

Physics-aware recurrent convolutional neural networks for modeling multiphase compressible flows

Xinlun Cheng^{a,b}, Phong C.H. Nguyen^a, Pradeep K. Seshadri^c, Mayank Verma^c, Zoë J. Gray^a, Jack T. Beerman^a, H.S. Udaykumar^c, Stephen S. Baek^{a,d,*}

^a School of Data Science, University of Virginia, 1919 Ivy Rd., Charlottesville, 22903, VA, USA

^b Department of Astronomy, University of Virginia, 530 McCormick Rd., Charlottesville, 22904, VA, USA

^c Department of Mechanical Engineering, University of Iowa, 103 S. Capitol St., Iowa City, 52242, IA, USA

^d Department of Mechanical and Aerospace Engineering, University of Virginia, 351 McCormick Rd., Charlottesville, 22904, VA, USA

ARTICLE INFO

Keywords:

Physics-informed machine learning
Physics-aware recurrent convolutional neural networks
Data-driven modeling

ABSTRACT

Multiphase compressible flow systems can exhibit unsteady and fast-transient dynamics, marked by sharp gradients and discontinuities, and material boundaries that interact with the evolving flow. The transient nature of the dynamics presents challenges to employing artificial intelligence (AI) and data-driven models for predicting flow behaviors. In this study, we explore the potential of physics-aware recurrent convolutional neural networks (PARC) to model the spatiotemporal dynamics of multiphase flows in the presence of shocks and reaction fronts. PARC is a neural network model that incorporates the generic form of the diffusion–advection–reaction equation in its network architecture, which mimics the process of solving the governing equations of fluid flows. In contrast to physics-informed machine learning approaches such as physics-informed neural networks (PINNs) where models are trained to directly minimize the residual of governing equations, PARC takes a dynamical systems viewpoint and does not seek to minimize potentially nonconvex and nonlinear loss terms. To assess the ability of PARC to accurately learn and simulate the physics of multiphase flows, we train and test PARC on various flow simulation problems, including the Burgers' equation, fluid flow behind a cylindrical cross-section, and unsteady shock interactions with a particle at varying Mach numbers. We analyze PARC's performance and examine sources of error in its prediction, in terms of differentiation and integration schemes and different weighting strategies for the model update. Based on our observations, we discuss PARC's capabilities and limitations in multiphase flow applications and propose future research directions.

1. Introduction

In recent years, the computational physics community has witnessed a surge in the adoption of machine-learning-based flow modeling methodologies in the literature. These methodologies, broadly known as *physics-informed machine learning* (PIML) (Karniadakis et al., 2021), employ machine learning models to approximate the solution space of the governing partial differential equations (PDEs) that describe the evolution of fields in the dynamics of continua.

A notable example of such data-driven physics modeling methods is the family of *physics-informed neural networks* (PINNs) (Raissi et al., 2019a). In a typical PINN framework, a general nonlinear dynamics PDE is represented as $\frac{\partial \mathbf{x}}{\partial t} = \mathcal{D}[\mathbf{x}; \lambda]$, $\mathbf{r} \in \Omega$, $t \in [0, T]$, where $\mathbf{x}(\mathbf{r}, t)$ is the time-evolving solution field, $\mathcal{D}[\cdot; \lambda]$ is a nonlinear operator parameterized by λ , and $\Omega \subset \mathbb{R}^d$ is a spatial domain where the physics is defined. PINN approximates the solution field $\mathbf{x}$ using an artificial neural network $\hat{\mathbf{x}}(\mathbf{r}, t; \theta)$ with network parameters θ . The network parameters are

trained to minimize both the prediction error $L_{\mathbf{x}} = \|\mathbf{x} - \hat{\mathbf{x}}\|$ and the PDE residual $L_r = \|\frac{\partial \mathbf{x}}{\partial t} - \mathcal{D}[\mathbf{x}; \lambda]\|$ at sample locations $\{(\mathbf{r}_i, t_i) \mid i = 1, \dots, N\}$ in spacetime. During this process, the differential operators in $\mathcal{D}[\mathbf{x}; \lambda]$ are computed via the automatic differentiation (Baydin et al., 2018) applied to the computational graph of the neural network $\hat{\mathbf{x}}$, as opposed to calculation of discrete gradients on a computational mesh.

Especially in fluid mechanics, PINNs have been tested on a variety of fluid conditions, such as wake flows (Cai et al., 2022), incompressible (Jin et al., 2021) and compressible flows (Mao et al., 2020), turbulent flows (Lucor et al., 2022), biomedical flows (Yin et al., 2021), and others (Yang et al., 2019; Raissi et al., 2019b; Lou et al., 2021). For these problems, PINNs have proved to be capable of approximating the solution space of the Navier–Stokes equations, while satisfying initial and boundary conditions, as well as other physical constraints.

* Corresponding author at: School of Data Science, University of Virginia, 1919 Ivy Rd., Charlottesville, 22903, VA, USA.
E-mail address: baek@virginia.edu (S.S. Baek).

However, challenges arise when it comes to the application of PINN and other PIML to multiphase flows. Multiphase flow systems are highly dynamic, often featuring unsteady and fast-transient behaviors characterized by sharp gradients and discontinuous boundaries that constantly interact with the evolving flow. The transient and discontinuous dynamics poses significant challenges when it comes to employing convolution neural networks for modeling and predicting flow behaviors.

For instance, previous research has highlighted a significant challenge in the gradient descent training of PINNs, in which multiple PDE residual loss terms may produce conflicting loss gradients hindering the convergence of the PINN training (Wang et al., 2021a). This can be particularly problematic for multiphase flow problems, where there are usually many coupled PDEs and physics constraints and, hence, a higher chance of producing conflicting loss terms.

Furthermore, PINNs are known to be biased towards low-frequency components (Wang et al., 2022). From the multiphase flow modeling standpoint, this means that PINNs are not adept at capturing high-frequency features such as sharp, localized features, discontinuities, unsteady oscillations, and other transient flow features, which are typical attributes of multiphase compressible flow phenomena.

To address these challenges with PINNs, there have been several remedies proposed. For example, adaptive weighting (Wang et al., 2022) and Fourier features (Wang et al., 2021b) have been suggested to mitigate spectral bias issues and enhance the convergence of PINN training. For example, Liu et al. (2024) introduced a physics-dependent weighting scheme that weighs PDE residuals differently based on the divergence of the velocity field, to enhance the convergence of PINN in discontinuous regions. Similarly, Mao et al. (2020) introduced adaptive sampling, in which data points are more densely sampled near discontinuities and sharp interfaces in the solution function of the Euler equation. On the other hand, curriculum training (Krishnapriyan et al., 2021), which involves initial training on simpler flow problems (with smooth flow fields), followed by adaptation to more complex scenarios, was also proven to be effective.

Here, we note that the vast majority of these previous works circumvent the issues regarding sharp features and discontinuities by meticulously tweaking the loss function landscape of PINNs, or by making significant modifications to the architecture or training algorithms underlying the neural networks. For instance, adaptive weighting and adaptive sampling essentially involve the process of manually balancing the influences of smooth and sharp regions by manipulating weighting formulae and sampling densities. Fourier features are also essentially a type of advanced weighting scheme, in which weighting factors are determined by local patterns of fluid fields. Curriculum training can also be understood along the same line, where the influences of smooth and sharp fluid features are varied across different stages of training.

In contrast with PINN-type frameworks, relatively understudied are the approaches based on the inductive design of neural network architectures. For instance, previous research by Nguyen et al. (2023a,b, 2024) introduced a PIML model called *physics-aware recurrent convolutional neural networks* (PARC). In contrast to PINN, where a neural network directly approximates the solution field $\mathbf{x}$, PARC utilizes convolutional neural networks to learn the nonlinear operator $\mathcal{D}[\mathbf{x}; \lambda]$ and its time integration to solve for $\mathbf{x}$, separately. Particularly, in the most recent variant of PARC, called PARCv2 (Nguyen et al., 2024), the network architecture was designed to model the generic form of reaction–diffusion–advection equations $\frac{\partial \mathbf{x}}{\partial t} = k \Delta \mathbf{x} - (\mathbf{u} \cdot \nabla) \mathbf{x} + \mathbf{R}$ which is a template for the transport equations solved in thermo-fluid mechanics. In the training phase of PARC assimilates data as well as time-updates of the data as it evolves as the solution of an advection–diffusion–reaction problem. No separate physics-informed loss terms are used in the training, although, in principle, PARC could be trained with PINN loss terms in a supplementary fashion.

Here, the main hypothesis is that the formulation and architecture of PARC could be more suitable for multiphase flow problems compared to other PIML approaches because of the following reasons.

First, PARC circumvents the issue of conflicting loss gradients, as there are no PDE residual terms involved in its training. Instead, PARC complies with the physics principles inductively using its reaction–diffusion–advection architecture. This is advantageous for multiphase problems, where there are typically a large number of PDEs and constraints involved in describing the physics.

Second, learning the nonlinear operator $\mathcal{D}$ and the time integration $u = \int_t \mathcal{D} dt$ separately, as in PARC, is arguably more advantageous than the direct approximation of the solution function $\mathbf{x}$ as in PINN. As mentioned, the solution function $\mathbf{x}$ in multiphase flows may include sharp gradients and discontinuous features, as well as transient features that evolve on short time scales. In our experiments, we empirically observe that learning the nonlinear operator $\mathcal{D}$ and integrating it separately preserves those sharp and transient features better. This is likely because the training of PARC inherently is a hybrid of the strong form (learning the nonlinear operator $\mathcal{D}$) and the weak form (learning the integral of $\mathcal{D}$), in contrast to PINN, where only the strong form (PDE residuals) is considered (Liu et al., 2024). Therefore PARC is trained using the data, and spatial and temporal gradients of data.

Last, the fact that PARC flexibility does not require explicitly defined PDE residual terms in the loss function makes it more practical for real-world multiphase flow modeling and simulation workflows. For example, compressible multiphase flow problems usually involve a large number of equations, with different material properties in each phase, and mixture rules in two-phase regions. The precise physical constraints (such as equations of state) to apply will vary spatially. Whereas the dependent variables of primary interest to describe the physics of the system may be relatively few and smoothly varying across phases. With the PDE residual approach, one needs to model all variables in play with appropriate constraints depending on the local phase, even if only a subset is of interest. Moreover, if there are no equations available, PARC offers versatility in modeling the flow within the general reaction–diffusion–advection framework, whereas PINNs may suffer from the lack of PDE residuals to inform the physics principles.

Therefore, the objectives of this paper are as follows. Firstly, in Section 2, we delineate the mathematical formulation of PARC and its computational implementation. We then conduct a comparative analysis between PARC and PINN on two standard benchmark problems, namely the Burgers' equation (Section 3.1) and the Navier–Stokes equations (Section 3.2). Subsequently, we elucidate PARC's capabilities and limitations in modeling real-world multiphase flow phenomena by testing it on unsteady shock calculations interacting with an embedded cylindrical particle (Section 4). In the concluding section (Section 6), we delve into the underlying causes of failure cases, offering insights into potential areas for future research and development.

2. Methods

2.1. Problem formulation

The transport of fluids, including both incompressible and compressible flows, and inert/reactive flows, can be characterized by the following general form of the convection–diffusion–reaction equations:

$$\frac{\partial \mathbf{x}}{\partial t} = k(\nabla \cdot \nabla) \mathbf{x} - (\mathbf{u} \cdot \nabla) \mathbf{x} + \mathbf{R}_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, \mathbf{c}), \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \mathbf{R}_{\mathbf{u}}(\mathbf{x}, \mathbf{u}, \mathbf{c}). \quad (2)$$

The state and velocity fields, $\mathbf{x}(\mathbf{r}, t)$ and $\mathbf{u}(\mathbf{r}, t)$ respectively, are defined in Galilean spacetime, represented by a spatial position vector $\mathbf{r} \in \mathbb{R}^d$, and a time variable t . The function $\mathbf{R}_{\mathbf{x}}(\mathbf{x}, \mathbf{u}, \mathbf{c})$ describes sources of generation or consumption related to the state fields with $\mathbf{c}$ serving as constant parameters, and $\mathbf{R}_{\mathbf{u}}(\mathbf{x}, \mathbf{u}, \mathbf{c})$ encapsulates forces (internal and external) exerted on a unit volume element. k is the diffusivity coefficient. Velocity diffusion term can be included if necessary by

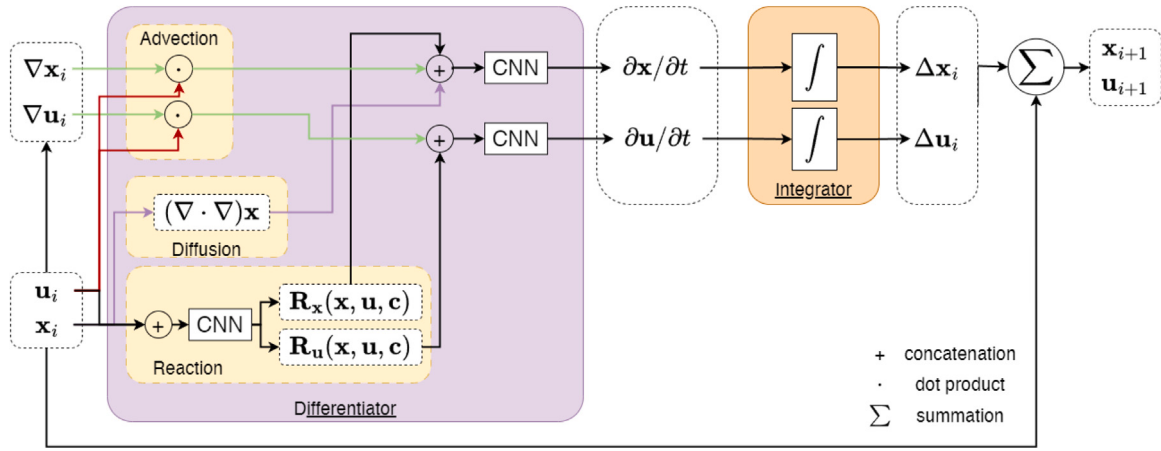


Fig. 1. PARCV2 architecture for multiphase flow modeling.

passing the velocity variables to the diffusion layer, the same way as the inclusion of diffusion of state variables, but we did not include it in any of our experiments. Thus it was omitted from Eq. (2).

For an initial value problem, the above PDE system is subjected to the following initial conditions:

$$\begin{aligned} \mathbf{x}|_{t=0} &= \mathbf{x}_0 \\ \mathbf{u}|_{t=0} &= \mathbf{u}_0, \end{aligned} \quad (3)$$

which may be solved numerically using a forward update scheme:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{x}_k + \int_{t_k}^{t_{k+1}} \frac{\partial \mathbf{x}}{\partial t} dt \\ \mathbf{u}_{k+1} &= \mathbf{u}_k + \int_{t_k}^{t_{k+1}} \frac{\partial \mathbf{u}}{\partial t} dt. \end{aligned} \quad (4)$$

In PARCV2, the problem of learning the dynamics of flow is accomplished via a two-stage process, in which convolutional neural networks F_x and F_u learn to estimate the partial differential equations in (1) and (2), respectively, followed by integral solvers Ψ_x and Ψ_u computing the time integrals $\int F_x dt$ and $\int F_u dt$.

2.2. Architecture design

Fig. 1 presents the architecture of PARCV2 for multiphase flow modeling. The network consists of two main components: the differentiator (purple box) and the integrator (orange box). At each step, the state variables $\mathbf{x}_i$ and velocity variables $\mathbf{u}_i$ at current time t_i are input to the differentiator, which calculates the time derivatives $\frac{\partial \mathbf{x}}{\partial t}$ and $\frac{\partial \mathbf{u}}{\partial t}$. The differentiator first computes the gradients of each component of the state and velocity fields, denoted as $\nabla \mathbf{x}^{(i)}$ and $\nabla \mathbf{u}^{(i)}$, respectively. These gradients are then used to compute the advection terms $(\mathbf{u} \cdot \nabla) \mathbf{x}$ and $(\mathbf{u} \cdot \nabla) \mathbf{u}$, as well as the diffusion term $(\nabla \cdot \nabla) \mathbf{x}$. The reaction terms $\mathbf{R}_x$ and $\mathbf{R}_u$ are computed from the input state $\mathbf{x}$ and velocity $\mathbf{u}$ variables using U-Net (Ronneberger et al., 2015). These reaction terms are then concatenated with the advection and diffusion terms channel-wise to derive the time derivatives of state and velocity fields $\frac{\partial \mathbf{x}}{\partial t}$ and $\frac{\partial \mathbf{u}}{\partial t}$. Finally, the integrator (orange box) integrates these time derivatives to predict the state and velocity fields of the next time step. The process is recursively applied for a fixed number of time steps.

2.3. Learning objective

The training of PARCV2 is cast as an optimization problem where the following loss function is minimized for each simulation instance $\{\mathbf{x}_k, \mathbf{u}_k, t_k\}$:

$$\begin{aligned} \mathcal{L}(\theta_x, \theta_u) &:= \sum_{t_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k - \Psi_x[F_x(\mathbf{x}_k|\theta_x)]\|_1 \\ &+ \sum_{t_k} \|\mathbf{u}_{k+1} - \mathbf{u}_k - \Psi_u[F_u(\mathbf{u}_k|\theta_u)]\|_1 \end{aligned} \quad (5)$$

where θ_x and θ_u denotes the network parameters of F_x and F_u , respectively. The total training loss is averaged over multiple simulation instances.

We employ the Adam solver (Kingma and Ba, 2017), a gradient-based optimizer, to solve the above optimization problem. The necessary gradient $\nabla \mathcal{L}/\theta_u$ and $\nabla \mathcal{L}/\theta_x$ are computed by differentiating through the integration operation in the forward pass.

3. Advection–diffusion modeling and subsonic incompressible flows

In this section, we evaluate the performance of PARCV2 on advection-diffusion modeling and subsonic incompressible flow problems to establish a baseline before expanding it to supersonic compressible flows. For this purpose, we also compare the performance of PARCV2 against other benchmark PIML models including PINN (PhysCRNet (Ren et al., 2022)), and Fourier Neural Operators (Li et al., 2020) (FNO, including both physics-informed and physics-naïve variants). We refer readers interested in the comparison of PARCV2 versus the original PARC to Nguyen et al. (2024), as huge improvements of PARCV2 was already covered in details in that work. For all experiments below, we used the PARCV2 architecture and training strategy used in the original work of Nguyen et al. (2024). Similarly, for other benchmark models, we also borrowed the same architecture and protocols as implemented in the original works, except for minor modifications to interface models with specific configurations of the experiment. In all of our experiments, pressure, density and temperature are treated as state variables.

In the analyses below, we focus on major attributes that are of interest to the multiphase flow community: sharp gradients, e.g. **shocks**, where the flow velocity field exhibits rapid changes in a small region spatially or temporally, unsteady flow conditions, e.g. where the flow field shows signs coherent vortical structure or interfacial deformation, and conservation constraints where certain property of the flow field should remain invariant with time as indicated in the governing equation.

3.1. 2D Burgers' equation

We first test the capability of PARCV2 for solving Burgers' equation describing the formation and decay of a shock wave, where a sharp gradient is generated from a relatively smooth flow field and subsequent decays due to the viscosity of the flow. We consider the following two-dimensional (2D) Burgers' equation problem with initial and boundary

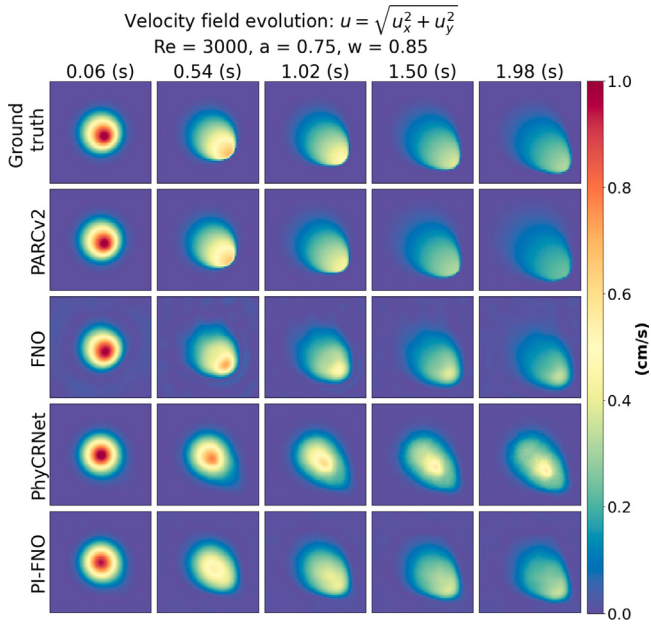


Fig. 2. Comparison of various ML models on 2D Burgers' equation.

conditions:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} &= -(\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{Re} (\nabla \cdot \nabla) \mathbf{u} \\ \mathbf{u}_0(\mathbf{r}, t; a, w) &= \begin{bmatrix} a \cdot \exp(-\|\mathbf{r}\|_2^2/w) \\ a \cdot \exp(-\|\mathbf{r}\|_2^2/w) \end{bmatrix} \\ \Omega &= [-3.0, 3.0] \times [-3.0, 3.0] \\ \mathbf{u}_0(\mathbf{r}, t) &= 0 \quad \forall \mathbf{r} \in \partial\Omega \end{aligned} \quad (6)$$

To produce ground truth data, the above problem was resolved on 64×64 square grids, with a total simulation length of 2 s resolved in 1,500 time steps. The original simulations were then downsampled to 100 evenly distributed time steps. No change to the spatial resolution was applied.

For the training dataset, the fluid parameters – the amplitude a (cm/s), width w (cm) of the initial velocity distribution and the Reynolds number Re (cm^2/s) – were chosen from the combinations of $a = \{0.5, 0.6, 0.7, 0.8, 0.9\}$, $w = \{0.7, 0.8, 0.9, 1.0\}$, and $Re = \{1000, 2500, 5000, 7500, 10000\}$. For the test dataset, the fluid parameters were chosen from $a = \{0.35, 0.40, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95, 1.00\}$, $w = \{0.55, 0.6, 0.65, 0.75, 0.85, 0.95, 1.05\}$, and $Re = \{100, 500, 3000, 6500, 12500, 15000\}$. These fluid parameters in the test set were chosen such that they could ascertain the models' capabilities in both interpolation and extrapolation.

Fig. 2 displays the predicted evolution of flow speed $\|\mathbf{u}\|$ in comparison with the ground truth. The columns in the figure show the predicted solutions at different instants of time (noted above the panels); the rows correspond to the different prediction techniques. The first row shows the ground truth, second PARCv2 predictions, third FNO, fourth PhyCRNet, and last physics informed FNO (PI-FNO). From visual inspection of the figure, PARCv2 appears to be capable of predicting the formation of the sharp gradient features on the shock front, as well as the decay on the tail end, to a high degree of accuracy, whereas other models tend to predict blurred and smeared fronts. In particular, the sharp gradient regions are most smeared out for the FNO-based models, likely due to the Fourier mode cutoff. This observation can be quantitatively confirmed by the root mean squared errors (RMSEs) reported in the first column of Table 1, in which PARCv2 yielded RMSE about half of those of the other models.

Table 1

Prediction accuracy and solution quality of PARCv2 and the other baseline models on the Burgers' equation. The prediction accuracy was evaluated with root mean squared error (RMSE) between the prediction and ground truth simulations in the test set. The solution quality was evaluated with the residual of the Burgers' equation. RMSE and PDE residual were calculated on the testing set, which uses the following combination of fluid parameters: $a = \{0.35, 0.40, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95, 1.00\}$, $w = \{0.55, 0.6, 0.65, 0.75, 0.85, 0.95, 1.05\}$, and $Re = \{100, 500, 3000, 6500, 12500, 15000\}$.

Model	RMSE (cm/s)	$\ f_u\ $ (cm/s^2)
DNS (Ground truth)	–	0.1241
FNO	0.0289	0.1537
PhyCRNet	0.0588	0.0560
PI-FNO	0.0338	0.1058
PARCv2	0.0129	0.1292

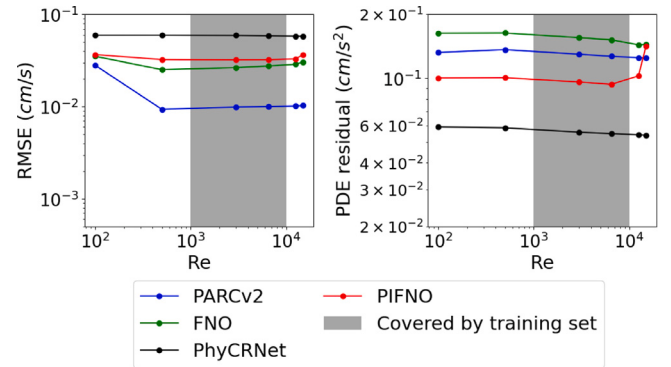


Fig. 3. Prediction accuracy and solution quality of various ML models on 2D Burgers' equation with different Reynolds numbers.

However, the solution quality reported in the second column of Table 1, as measured by the norm of the residual of the Burgers' equation calculated from the predicted flow fields, yields a different picture. As expected, the physics-informed models, namely PhyCRNet and PI-FNO, show the best solution quality, because these two physics-informed models were explicitly trained to minimize the PDE residual. On the contrary, PARCv2 shows a relatively larger deviation from the Burgers' Equation. However, considering that the ground truth data from DNS exhibits a similar solution quality, it is difficult to conclude that the problem is inherent to PARCv2. I.e. PARCv2 learned from the ground truth solution and assimilated the level of solution error present in the ground truth, whereas the PINN approaches additionally "corrected" the ground truth solution to better satisfy the PDE constraint. Therefore, while the velocity field solution produced by PARCv2 more closely emulated the ground truth it carried along the solution error at the same level as ground truth.

Furthermore, Fig. 3 compares the prediction accuracy and solution quality across various Reynolds numbers. In general, all models exhibited good generalizability outside the Reynolds numbers for which they had been trained. The relative performances of the models were consistent across the range, except for PI-FNO whose PDE residual spiked in the high Reynolds number regime. Visual examination of those predicted sequences shows a degradation of performance at the sharp gradient shock front, producing noticeable artifacts and is likely due to the effect of Fourier frequency cutoff in the PI-FNO approach.

3.2. Navier–Stokes equations

A frequently used test problem in the subsonic incompressible regime is the flow around an obstacle with a cylindrical cross-section.

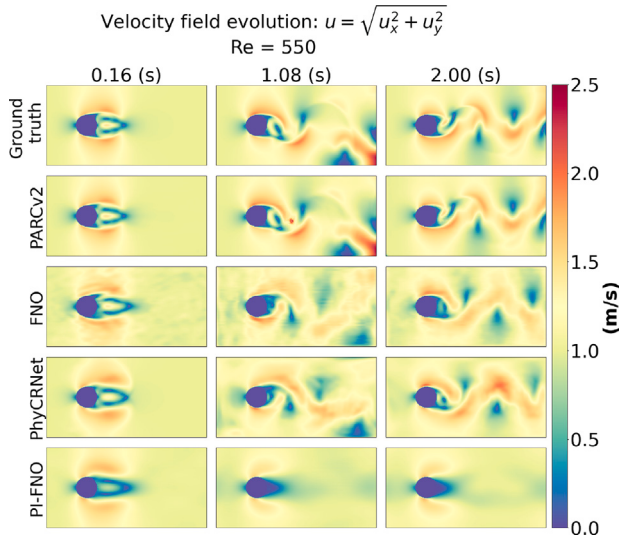


Fig. 4. Evolution of the velocity field of a flow near an embedded cylindrical cross-section predicted by the PIML models.

For this problem, we solve the following form of incompressible Navier–Stokes equations with uniform viscosity:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} &= -\frac{1}{\rho} \nabla p + \nu (\nabla \cdot \nabla) \mathbf{u} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned} \quad (7)$$

We assume a constant flux of fluid flowing into the domain from the left to right horizontally, with fixed viscosity of $\nu = 1 \text{ m}^2/\text{s}$ and velocity of $u_0 = 1 \text{ m/s}$. The flow collides with a cylinder with a 0.25 m diameter at 0.5 m from the left entrance point. The simulation is resolved on a rectangular region 2 m in width and 1 m in height.

For the training and test datasets, direct numerical simulations with Reynolds numbers ranging from 30 to 1,000 were generated. For the training set, the following Reynolds numbers were chosen $Re = \{100, 150, 200, 250, 300, 400, 450, 500, 600, 650, 700, 800\}$. For the testing set, the following Reynolds numbers were chosen $Re = \{20, 60, 80, 140, 350, 550, 750, 850, 900, 950, 1000\}$. In this particular experiment, we aim to investigate the capability of various ML models to predict the unsteady flow and their ability to adhere to conservation laws in the governing equations.

The finite volume method in the ANSYS Fluent package (Ansys Inc., 2023) was used to generate all simulation data. We rasterized each output frame from the solver to a 128×256 image, such that a unit pixel represents $0.0078125 \text{ m} \times 0.0078125 \text{ m}$ area. The total flow time is 2 s, and the simulations were downsampled across 38 evenly spaced time steps.

Fig. 4 illustrates the evolution of the velocity field predicted by the different PIML models (arranged in rows) on a set-aside test case ($Re = 550$). As shown in the ground truth, presented in the top row, the main challenge here is to capture the unsteady flow field and the vortex shedding in the wake of the cylinder. Most of the PIML models were capable of capturing the vortices to a certain degree. However, PI-FNO did not produce the vortices at all. On the other hand, FNO, while it was capable of producing vorticity patterns, exhibited noticeable artifacts in both the irrotational part of the flow field and in the vortical patterns. This is likely an indication that both models suffer from the effect of the Fourier frequency cutoff. Moreover, PI-FNO appears to have failed to converge to an accurate solution due to a strong overregularization by the PDE residuals. The RMSE and solution quality values for PI-FNO, reported in Table 2, indeed support this argument, where the PDE residual for PI-FNO was noticeable lower than all other PIML models.

Table 2

Prediction accuracy and solution quality of PARCv2 and the other baseline models on the Navier–Stokes equations experiment. The prediction accuracy was evaluated with root mean squared error (RMSE) between the prediction and ground truth simulations in the test set. The solution quality was evaluated with the residual of the Navier–Stokes equations and the violation of the divergence-free condition.

Model	RMSE (cm/s^2)	$\ f_u\ $ (m/s^2)	ϵ_{div} (1/s)
DNS (Ground truth)	–	2.2339	0.0198
FNO	0.2411	3.2804	1.0471
PhyCRNet	0.2324	2.6994	0.0597
PI-FNO	0.2230	1.4488	0.0307
PARCv2	0.1556	3.0402	0.3655

Similar to the Burgers' equation study, there was a noticeable tradeoff between the RMSE and the solution quality as well. In this experiment, we measured the residuals of the momentum equation $\|f_u\|$ and the continuity equation ϵ_{div} . As expected, the physics-informed models (PhyCRNet and PI-FNO) produced the smallest values of the PDE residuals, and the models without an explicit physics-informed loss term (FNO and PARCv2) exhibited the largest PDE residuals.

Comparing across different Reynold numbers (as presented in Fig. 5), we noticed that while most models would produce increasing RMSE with larger Reynold numbers, PARCv2 exhibit a different behavior. An increase in prediction error at both the large and small ends of the Reynold numbers was observed within the coverage of the training set. RMSE of simulations with Reynold numbers not covered by the training set report similar values with the Reynold numbers on the edge of the training set coverage, indicating good generalization of PARCv2. PDE residual shows a similar trend: larger PDE residual with increasing Reynold number. This trend remains consistent regardless of the ML model and continues outside the coverage of training set. Divergent-free error remains roughly constant with Reynold number, and this trend applies to all models compared here.

4. Supersonic compressible flow

We now extend PARCv2 to a supersonic compressible flow problem with an embedded cylindrical obstacle. The high-speed, compressible flow will create many numerical challenges such as sharp gradients and discontinuities formed by the compressible fluid, as well as the overall instabilities leading to vortices in the fast-evolving fields. To test the capabilities and limitations of PARCv2 itself, we do not introduce any ad-hoc remedies for these numerical challenges, such as adaptive weighting (Liu et al., 2024) and sampling (Mao et al., 2020) near discontinuities, or smooth-to-sharp training curriculum (Krishnapriyan et al., 2021). Instead, we test the inherent behavior of PARCv2, in the same configuration as in the subsonic incompressible flow problems, on this more challenging multiphase flow problem and probe its limits.

The data used to train PARC for supersonic flow modeling was derived from direct numerical simulation where we solved a system of hyperbolic conservation laws as mentioned in Das et al. (2017), Turner et al. (2024), De Palma et al. (2006), Awasthi et al. (2022), Takahashi et al. (2014). The compressible supersonic flow has been modeled by solving the following set of equations:

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0$$

$$\mathbf{U} = \rho \begin{bmatrix} 1 \\ u \\ v \\ E \end{bmatrix}, \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{bmatrix} \quad (8)$$

where u and v are the fluid velocities in the x and y direction, ρ is the density of fluid, p is the pressure, and E is the total specific energy.

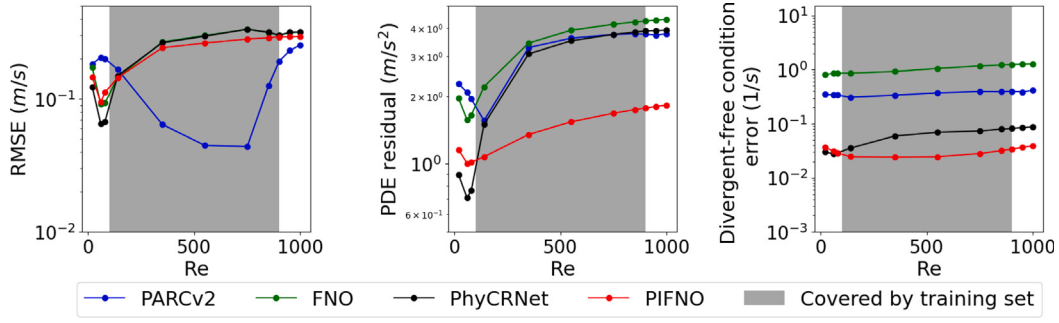


Fig. 5. A comparison of prediction accuracy (RMSE with the ground truth), solution quality (PDE residual of the predicted sequence) and divergence of the predicted flow field.

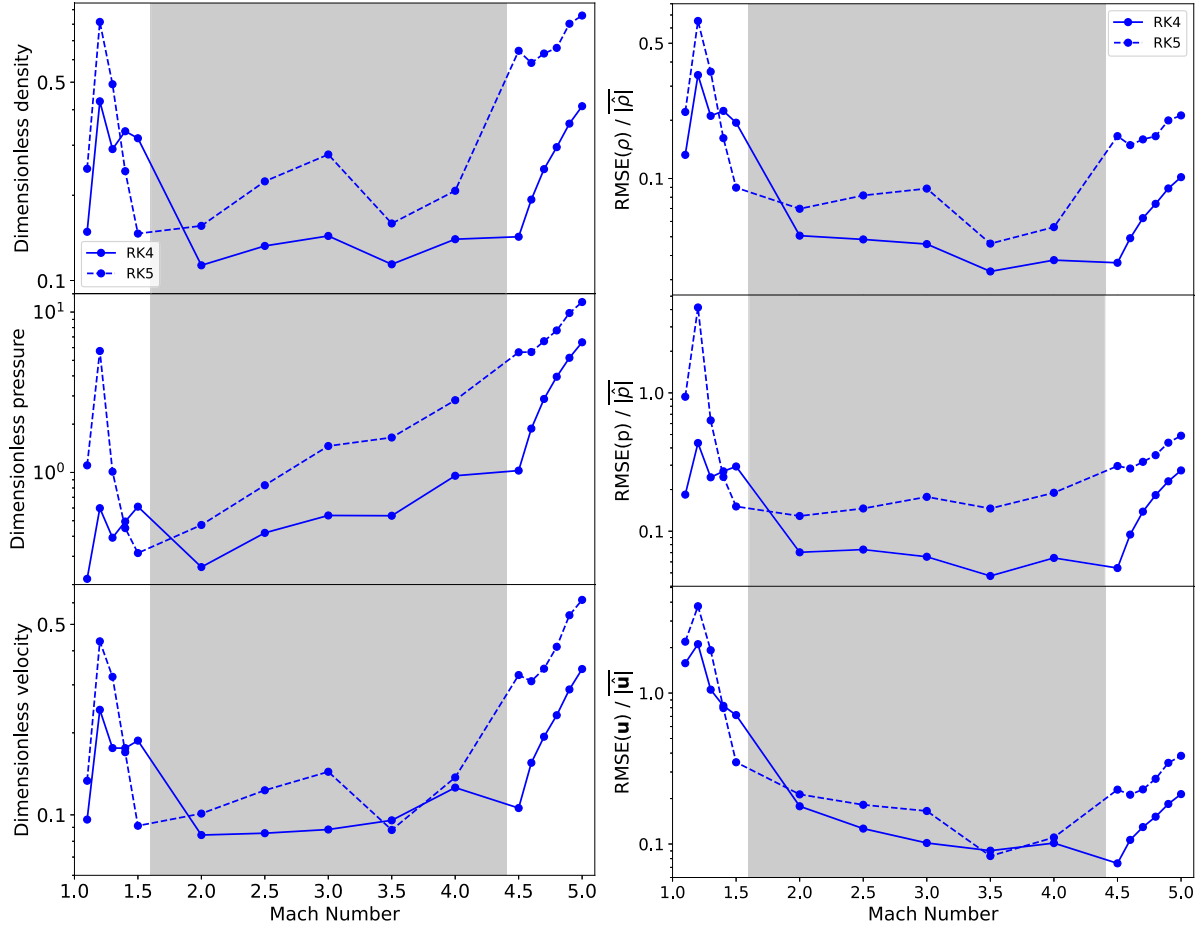


Fig. 6. The absolute (left panel) and relative (right panel) RMSE loss of density (top), pressure (middle) and velocity (bottom) as a function of Mach number of the initial condition of the simulations for RK4 integrator (solid line) and RK5 integrator (dashed line). Gray shaded region indicates the range of Mach number with training set coverage.

It can be given as $E = e + \frac{1}{2}(u^2 + v^2)$, where e is the specific internal energy. For the closure of the above set of equations, the equation of state for the ideal gas is used: $p = \rho e(\gamma - 1)$. Atmospheric conditions are used as reference values.

We consider a domain that has a cylindrical obstacle with diameter D and spans $0 \leq X/D \leq 15$ in the streamwise direction and $0 \leq Y/D \leq 10$ in the transverse direction. The cylindrical obstacle was positioned at $x = 5D$ and $y = 5D$. We have performed the simulations over the range of supersonic shock Mach numbers from 1.1 to 5.0 with a step of size $\Delta M_s = 0.1$ to generate a comprehensive training set. The simulations are advanced in time based on the CFL number of 0.5. The Neumann boundary condition was used for the outlet and the channel walls. A Dirichlet boundary condition based on the shock Mach number (obtained from the Rankine–Hugoniot equations (Anderson,

2003)) was enforced at the inlet. The initial condition in the domain is set to atmospheric conditions ($P_\infty = 101325$ Pa, $T_\infty = 298$ K, $\rho_\infty = 1.23$ kg/m³, $U = (0, 0)$). The simulations are run for twice the time taken by the moving shock to cross the complete domain. A fifth-order WENO-LLF scheme is used for spatial discretization of the conserved variables. A third-order TVD Runge–Kutta explicit scheme is used for the temporal integration of the governing equations. A mesh resolution of 1500×1000 is considered. After the completion of each simulation, 40 snapshots with 176×112 cells are collected at equal time and spatial intervals to train the model.

We selected simulations with varying Mach numbers between $1.6 \leq M_s \leq 4.4$ as the training set, except for Mach numbers 2.0, 2.5, 3.0, 3.5, and 4.0 which were left out as the testing set. These left-out test samples are used to assess the capability of PARC in interpolating

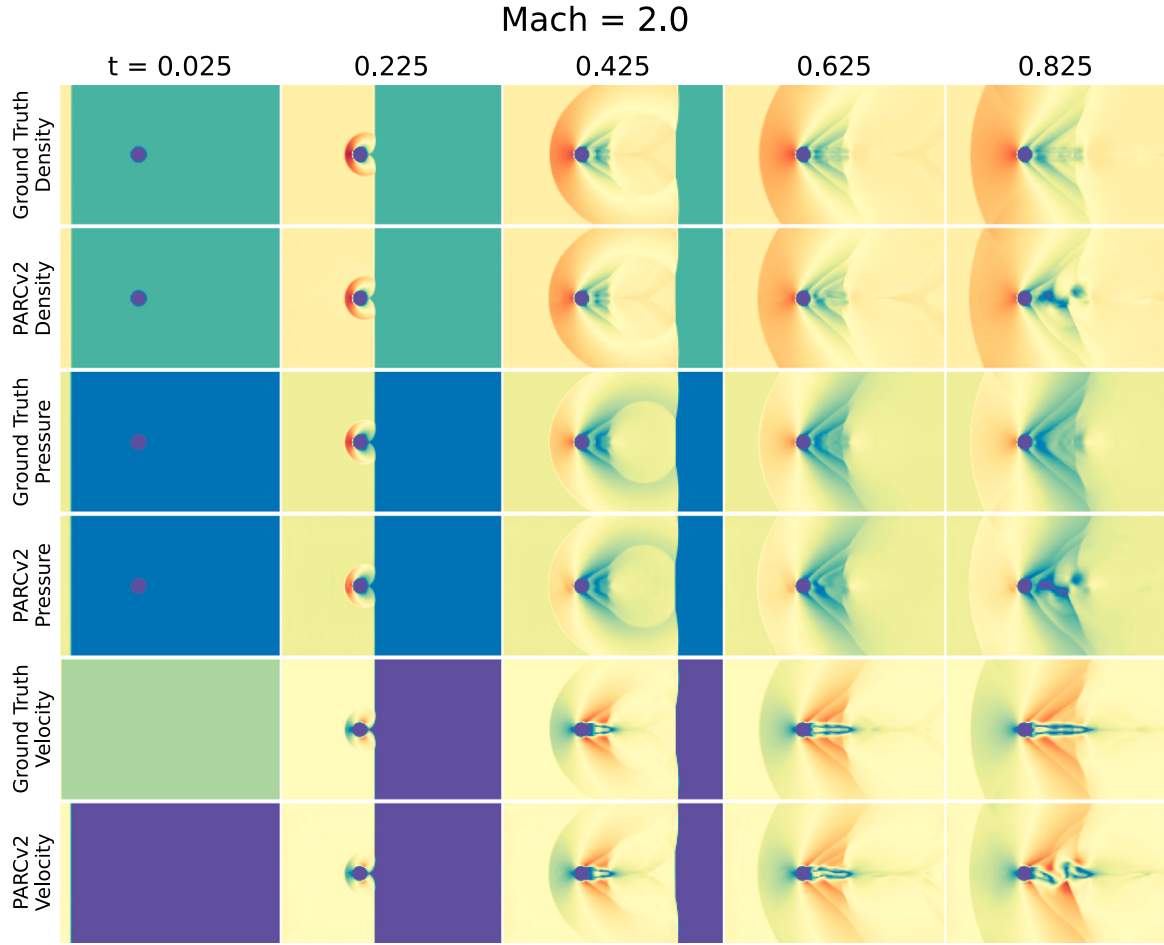


Fig. 7. Ground truth and predicted simulation of Mach number 2.0. While PARCv2 predicted the beginning of the simulation and the formation of bow shock very accurately, the prediction of the evolution of the wake flow at the end of the simulation shows a significant difference.

fluid behaviors for Mach numbers not encountered during training: the objective is to evaluate PARC's ability to predict fluid behaviors in an unseen speed regime based on its exposure to previously observed fluid speed conditions. Additionally, simulations with Mach number $M_s < 1.6$ or $M_s > 4.4$ were reserved for testing how well PARC can extrapolate the results to unseen Mach numbers outside of the training set range.

We treated density and pressure as state variables (i.e. components of $\mathbf{x}$ in Eq. (1) and Eq. (2)), and the two velocity components as such, i.e. components of $\mathbf{u}$ in Eq. (1) and Eq. (2). The differentiator was constructed to explicitly include advection and diffusion terms of density, only advection terms for pressure and velocity components, and reaction terms. We trained the differentiator for 1,250 epochs at a constant learning rate of 10^{-5} , with RK4 integrator and MSE loss function. No data-driven integrator was used in this experiment, as our attempts show that data-driven integration does not result in any significant changes in the prediction quality.

4.1. Mach number within training set coverage

We first examined the performance of the models on Mach numbers that are within the domain of the training set, namely test cases with Mach numbers $1.5 < M_s < 4.5$. From the shaded regions in Fig. 6, we notice that RMSE loss increases with a larger Mach number regardless of the integrator, but the relative RMSE value does not show such a trend. This is an indication that the increase in RMSE can be entirely attributed to the larger density, pressure, and velocity values at higher Mach numbers, and the models are capable of producing consistently satisfactory results across a wide range of Mach numbers.

Comparing the model using the RK4 integrator against the one using the RK5 integrator, it is evident that the model with the RK4 integrator produces smaller RMSEs regardless of the physical quantity being predicted or the Mach number. While one would expect a higher-order integrator to produce a smaller error, a higher-order integrator also requires calling the differentiator more times, potentially introducing more difficulties, such as vanishing gradients during backpropagation in the training process. Therefore, the expected less numerical error in integration might be canceled out by the increased error in differentiator.

Interestingly, we noticed that the predicted Mach 2 sequence has the highest relative RMSE velocity, especially for RK4 integrator, and the trend of increasing RMSE continues down to lower Mach number cases. Visual inspection of Mach 2.0 prediction and ground truth simulation reveals that while PARCv2 predicted the beginning of the simulation and the formation of the bow shock accurately, the evolution of the wake flow at the end of the simulation (last row of Fig. 7) shows a significant difference between the ground truth simulation, where the flow is smooth and steady, and the PARC prediction, where the flow is unsteady, leading to instabilities and formation of spurious vorticity.

Visual inspection of Mach 3 (Fig. 8) and 4 (Fig. 9) does not reveal the same phenomenon, but provides additional insights into the behavior at Mach 2. We noticed that vortex formation and shedding, a phenomenon that only occurs when the Mach number is greater than 2.1 in the ground truth simulation, is predicted to occur at Mach 2. Considering that a change in the physics regime happens at Mach 2.1, with a bifurcation from steady to unsteady flows, we conclude that PARCv2 learned the dynamics in high Mach numbers to a high degree

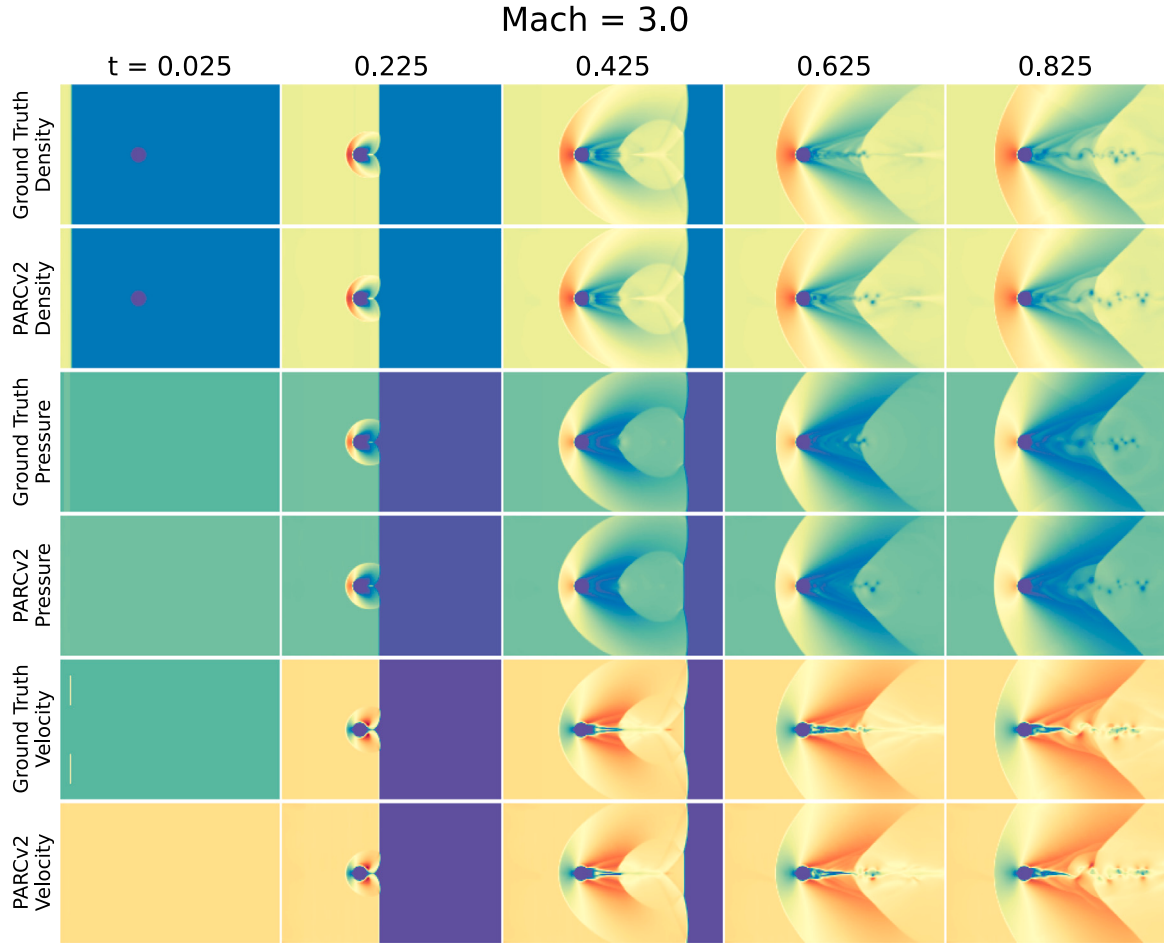


Fig. 8. Ground truth and predicted simulation of Mach number 3.0. The predicted bow shock is visually indistinguishable from ground truth. While the predicted wake flow is less sharp than that from ground truth, the overall shape as well as vortex formation and shedding is present.

of accuracy but may have failed to learn the dynamics in lower Mach numbers. Examination of lower Mach numbers beyond the coverage of the training set is likely to produce more insight, which will be covered in subsequent subsections.

Meanwhile, one physical phenomenon that is vital in this experiment is the creation and subsequent evolution of a bow shock. In dimensionless time, the incoming flow impinges on the cylinder at $t^* = 1/5$ (the 8th frame of simulation) regardless of the speed of the flow, and the bow shock is formed at that instance. However, for low Mach numbers, the bow shock is reflected by the cylinder and propagates in the direction of the incoming flow, while in the high Mach number cases ($M_s > 2.1$) the bow shock achieves a steady standoff distance from the cylinder. Therefore, we traced the position of the shock front from its formation and plotted the dimensionless distance of the shock front to the center of the cylinder with dimensionless time in Fig. 10. Overall, the predicted simulations show similar bow shock evolution to those in the corresponding ground truth over the entire Mach number range. As established before, predictions at Mach 2 exhibit the largest deviation from the ground truth. Furthermore, it is noticeable that PARCv2 models consistently over-predict the distance of the bow shock by a small amount, even at the time of shock formation. The effect of the integrator on the evolution of the bow shock is not prominent, as can be seen by the largely overlapped dashed (RK4) and dotted (RK5) lines regardless of Mach number. Previous application of PARCv2 for prediction of shock dynamics in porous energetic materials reveals that the model tends to under-predict sharp changes (Nguyen et al., 2023a, 2024), and we suspect the same phenomenon is also present here.

4.2. Extrapolation to higher mach numbers

We further examine the ability of PARCv2 to predict flows at higher Mach number cases ($M_s > 4.5$), where we no longer have any training set coverage. We noticed a sharp increase in both absolute and relative RMSE in Fig. 6 to the right of the shaded area (training set coverage). This indicates the model struggles in predicting unseen flow speed scenarios more in the high Mach number regime. Nevertheless, for PARCv2 with RK4 integrator, the RMSE is small enough to produce visually indistinguishable and numerically stable simulations as shown in Fig. 11 for Mach 4.5 and Fig. 12 for Mach 5. We do notice that the predicted vortex in the later stages of the simulation at Mach 5 (last column of Fig. 12) is stronger and very similar to Mach 4 and Mach 4.5, providing potential evidence that the model tends to deviate from the governing physics when extrapolated farther enough into unseen territories.

Comparison between the two integrators provides further support to the previously discussed disadvantage of using higher-order integrators with PARCv2. RK5 model shows a much larger increase in RMSE than RK4 model, as can be seen in Fig. 6 where the dashed lines are still above the solid lines on the right side of the shaded area. Furthermore, a visual inspection of the RK5 model shows numerical instability in the later stages of the simulation. While, in principle, higher-order integrators are more numerically resilient to error accumulation, the added number of calls to differentiator per integration step and subsequent increased difficulty in training due to this increased number of recursions likely offsets the advantage of RK5 versus RK4.

This numerical instability can also be seen in Fig. 10, where the RK5 prediction at Mach 5 (yellow dotted line) starts to decrease while both

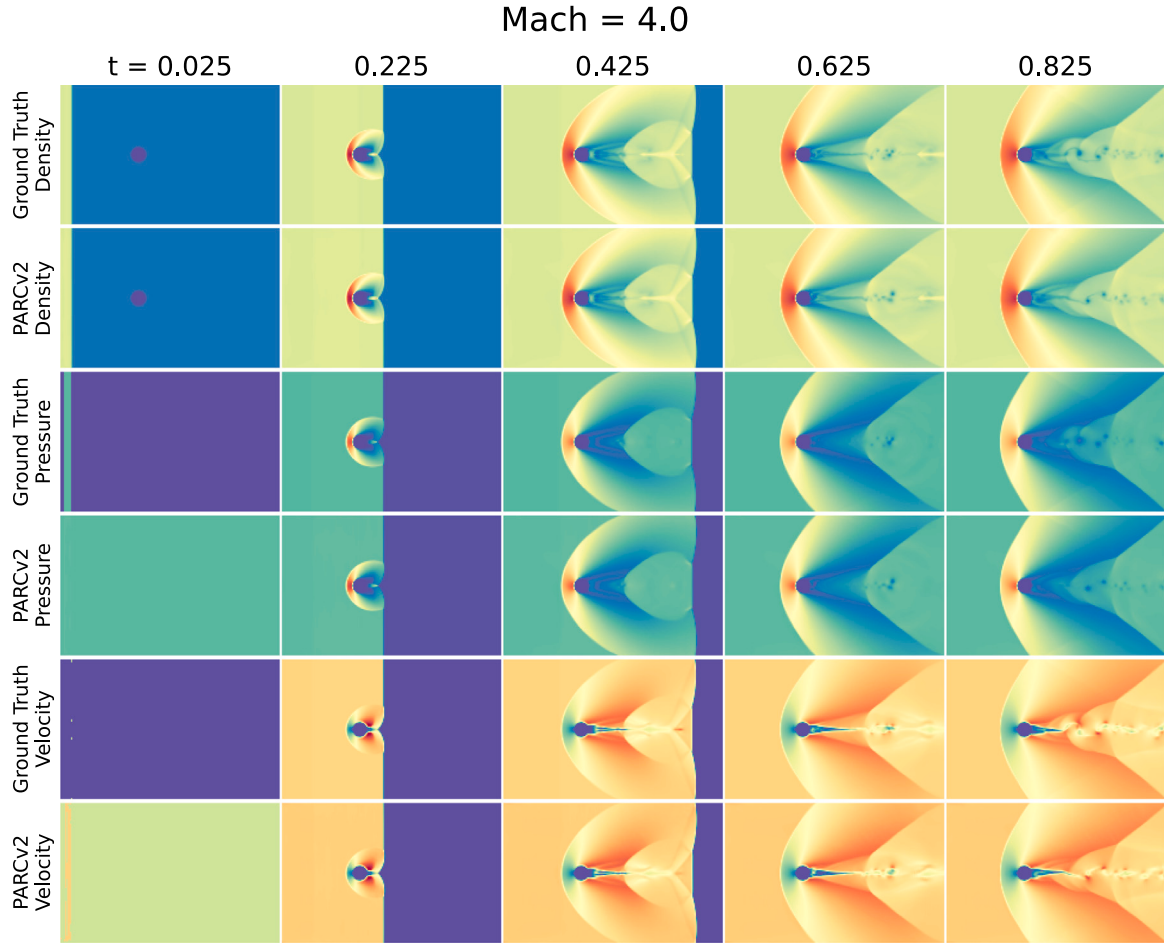


Fig. 9. Ground truth and predicted simulation of Mach number 4.0. Compared to Mach number 3.0, PARCv2 predicted simulation is even closer to the ground truth.

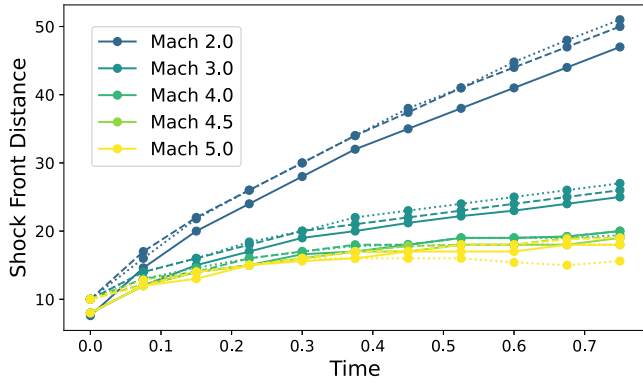


Fig. 10. Dimensionless distance of the shock front from the center of the cylinder as a function of dimensionless time at given Mach numbers. The color indicates the Mach number and the line style stands for the ground truth (solid), the PARCv2 model with the RK4 integrator (dashed), and the model with the RK5 integrator (dotted). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the ground truth (yellow solid line) and the RK4 prediction (yellow dashed line) stabilizes at around a value of 15.

4.3. Extrapolation to lower mach numbers

While the test cases with training set coverage and extrapolation to higher Mach number cases produce predicted sequences that are

close to the ground truth simulations with the same initial condition, we discovered that the PARCv2 model struggles with extrapolation in lower Mach number cases $M_s < 1.5$. From the comparison of RMSE loss in Fig. 6, increases are seen in both the absolute and relative RMSE to the left of the shaded area where the low Mach number extrapolation cases are depicted. As mentioned in Section 4.1, quantitative and visual differences between the ground truth and predicted sequences have already shown up in the Mach 2 test case. This trend in deteriorating performance is confirmed by further inspection of Mach 1.2 (Fig. 14) and Mach 1.5 (Fig. 13) which reveals almost a complete failure of PARCv2 in the prediction of low Mach number extrapolation regimes. While PARCv2 is still capable of handling the simple case of traveling of the fluid inflow at the earliest stages of the simulation (first column of Figs. 13 and 14), it is incapable of reproducing the formation and evolution of bow shock (second, third and fourth column of Figs. 13 and 14) or the stable wake flow behind the cylinder at later stages of the simulation (last row of Figs. 13 and 14). One can also observe that the Mach 1.2 case shows the worst predicted sequence by either the highest RMSE loss, especially relative value in Fig. 6, or the lack of any expected patterns in the predicted sequence in Fig. 14. As such, tracing of the bow shock front becomes impossible in these low Mach number cases and thus these cases are not present in Fig. 10.

We surmise three possible reasons for such a failure. First, the values of density, pressure, and velocity at lower Mach number cases are smaller, thus incurring a smaller penalty in the loss function. As neural networks tend to capture the most prominent patterns first and more details and smaller deviations from ground truth later during training, it is likely that capturing the patterns in lower Mach number cases would take considerably longer epochs with very little decrease in the

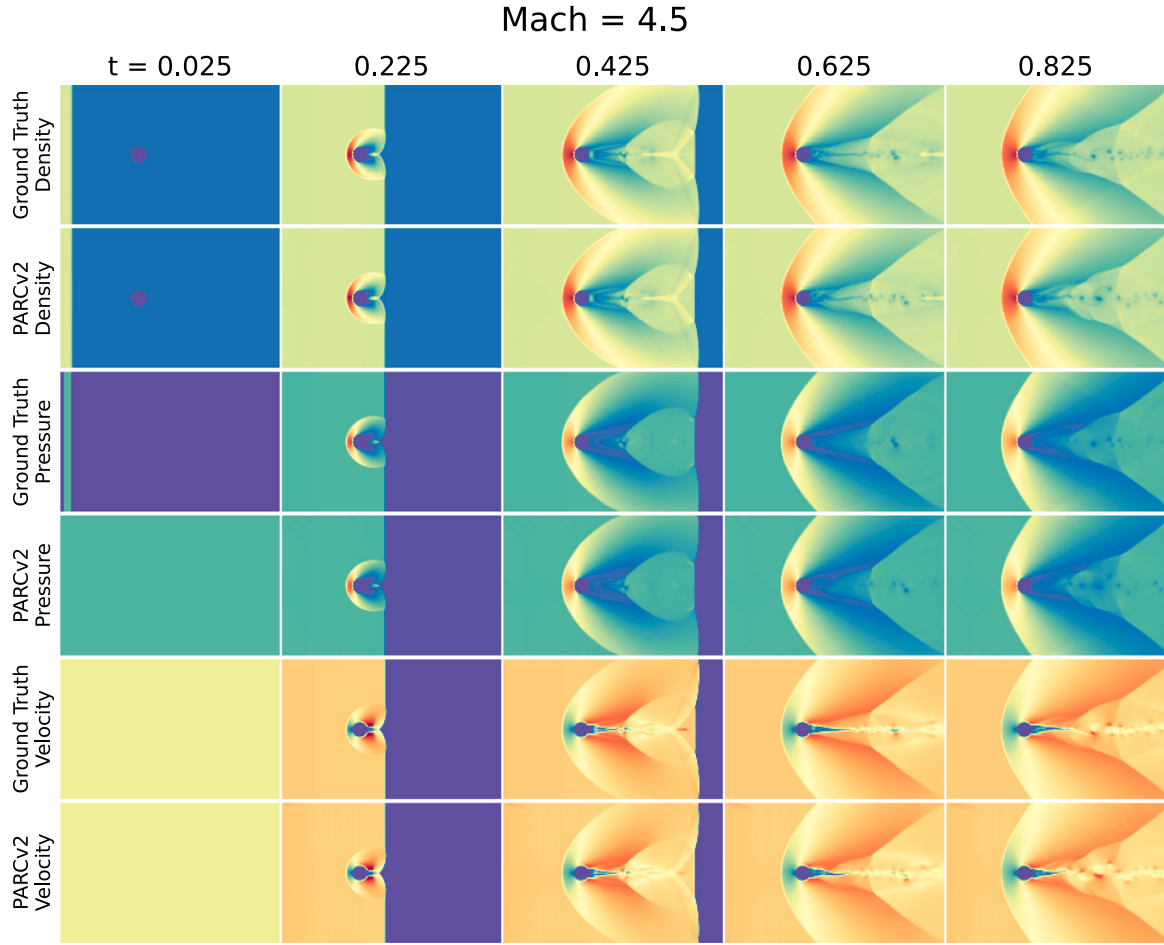


Fig. 11. Ground truth and predicted simulation at Mach 4.5. PARCv2 predicted simulation still captures the bow shock, its evolution with time and the vortex behind the cylinder.

average loss value. As such, the stopping criteria will be triggered long before convergence at lower Mach number cases. Second, there are more simulations beyond Mach 2.1 in our training set, and since each simulation was seen by the model only once per epoch, the model is biased towards high Mach number cases during training. As the training cases are equally spaced in Mach number, we have 5 training cases with Mach 2.1 and lower, and more than 15 cases above. Third, there is a distinct bifurcation of the overall dynamics of the system at Mach 2.1; the flow patterns at the lower and higher Mach number cases are quite different. Mach 2.1 marks the point of transition in post-shock flow from subsonic to supersonic. As such, the dynamics of the flow changes from acoustic signals propagating upstream to only the shock waves propagating upstream. There the low Mach number cases, especially in the upstream part of the domain, consist of substantially more complex flow dynamics. A close examination of ground truth simulation shows two major differences. At the lower Mach numbers, the transient bow shock travels far upstream, eventually exiting the simulation domain and the wake flow behind the cylinder is smooth and steady with no vortical patterns in the wake. On the other hand, higher Mach number cases are markedly different, with the bow shock reaching a stable distance from the cylinder, and the flow behind the cylinder develops unsteady vortical patterns. It is likely that the neural network only learned one regime but failed to learn another. This is supported by the results of Mach 2 (Fig. 7), where as previously mentioned the predicted sequence shows signs of vortical flow while the ground truth does not. We propose that an intentional bias towards lower Mach number cases by either increasing the number of such simulations in the training set

through generating more training samples or training on such cases more than once within each epoch or finetuning the model on only the lower Mach number cases after training on the entire training set might mitigate this issue. This is being explored in current work.

4.4. Extrapolation in time

To examine the performance of PARCv2 in extrapolation in time, we selected 3 cases—Mach numbers 2, 3.5 and 5, and predicted the flow evolution through $t^* = 2$, which is double the simulation duration of the training data. These 3 specific Mach number cases were selected because they cover a wide range of initial conditions, both within and outside the coverage of the training set. No cases with Mach numbers $M_s < 2$ were chosen since PARCv2 already struggled with $0 < t^* < 1$.

In Fig. 18, RMSE as a function of simulation time is presented. We noticed that both Mach 3.5 and Mach 5 cases do not show any significant increase in RMSE during the extrapolation. Visual examination of predicted simulation snapshots in Figs. 15 and 16 confirmed this finding, as the predictions remain close to the ground truth. PARCv2 successfully predicted the oscillatory vorticity pattern in the wake flow. We also noticed that the vorticity becomes less extreme and closer to the ground truth simulation as time goes on in the Mach 5 case. One interesting phenomenon is that there are significant decreases in RMSE regardless of variables at $t^* = 0.5$ for Mach 2 and Mach 3.5 cases. After checking the snapshots, we discovered that this is also the exact time where the shock waves exit the simulation domain. Therefore, we hypothesize that PARCv2 tends to have a larger prediction error when

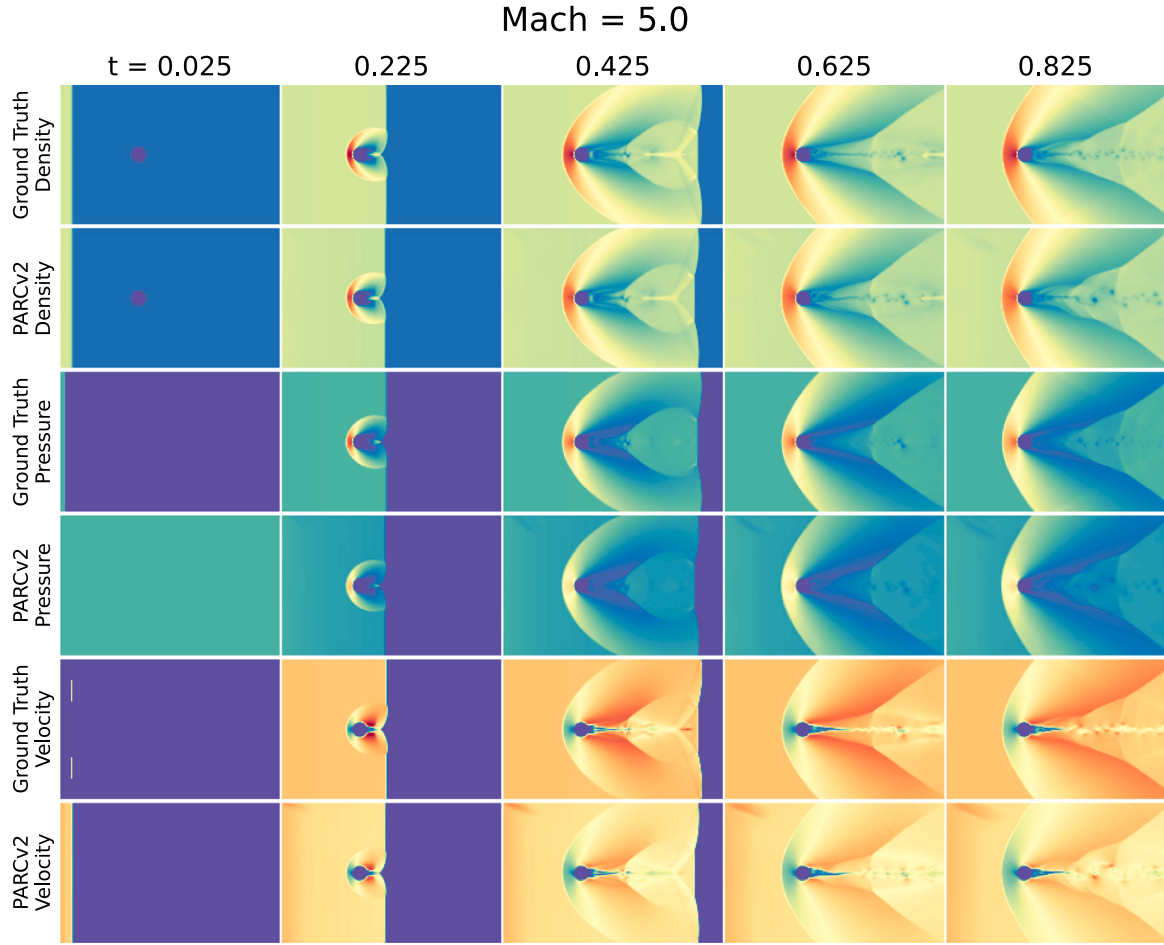


Fig. 12. Ground truth and predicted simulation at Mach 5. PARCv2 predicted simulation still captures the bow shock, its evolution with time and the vortex behind the cylinder, but the predicted vortex is more extreme and similar to those lower Mach number cases.

predicting sharp features, a phenomenon observed in the energetic material experiment in [Nguyen et al. \(2024\)](#).

As expected, the Mach 2 case shows significantly worse performance during extrapolation in time (see [Figs. 17 and 18](#)). Instead of RMSE stabilizing around a certain value, the RMSE values continue their upward trends into the extrapolation in time and reach a higher value than the case of Mach 3.5, despite the latter case having larger values in density, pressure, and velocity. After the bow shock exits in the simulation domain in the direction of the inflow, PARCv2 prediction becomes unstable and starts to show large-scale artifacts.

5. Discussion

Here we provided a comparison of prediction time per simulation between PARCv2, other models and DNS solver in the three experiments presented in this work (see [Table 3](#)). Compared to the DNS solver, PARCv2 is several orders of magnitude faster during prediction across all three experiments. Compared to other ML/PIML models, PARCv2 exhibits clear advantage in prediction time of more complex problems (e.g. Navier–Stokes), while in simpler problems (e.g. Burger’s) its prediction time is comparable to FNO and PIFNO. Note that FNO and PIFNO have the same architecture thus the same prediction time.

The guiding philosophy of PARCv2 entails the emulation of its network architecture to mirror the governing PDEs. Specifically, in this paper, PARCv2 replicated the general transport physics PDEs, encompassing both incompressible and compressible variants, characterized by unsteady convection–diffusion–reaction formulations. Note that no

Table 3

A comparison of prediction time for one simulation between PARCv2, other PIML models and DNS solver. Measurements of ML model prediction time were performed on one Nvidia A100 GPU, and are provided in unit of seconds in the table. Measurements of DNS running time were performed on computing clusters with 2.6 GHz AMD 7H12 Rome, and are provided in unit of CPU core hours in the table.

Prediction time per simulation	Burger	N-S	Supersonic
PhyCRNet (sec)	0.264	0.713	–
FNO (sec)	0.433	0.939	–
PIFNO (sec)	0.433	0.939	–
PARCv2 (sec)	0.445	0.278	11.0 (s)
DNS (core hours)	0.14	~ 12	~ 512

changes to the PARCv2 architecture or training/loss modalities were made for any of the cases simulated in the paper. This is a departure from the conventional approach of embedding physics PDEs directly into loss functions and offers distinct advantages, as can be evidenced from the results in this paper.

For example, one of the discoveries from our experiments was the tension between prediction accuracy (RMSE) and solution quality (PDE residuals). This raises an intriguing question about the contrasting learning philosophies between PARC and PINNs. While physics-informed regularization in PINNs can be seen as a soft constraint

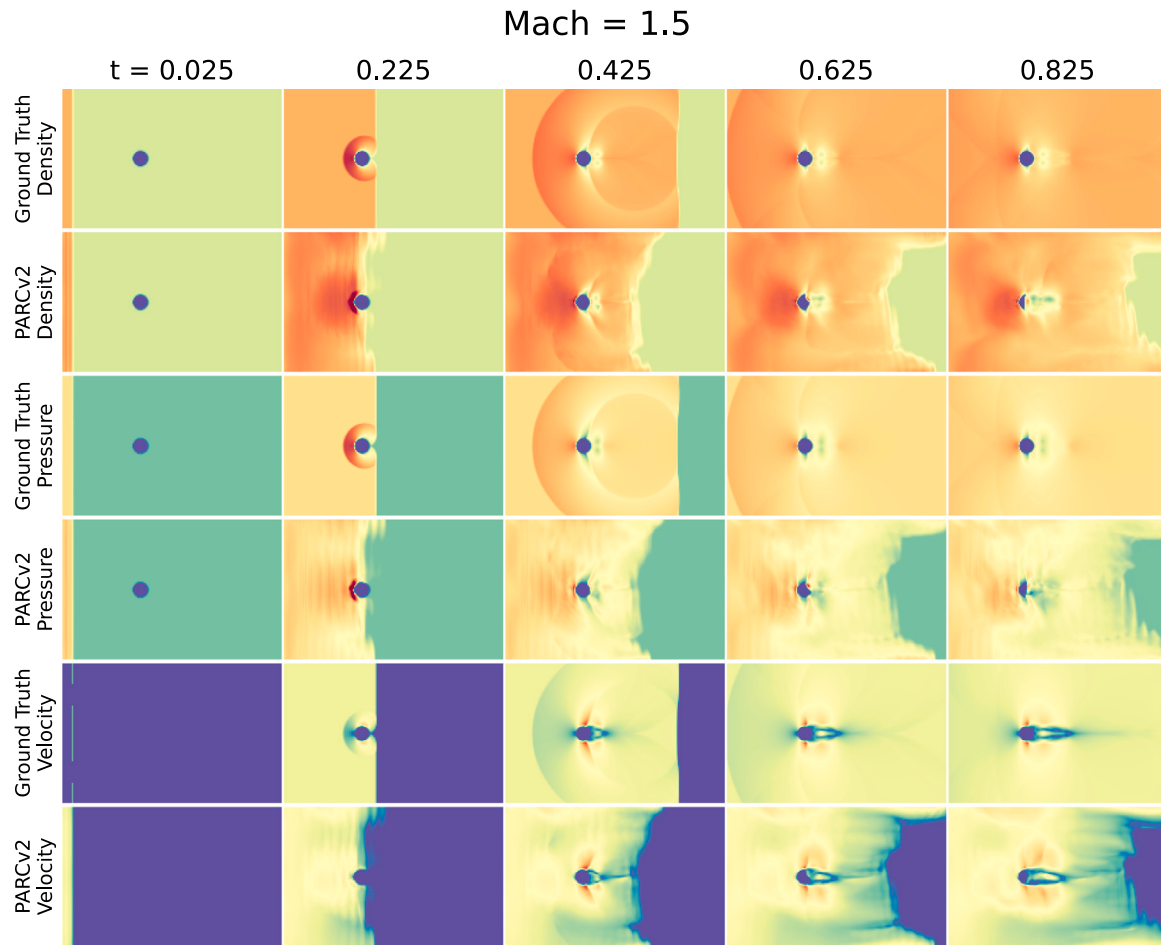


Fig. 13. Ground truth and predicted simulation at Mach 1.5. PARCv2 failed to predict a reasonable sequence and large-scale differences can be visually identified even without any quantitative comparison.

guiding the network towards specific regions in the search space, PARC's inductive bias approach imposes a hard constraint on the solution function's topology in the search space. Moreover, as physics constraints are integrated into the architecture design, training PARC becomes a single-objective optimization problem, unlike the multi-objective optimization problem presented by PINN. This complexity may make the loss landscape of PINN challenging to navigate, with a vast search space (the 'space of all functions') and conflicting objectives (Krishnapriyan et al., 2021). In contrast, PARC operates within a smaller search space due to architectural constraints on function topology, eliminating the risk of conflicting objectives and facilitating convergence.

The performance of PINN tends to be sensitive to the selection of sampling methods and the weighting of loss terms. In fact, one may argue that the comparisons in this present work are not entirely fair since the sampling methods and weighting strategies for PhyCRNet and PI-FNO were not meticulously tailored to optimize the predictions for each specific problem. While it is true that with different sampling methods and weighting schemes, the performance of PhyCRNet and PI-FNO could have been improved, this underscores an issue of the need to experiment with various hyperparameters and ad-hoc processes during PINN training. In contrast, PARC requires less tuning of settings to be effective across different dynamics regimes.

Furthermore, the primary challenge in data-driven modeling resides in determining the pertinent variables for learning. In solving different systems governed by different sets of governing equations, such as the Burgers' and Navier–Stokes equations, the selection of variables is

straightforward due to the limited number of equations and variables. However, in instances where the number of equations and variables expand significantly, discerning the variables to be learned becomes imperative. This problem is frequently encountered in reactive multiphase flow simulations yet PARC has the inherent adaptability to accommodate incompressible and compressible flow regimes without necessitating structural alterations, manipulation of the weighting of loss terms, and adaptive sampling mitigates this issue. In addition, PARC exhibits versatility even in scenarios where PDEs are elusive by solely functioning on available data.

6. Conclusion and future work

We presented an inductive bias physics-aware deep learning method, PARCv2, capable of simulating compressible multiphase flows. PARCv2 features high prediction accuracy and acceptable physics awareness when compared to other state-of-the-art PIML models on subsonic flow problems. Additionally, experiments on supersonic compressible flow demonstrated that PARCv2 can capture the bow shock and vortical flow in supersonic flows, with a caveat that it may quickly deteriorate in the low Mach number regime.

For future research, different loss functions and weighting schemes may be worthwhile to explore. What we find intriguing from the supersonic flow experiment in Section 4 was the fact that, while capable of modeling sharp gradients with large values in the high Mach number regime, PARC struggled in modeling low Mach number flows, rendering an opposite behavior to other physics-informed models. We suspect that

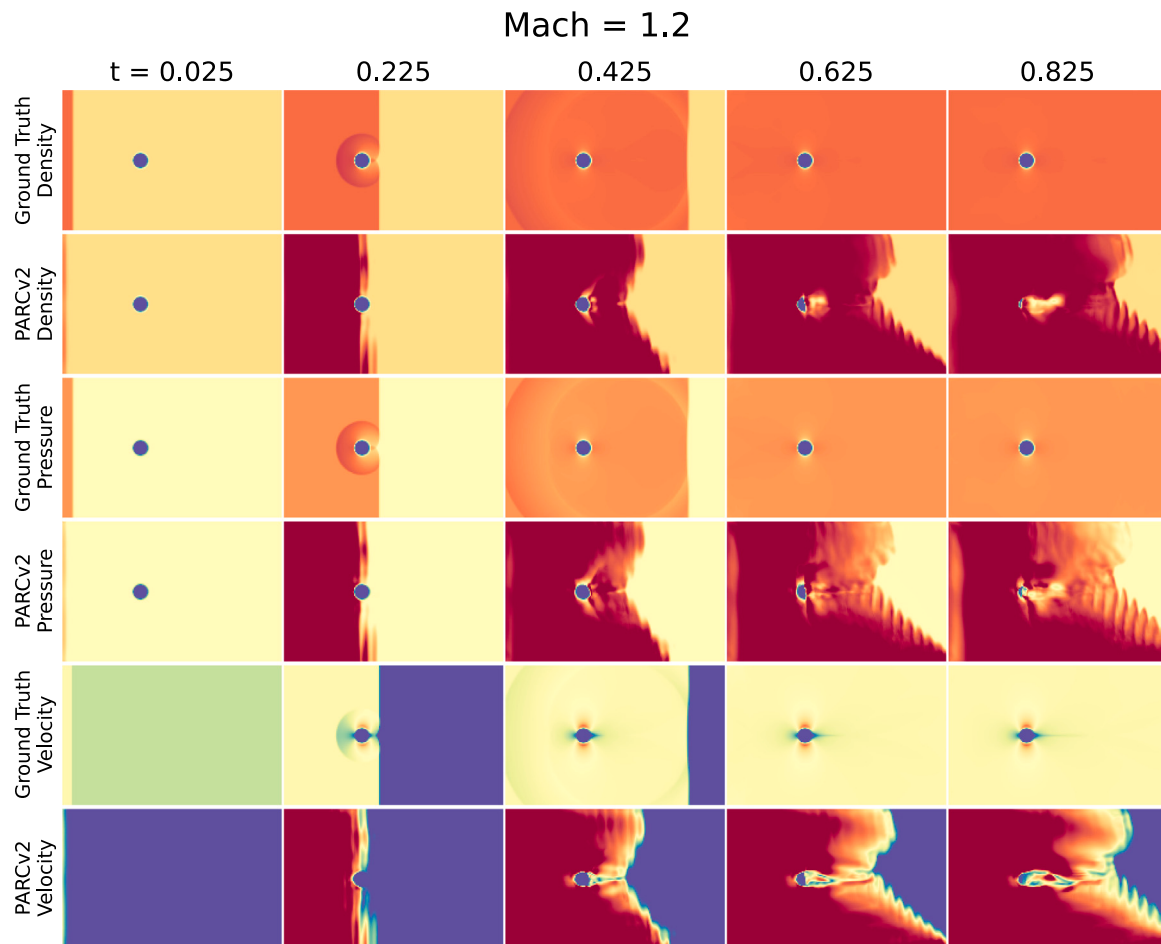


Fig. 14. Ground truth and predicted simulation at Mach 1.2. PARCv2 failed to predict a reasonable sequence and large-scale differences can be visually identified even without any quantitative comparison.

the current loss function design is inherently biased toward data points with large values (e.g., high Mach number flows). To this end, gradient-based loss weighting (Liu et al., 2024) or feature-based weighting (Lu et al., 2023) may be introduced to augment PARC training.

Furthermore, an adaptive resolution might be implemented in PARC to resolve sharp and discontinuous features more effectively. We are currently exploring techniques such as continuous convolution (Shocher et al., 2020) as the means to attain adaptive resolution in PARC. The key idea here is that, instead of learning filters on a fixed grid of pixels, we may define convolution operations as continuous functions, so that they can be resolved in different resolutions across the domain.

CRediT authorship contribution statement

Xinlun Cheng: Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis. **Phong C.H. Nguyen:** Writing – review & editing, Validation, Methodology, Investigation, Formal analysis. **Pradeep K. Seshadri:** Writing – review & editing, Validation, Data curation. **Mayank Verma:** Writing – review & editing, Validation, Data curation. **Zoë J. Gray:** Writing – review & editing, Validation. **Jack T. Beerman:** Writing – review & editing, Software. **H.S. Udaykumar:** Writing – review & editing, Supervision, Funding acquisition, Conceptualization. **Stephen S. Baek:** Writing – review &

editing, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

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.

Data availability

Data will be made available on request.

Acknowledgments

We appreciate the anonymous reviewers for their comments on improving this work. This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and was supported by the LLNL-LDRD Program under Project No. 24-SI-004. This work also was partially supported by the Army Research Office (ARO) Energetics Basic Research Center (EBRC) program under Grant No. W911NF-22-2-0164 and the National Science Foundation under Grant No. DMREF-2203580.

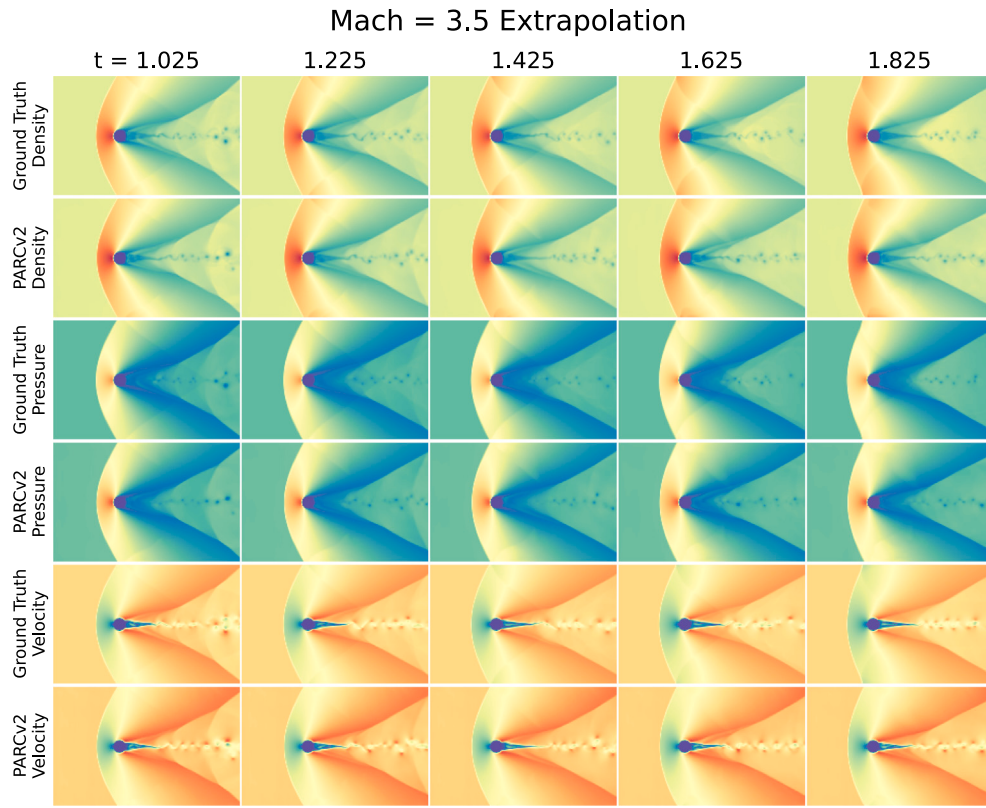


Fig. 15. Ground truth and predicted simulation at Mach 3.5 beyond $t^* = 1$. Visual inspection did not reveal any decline in prediction accuracy when extrapolating in time beyond the coverage of the training set. Oscillatory wake flow patterns were reproduced by PARCV2.

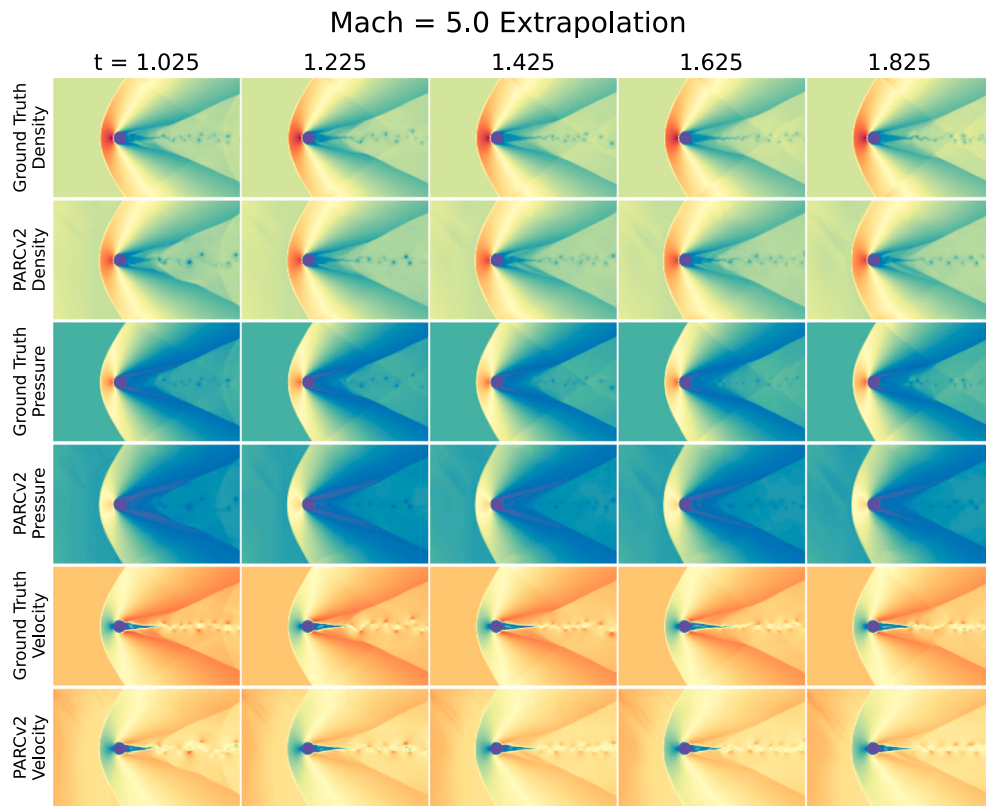


Fig. 16. Ground truth and predicted simulation at Mach 5.0 beyond $t^* = 1$. While the predicted simulation shows more extreme vorticity at the beginning, it gradually dies down during the extrapolation.

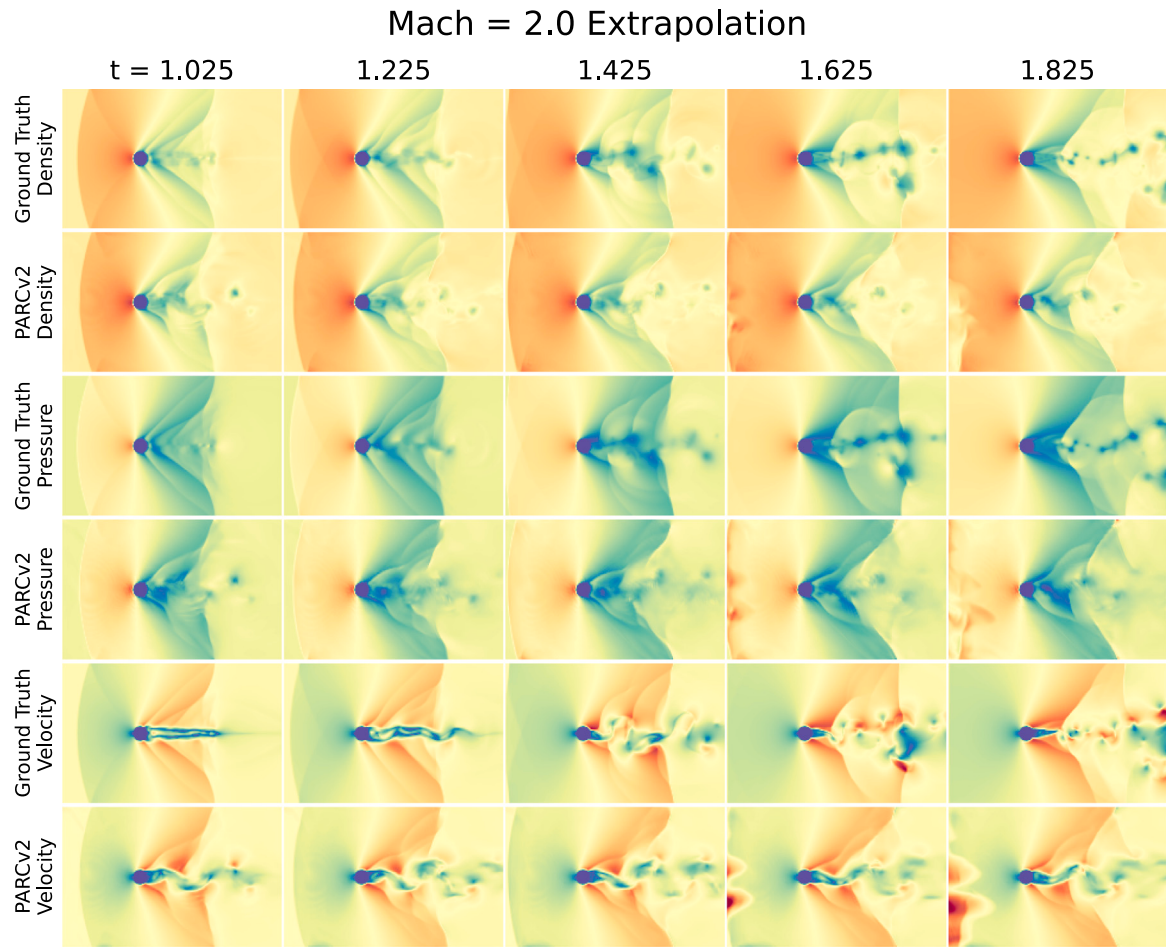


Fig. 17. Ground truth and predicted simulation at Mach 2.0 beyond $t^* = 1$. A gradual deterioration in PARCv2 performance can be observed when extrapolating in time beyond training set coverage at low mach number cases, especially after the bow shock exited the domain.

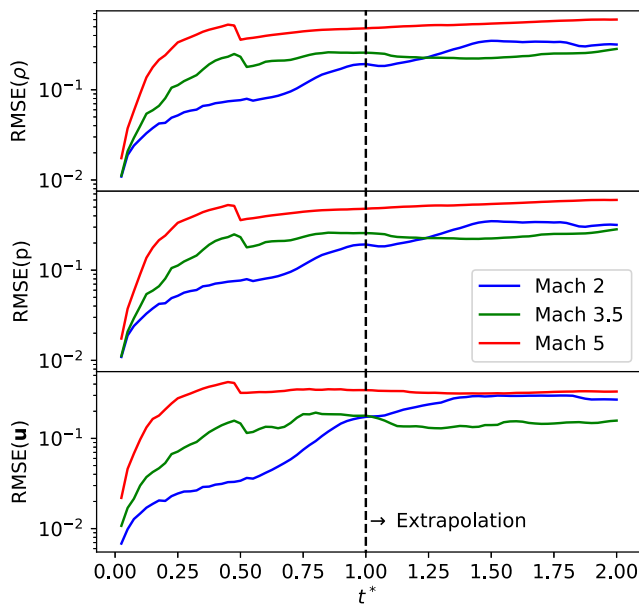


Fig. 18. RMSE as a function of non-dimensionalized time for Mach 2, 3.5, and 5 cases. The starting point of extrapolation in time is indicated with a dashed line.

References

- Anderson, J., 2003. Modern compressible flow: With historical perspective. In: *Aeronautical and Aerospace Engineering Series*, McGraw-Hill Education, URL <https://books.google.com/books?id=woeqa4-a5EgC>.
- Ansys Inc., 2023. Ansys fluent - Fluid Simulation Software. URL <https://www.ansys.com/products/fluids/ansys-fluent>. (Accessed: 26 February 2023).
- Awasthi, M., McCreton, S., Moreau, D., Doolan, C., 2022. Supersonic cylinder wake dynamics. *J. Fluid Mech.* 945, A4. <http://dx.doi.org/10.1017/jfm.2022.517>.
- Baydin, A.G., Pearlmutter, B.A., Radul, A.A., Siskind, J.M., 2018. Automatic differentiation in machine learning: a survey. *J. March. Learn. Res.* 18, 1–43.
- Cai, S., Mao, Z., Wang, Z., Yin, M., Karniadakis, G.E., 2022. Physics-informed neural networks (PINNs) for fluid mechanics: a review. *Acta Mech. Sinica* 37 (12), <http://dx.doi.org/10.1007/s10409-021-01148-1>.
- Das, P., Sen, O., Jacobs, G., Udaykumar, H.S., 2017. A sharp interface cartesian grid method for viscous simulation of shocked particle-laden flows. *Int. J. Comput. Fluid Dyn.* 31 (6–8), 269–291. <http://dx.doi.org/10.1080/10618562.2017.1351610>.
- De Palma, P., de Tullio, M., Pascasio, G., Napolitano, M., 2006. An immersed-boundary method for compressible viscous flows. *Comput. & Fluids* 35 (7), 693–702. <http://dx.doi.org/10.1016/j.compfluid.2006.01.004>, URL <https://www.sciencedirect.com/science/article/pii/S0045793006000065>.
- Jin, X., Cai, S., Li, H., Karniadakis, G.E., 2021. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *J. Comput. Phys.* 426, 109951. <http://dx.doi.org/10.1016/j.jcp.2020.109951>, URL <https://www.sciencedirect.com/science/article/pii/S0021999120307257>.
- Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L., 2021. Physics-informed machine learning. *Nat. Rev. Phys.* 3 (6), 422–440.
- Kingma, D.P., Ba, J., 2017. Adam: A method for stochastic optimization. *arXiv:1412.6980*.
- Krishnapriyan, A., Gholami, A., Zhe, S., Kirby, R., Mahoney, M.W., 2021. Characterizing possible failure modes in physics-informed neural networks. *Adv. Neural Inf. Process. Syst.* 34, 26548–26560.
- Li, Z., Kovachki, N., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A., Anandkumar, A., 2020. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*.

- Liu, L., Liu, S., Xie, H., Xiong, F., Yu, T., Xiao, M., Liu, L., Yong, H., 2024. Discontinuity computing using physics-informed neural networks. *J. Sci. Comput.* 98 (1), 22.
- Lou, Q., Meng, X., Karniadakis, G.E., 2021. Physics-informed neural networks for solving forward and inverse flow problems via the Boltzmann-BGK formulation. *J. Comput. Phys.* 447, 110676. <http://dx.doi.org/10.1016/j.jcp.2021.110676>, URL <https://www.sciencedirect.com/science/article/pii/S0021999121005714>.
- Lu, H., Nemani, V.P., Barzegar, V., Allen, C., Hu, C., Laflamme, S., Sarkar, S., Zimmerman, A.T., 2023. A physics-informed feature weighting method for bearing fault diagnostics. *Mech. Syst. Signal Process.* 191, 110171.
- Lucor, D., Agrawal, A., Sargent, A., 2022. Simple computational strategies for more effective physics-informed neural networks modeling of turbulent natural convection. *J. Comput. Phys.* 456, 111022. <http://dx.doi.org/10.1016/j.jcp.2022.111022>, URL <https://www.sciencedirect.com/science/article/pii/S0021999122000845>.
- Mao, Z., Jagtap, A.D., Karniadakis, G.E., 2020. Physics-informed neural networks for high-speed flows. *Comput. Methods Appl. Mech. Engrg.* 360, 112789. <http://dx.doi.org/10.1016/j.cma.2019.112789>, URL <https://www.sciencedirect.com/science/article/pii/S0045782519306814>.
- Nguyen, P.C.H., Cheng, X., Azarfar, S., Seshadri, P., Nguyen, Y.-T., Kim, M., Choi, S., Udaykumar, H., Baek, S.S., 2024. PARCv2: Physics-aware recurrent convolutional neural networks for spatiotemporal dynamics modeling. *arXiv preprint arXiv:2402.12503*.
- Nguyen, P.C., Nguyen, Y.-T., Choi, J.B., Seshadri, P.K., Udaykumar, H., Baek, S., 2023a. PARC: Physics-aware recurrent convolutional neural networks to assimilate meso scale reactive mechanics of energetic materials. *Sci. Adv.* 9 (17), eadd6868. <http://dx.doi.org/10.1126/sciadv.add6868>.
- Nguyen, P.C., Nguyen, Y.-T., Seshadri, P.K., Choi, J.B., Udaykumar, H., Baek, S., 2023b. A physics-aware deep learning model for energy localization in multiscale shock-to-detonation simulations of heterogeneous energetic materials. *Propellants Explosives Pyrotechn.* 48 (4), e202200268. <http://dx.doi.org/10.1002/prop.202200268>.
- Raissi, M., Perdikaris, P., Karniadakis, G., 2019a. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378, 686–707. <http://dx.doi.org/10.1016/j.jcp.2018.10.045>, URL <https://www.sciencedirect.com/science/article/pii/S0021999118307125>.
- Raissi, M., Wang, Z., Triantafyllou, M.S., Karniadakis, G.E., 2019b. Deep learning of vortex-induced vibrations. *J. Fluid Mech.* 861, 119–137. <http://dx.doi.org/10.1017/jfm.2018.872>.
- Ren, P., Rao, C., Liu, Y., Wang, J.-X., Sun, H., 2022. PhyCRNet: Physics-informed convolutional-recurrent network for solving spatiotemporal PDEs. *Comput. Methods Appl. Mech. Engrg.* 389, 114399. <http://dx.doi.org/10.1016/j.cma.2021.114399>, URL <https://www.sciencedirect.com/science/article/pii/S0045782521006514>.
- Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Springer International Publishing, Cham, pp. 234–241.
- Shocher, A., Feinstein, B., Haim, N., Irani, M., 2020. From discrete to continuous convolution layers. *arXiv preprint arXiv:2006.11120*.
- Takahashi, S., Nonomura, T., Fukuda, K., 2014. A numerical scheme based on an immersed boundary method for compressible turbulent flows with shocks: Application to two-dimensional flows around cylinders. *J. Appl. Math.* 2014 (SI02), 1–21. <http://dx.doi.org/10.1155/2014/252478>.
- Turner, J.M., Seo, J.H., Mittal, R., 2024. A high-order sharp-interface immersed boundary solver for high-speed flows. *J. Comput. Phys.* 500, 112748. <http://dx.doi.org/10.1016/j.jcp.2023.112748>, URL <https://www.sciencedirect.com/science/article/pii/S0021999123008446>.
- Wang, S., Teng, Y., Perdikaris, P., 2021a. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J. Sci. Comput.* 43 (5), A3055–A3081.
- Wang, S., Wang, H., Perdikaris, P., 2021b. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Comput. Methods Appl. Mech. Engrg.* 384, 113938. <http://dx.doi.org/10.1016/j.cma.2021.113938>, URL <https://www.sciencedirect.com/science/article/pii/S0045782521002759>.
- Wang, S., Yu, X., Perdikaris, P., 2022. When and why PINNs fail to train: A neural tangent kernel perspective. *J. Comput. Phys.* 449, 110768.
- Yang, X.I.A., Zafar, S., Wang, J.-X., Xiao, H., 2019. Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Phys. Rev. Fluids* 4, 034602. <http://dx.doi.org/10.1103/PhysRevFluids.4.034602>, URL <https://link.aps.org/doi/10.1103/PhysRevFluids.4.034602>.
- Yin, M., Zheng, X., Humphrey, J.D., Karniadakis, G.E., 2021. Non-invasive inference of thrombus material properties with physics-informed neural networks. *Comput. Methods Appl. Mech. Engrg.* 375, 113603. <http://dx.doi.org/10.1016/j.cma.2020.113603>, URL <https://www.sciencedirect.com/science/article/pii/S004578252030788X>.