU/GPU benchmark

We provide the CPU/GPU time taken to generate the training dataset and train the neural network, see Table D.3. In Table D.4, we compare the wall-clock times of the NN scheme with the 8-th-order WCD scheme for weak, moderate, and strong shock Riemann problems. We choose the following Riemann problems $(u_L, u_R) = (4, -2)$, $(u_L, u_R) = (25, -2)$, and $(u_L, u_R) = (55, -2)$. The numerical setups are detailed in sections 5.1, 5.2, and 5.3, respectively. We have averaged the wall-clock times of the NN scheme and the WCD scheme over 10 runs to get stable averages. Data generation and numerical simulations are performed on an AMD EPYC 7302P CPU. The network training is performed on one Nvidia GTX 2080 GPU.

Table D.3
Wall-clock times (in seconds) for training data generation and the network training.

Model	Data Generation	Training
Weak	34.28	1200.88
Moderate	876.98	7389.39
Strong	426.02	3843.03

Table D.4
Comparison of wall-clock times (in seconds) between the WCD scheme and our NN scheme. The numbers in brackets represent the time needed to evaluate the network inside the finite-volume solver.

Test case	WCD scheme	NN scheme
Weak	0.36	3.41 (3.07)
Moderate	10.29	35.34 (30.75)
Strong	10.44	34.98 (30.51)

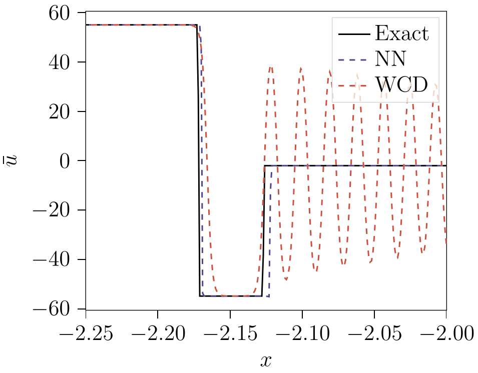

Fig. E.16. NN scheme solution of the Riemann problem for $u_L = 55$ and $u_R = -2$ on an unseen spatial domain. Solution is shown at $t = 4.408 \times 10^{-4}$.

For the weak shock test case which is calculated on a coarse mesh ($n_x = 200$), the NN scheme is roughly 10 times slower than the WCD scheme. Most of the computation time is spent for the network evaluation. This is not surprising as in classical WENO schemes the evaluation of the WENO coefficients is typically very time consuming as well. For the higher resolved test cases ($n_x = 2000$) with moderate and strong shocks, the NN scheme needs roughly 3.5 times the computational resources of the WCD scheme. Considering the substantially increased approximation quality, the computational overhead of the NN scheme is acceptable.

Appendix E. Additional test cases

E.1. Translational invariance

We repeat the experiment with the strong shock, $u_L = 55$ and $u_R = -2$, from Sec. 5.3 on a different spatial domain. We want to demonstrate that the NN scheme is translational invariant. The domain is changed to $[-4, 0]$. The initial discontinuity is located at $x = -3.0$. We choose a resolution of $\Delta x = 1 \times 10^{-3}$ for the NN scheme and compare with the 8-th-order WCD scheme at the same resolution.

Fig. E.16 shows that the NN scheme is translational invariant. Change of domain does not affect the approximation quality of the NN scheme.

E.2. Shock strength study

We assess the extrapolation capabilities of the NN scheme. Fig. E.17 shows solutions of the strong shock model with shock strengths significantly larger than the ones in the training set. We keep $u_R = -5$ fixed and increase u_L step by step, $u_L = \{60, 70, 75, 80\}$. The shock strength is increased from around 120 to 160. The maximum shock strength in the training set was around 115, corresponding to $u_L = 57.5$. The resolution is $\Delta x = 1 \times 10^{-3}$. Results are compared with an 8-th-order WCD scheme at the same resolution.

The NN scheme extrapolates well for shocks up to a strength of 150. The configurations with $u_L = 70$ and $u_L = 75$ show slight overshoots at the nonclassical shock, however the overall shock structure as well as the intermediate value is captured very well. For the strongest shock case, the NN scheme fails to approximate nonclassical and classical shocks. The NN scheme tries to approximate the intermediate state but fails to resolve the nonclassical-classical shock structure.

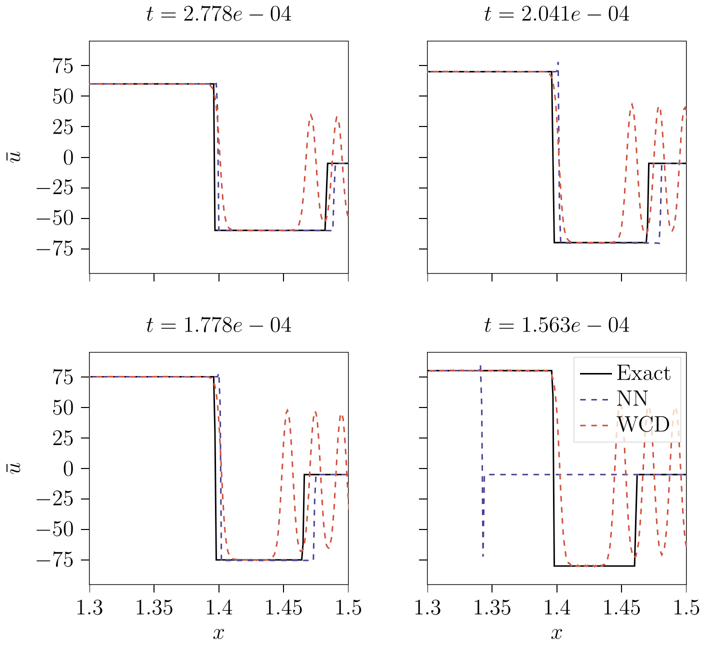

Fig. E.17. Extrapolation capabilities of the NN scheme. From top left to bottom right: $u_L = \{60, 70, 75, 80\}$ and $u_R = -5$.

References

[1] P.G. LeFloch, Hyperbolic Systems of Conservation Laws: The Theory of Classical and Nonclassical Shock Waves, Lectures in Mathematics, ETH Zürich, Birkhäuser Verlag, 2002.

[2] P.G. LeFloch, S. Mishra, Numerical methods with controlled dissipation for small-scale dependent shocks, arXiv:1312.1280, https://doi.org/10.1017/S0962492914000099, 2014.

[3] R.J. LeVeque, Numerical Methods for Conservation Laws, Birkhäuser, Basel, 1992.

[4] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics a Practical Introduction, 3rd edition, Springer Verlag, 2009.

[5] A. Harten, B. Engquist, S. Osher, S.R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III, J. Comput. Phys. 131 (1) (1997) 3–47, https://doi.org/10.1006/jcph.1996.5632.

[6] X.D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1) (1994) 200–212, https://doi.org/10.1006/jcph.1994.1187.

[7] L. Fu, X.Y. Hu, N.A. Adams, A family of high-order targeted ENO schemes for compressible-fluid simulations, J. Comput. Phys. 305 (2016) 333–359, https://doi.org/10.1016/j.jcp.2015.10.037.

[8] L. Fu, X.Y. Hu, N.A. Adams, Targeted ENO schemes with tailored resolution property for hyperbolic conservation laws, J. Comput. Phys. 349 (2017) 97–121, https://doi.org/10.1016/j.jcp.2017.07.054.

[9] J. Ernest, P.G. LeFloch, S. Mishra, Schemes with well-controlled dissipation, SIAM J. Numer. Anal. 53 (1) (2015) 674–699, https://doi.org/10.1137/130933551.

[10] T.Y. Hou, P.G. LeFloch, Why nonconservative schemes converge to wrong solutions: error analysis, Math. Comput. 62 (206) (1994) 497, https://doi.org/10.2307/2153520, https://www.jstor.org/stable/2153520.

[11] B.T. Hayes, P.G. LeFloch, Nonclassical shocks and kinetic relations: strictly hyperbolic systems, SIAM J. Math. Anal. 31 (5) (2000) 941–991, https://doi.org/10.1137/S0036141097319826.

[12] N.A. Adams, S. Hickel, S. Franz, Implicit subgrid-scale modeling by adaptive deconvolution, J. Comput. Phys. 200 (2) (2004) 412–431, https://doi.org/10.1016/j.jcp.2004.04.010.

[13] D. Jacobs, B. McKinney, M. Shearer, Traveling wave solutions of the modified Korteweg-deVries-Burgers equation, J. Differ. Equ. 116 (2) (1995) 448–467, https://doi.org/10.1006/jdeq.1995.1043.

[14] B.T. Hayes, P.G. LeFloch, Non-classical shocks and kinetic relations: scalar conservation laws, Arch. Ration. Mech. Anal. 139 (1) (1997) 1–56, https://doi.org/10.1007/s002050050046.

[15] B.T. Hayes, P.G. LeFloch, Nonclassical shocks and kinetic relations: finite difference schemes, SIAM J. Numer. Anal. 35 (6) (1998) 2169–2194, https://doi.org/10.1137/S0036142997315998.

[16] N. Bedjaoui, P.G. LeFloch, Diffusive-dispersive traveling waves and kinetic relations. Part I: nonconvex hyperbolic conservation laws, J. Differ. Equ. 178 (2) (2002) 574–607, https://doi.org/10.1006/jdeq.2000.4009.

[17] A.L. Bertozzi, A. Münch, M. Shearer, Undercompressive shocks in thin film flows, Phys. D, Nonlinear Phenom. (1999), https://doi.org/10.1016/S0167-2789(99)00134-7.

[18] A.L. Bertozzi, M. Shearer, Existence of undercompressive traveling waves in thin film equations, SIAM J. Math. Anal. 32 (1) (2000) 194–213, https://doi.org/10.1137/S0036141099350894.

[19] P.G. LeFloch, M. Shearer, Non-classical Riemann solvers with nucleation, Proc. R. Soc. Edinb., Sect. A, Math. 134 (5) (2004) 961–984, https://doi.org/10.1017/s0308210500003577.

[20] P.G. LeFloch, M. Mohammadian, Why many theories of shock waves are necessary: kinetic functions, equivalent equations, and fourth-order models, J. Comput. Phys. 227 (8) (2008) 4162–4189, https://doi.org/10.1016/j.jcp.2007.12.026.

[21] G.M. Coclite, K.H. Karlsen, A singular limit problem for conservation laws related to the Camassa-Holm shallow water equation, Commun. Partial Differ. Equ. 31 (8) (2006) 1253–1272, https://doi.org/10.1080/03605300600781600.

[22] A. Bressan, A. Constantin, Global dissipative solutions of the Camassa–Holm equation, Anal. Appl. 05 (01) (2007) 1–27, https://doi.org/10.1142/s0219530507000857.

[23] M. Slemrod, Admissibility criteria for propagating phase boundaries in a van der Waals fluid, Arch. Ration. Mech. Anal. 81 (4) (1983) 301–315, https://doi.org/10.1007/BF00250857.

[24] A. Beljadid, P.G. LeFloch, S. Mishra, C. Parés, Schemes with well-controlled dissipation. Hyperbolic systems in nonconservative form, Commun. Comput. Phys. 21 (4) (2017) 913–946, https://doi.org/10.4208/cicp.OA-2016-0019.

[25] B. Boutin, C. Chalons, F. Lagoutière, P.G. Lefloch, Convergent and conservative schemes for nonclassical solutions based on kinetic relations, Interfaces Free Bound. 10 (3) (2008) 399–421.

[26] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, Annu. Rev. Fluid Mech. 52 (1) (2020) 477–508, https://doi.org/10.1146/annurev-fluid-010719-060214, arXiv:1905.11075.

[27] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, Annu. Rev. Fluid Mech. 51 (1) (2019) 357–377, https://doi.org/10.1146/annurev-fluid-010518-040547, https://www.annualreviews.org/doi/10.1146/annurev-fluid-010518-040547.

[28] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations, Science 367 (6481) (2020) 1026–1030, https://doi.org/10.1126/science.aaw4741.

[29] M. Raissi, P. Perdikaris, G.E. Karniadakis, Numerical Gaussian processes for time-dependent and non-linear partial differential equations, SIAM J. Sci. Comput. (2017), arXiv:1703.10230, https://doi.org/10.1137/17M1120762, http://arxiv.org/abs/1703.10230.

[30] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707, https://doi.org/10.1016/j.jcp.2018.10.045.

[31] S.L. Brunton, J.L. Proctor, J.N. Kutz, W. Bialek, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, Proc. Natl. Acad. Sci. USA 113 (15) (2016) 3932–3937, https://doi.org/10.1073/pnas.1517384113, arXiv:1509.03580.

[32] S.H. Rudy, S.L. Brunton, J.L. Proctor, J.N. Kutz, Data-driven discovery of partial differential equations, Sci. Adv. 3 (4) (2017) e1602614, https://doi.org/10.1126/sciadv.1602614, arXiv:1609.06401.

[33] S. Thaler, L. Paehler, N.A. Adams, Sparse identification of truncation errors, J. Comput. Phys. 397 (apr 2019), https://doi.org/10.1016/j.jcp.2019.07.049, arXiv:1904.03669, http://arxiv.org/abs/1904.03669.

[34] Z. Long, Y. Lu, X. Ma, B. Dong, PDE-NET: learning PDEs from data, in: Proceedings of the 35th International Conference on Machine Learning, 2018, arXiv:1710.09668v2.

[35] B. Lusch, J.N. Kutz, S.L. Brunton, Deep learning for universal linear embeddings of nonlinear dynamics, Nat. Commun. 9 (1) (2018) 1–10, https://doi.org/10.1038/s41467-018-07210-0, arXiv:1712.09707.

[36] Y. Bar-Sinai, S. Hoyer, J. Hickey, M.P. Brenner, Learning data-driven discretizations for partial differential equations, Proc. Natl. Acad. Sci. USA 116 (31) (2019) 15344–15349, https://doi.org/10.1073/pnas.1814058116, arXiv:1808.04930.

[37] B. Stevens, T. Colonius, FiniteNet: a fully convolutional LSTM network architecture for time-dependent partial differential equations, arXiv:2002.03014, http://arxiv.org/abs/2002.03014, feb 2020.

[38] B. Stevens, T. Colonius, Enhancement of shock-capturing methods via machine learning, arXiv:2002.02521, http://arxiv.org/abs/2002.02521, feb 2020.

[39] J. Magiera, D. Ray, J.S. Hesthaven, C. Rohde, Constraint-aware neural networks for Riemann problems, J. Comput. Phys. 409 (2020) 109345, https://doi.org/10.1016/j.jcp.2020.109345, arXiv:1904.12794.

[40] C.W. Shu, B. Cockburn, C. Johnson, E. Tadmor, Advanced numerical approximation of nonlinear hyperbolic equations, in: Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws, in: Lecture Notes in Mathematics, vol. 1697, 1998, pp. 325–432, URL: papers2://publication/uuid/72FB4A02-DC77-480C-8F60-D72A0E7091F0.

[41] D.P. Kingma, J.L. Ba, Adam: a method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, International Conference on Learning Representations, ICLR, 2015, arXiv:1412.6980.

[42] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, https://www.deeplearningbook.org/.

[43] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, G. Brain, TensorFlow: a system for large-scale machine learning TensorFlow: a system for large-scale machine learning, in: Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), 2016, https://tensorflow.org.

[44] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010 Proceeding, 2010, http://www.iro.umontreal.