INCOMPLETE TO COMPLETE MULTIPHYSICS FORECASTING - A HYBRID APPROACH FOR LEARNING UNKNOWN PHENOMENA

Anonymous authors
Paper under double-blind review

ABSTRACT

Modeling complex dynamical systems where only partial knowledge of their physical mechanisms is available is a crucial problem across all scientific and engineering disciplines. Purely data-driven approaches, which only make use of an artificial neural network and data, often fail to accurately simulate the evolution of the system dynamics over a sufficiently long time and in a physically consistent manner. Therefore, we propose a hybrid approach that uses a neural network model in combination with an incomplete PDE solver that provides known but incomplete physical information. In this study, we demonstrate that the results obtained from the incomplete PDEs can be efficiently corrected at every time step by the proposed hybrid neural network – PDE solver model so that the effect of the unknown physics present in the system is correctly accounted for. For validation purposes, the obtained simulations of the hybrid model are successfully compared against results coming from the complete set of PDEs describing the full physics of the considered system. We demonstrate the validity of the proposed approach on a reactive flow, an archetypal multi-physics system that combines fluid mechanics and chemistry, the latter being the physics considered unknown. Experiments are made on planar and Bunsen-type flames at various operating conditions. The hybrid neural-network PDE approach correctly models the flame evolution of the cases under study for significantly long time windows, yields improved generalization, and allows for larger simulation time steps.

1 INTRODUCTION

Modeling and forecasting of complex physical systems described by nonlinear partial differential equations (PDEs) are central to various domains with applications ranging from weather forecasting (Kalnay, 2003), design of airplane wings (Rhie & Chow, 1983), to material science (Wheeler et al., 1992). Typically, a chosen set of PDEs is solved iteratively until convergence of the solution. Modeling complex physical dynamics requires a good understanding of the underlying physical phenomena. For cases where the complete physics information is missing, deep learning models can be employed to complete the physical description when additional data of the system is available. Deep learning methods have shown promise to account for these unknown components of the system (Yin et al., 2021). We consider a set of partial differential equations with partially unknown physics represented. The corresponding PDE model for a general state ϕ is given by

$$\frac{\partial \phi}{\partial t} = \mathcal{P}_i(\phi, u\frac{\partial \phi}{\partial x}, \frac{\partial^2 \phi}{\partial x^2}, ...) + S_\phi, \tag{1}$$

where $\mathcal{P}_i$ models the known but incomplete PDE description, and S_ϕ represents the unknown physics.

Within the scope of this model, the influence of S_ϕ can lead to fundamentally different scenarios. A commonly targeted case is when the governing equations of the complete PDE description are computationally too expensive to solve, turbulence modeling in computational fluid dynamics (CFD) being a good example. In CFD, a spacial filtering is performed on the original governing PDEs. This step introduces unclosed terms in the model equations that correspond to unrepresented physics in equation 1 due to the effects of the filtered scales. This is a widely studied problem, where the use

of deep learning models is currently being explored (Lapeyre et al., 2019; Kochkov et al., 2021; Stachenfeld et al., 2022). In the following, we are targeting a more challenging problem, where increasing spacial resolution and/or reducing time-scales of the incomplete PDE solver does not lead to a converged full solution. Rather, the incomplete and complete PDEs produce drastically different solutions due to the unknown physics. The central learning objective is to correct this behavior and retrieve the evolution that would be obtained with the complete PDE description.

To the best of our knowledge, we are the first to leverage the framework of incomplete PDE solvers and neural networks (NNs) to account for the effects of an incomplete physics model. The NN has the goal to complete the PDE description, where the differences in complete and incomplete PDE solutions are beyond the effects of spacial and temporal scales. We showcase that combining the trained NN model with a differentiable solver for the incomplete PDE can faithfully reproduce physically accurate temporal solutions of the complete PDE solver with stable long-term rollouts.

We demonstrate the capabilities of our approach for real-world scenarios of complex reactive flows for which the underlying chemistry reactions are unknown. Reactive flow modeling has applications in numerous domains such as combustion processes in gas turbines (Lieuwen, 2012), climate modeling (Jacobson, 1999; Rolnick et al., 2022) and astrophysics simulations (Gamezo et al., 2003). Resolving the Navier-Stokes equations lies at the core of these problems, where additionally the transport of different species of relevance must also be accounted for, together with their production or consumption often following complex reaction mechanisms (Poinsot & Veynante, 2005). For chemically reacting flows, generation or consumption of multiple species via some chemical reaction are modeled using a net source term. It is a well-known fact that the incorporation of a detailed chemical kinetic mechanism in a reacting flow model can result in a stiff system of governing equations (Wanner & Hairer, 1996; Najm et al., 1998; Knio et al., 1999). We consider the reactive flow simulation to be the complete PDE description, while the non-reactive flow simulation represents the incomplete PDE basis, where the chemical kinetic mechanisms are collected in the unknown physics term of equation 1. Figure 1 shows a visual example of the fundamental differences in system dynamics that can be caused by unknown reaction terms. We showcase the effectiveness of our approach for different cases of planar 2D premixed methane-air flames, and the varying transient evolution of Bunsen-type flames. We show that the proposed approach can handle large domains with highly resolved flames, which is closer to the practical flame domains used in many industrial applications. Specifically we concentrate on training a NN model to correct the spatio-temporal effects of energy and species transport source terms. We show that in addition to recovering the desired solutions, this approach overcomes inherent problems of temporal stiffness due to the complex reaction mechanism.

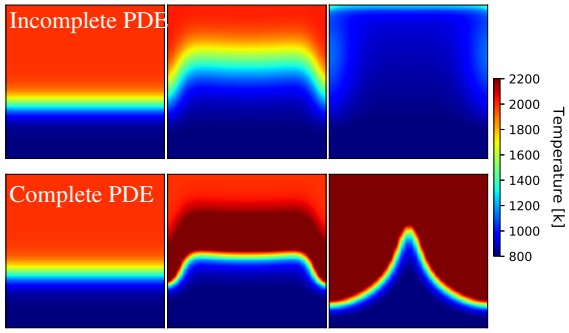

Figure 1: The incomplete/non-reactive (top) and complete/reactive (bottom) PDE solvers we consider can yield fundamentally different time evolutions, as shown here for a sample temperature field.

2 Background

Deep learning methods have been widely used to model the solutions of partial differential equations (Lagaris et al., 1998; Long et al., 2018; Han et al., 2018; Bar-Sinai et al., 2019) and in particular Navier-Stokes equations (Thuerey et al., 2020; Fukami et al., 2019). These models can be very fast and do not suffer from the timestep stability issues associated with traditional numerical solvers. Nevertheless, as these purely data-driven approaches lack the physical understanding of the system being modeled, they generally fail in generalizing to other operating conditions (Kim et al., 2019; Lapeyre et al., 2019). To leverage the potential of deep learning in physical simulations, it is therefore necessary to incorporate some physical information within the deep learning framework.

Deep learning models can enforce physical constraints partially through the loss function (Bar-Sinai et al., 2019; Raissi et al., 2019) or changes in neural network architecture (Greydanus et al., 2019; Lu et al., 2019). However, these approaches struggle to enforce physical constraints such as boundary conditions or conservation of momentum due to their nonlinear nature. Neural operators (Li et al., 2020b; Bhattacharya et al., 2020; Patel et al., 2021), specifically Fourier neural operator (FNO) Li et al. (2020a) introduce an interesting line of work by learning mesh-free, infinite-dimensional operators with neural networks, but do not generalize over longer simulation steps.

Hybrid approaches that combine machine learning techniques with numerical PDE solvers (Wang et al., 2020), have attracted a significant amount of interest due to their capabilities for generalization Chen et al. (2018). In this context, neural networks are typically used to model or replace a part of the conventional PDE solver to improve aspects of the solving process. For example, Tompson et al. (2017), Özbay et al. (2021) and Ajuria Illarramendi et al. (2020) proposed a convolutional neural network based approach to solve the Poisson equation in CFD simulation. In recent years, a number of deep learning based models have been introduced to accurately model turbulent flows (Pathak et al., 2020; Stachenfeld et al., 2022; Dresdner et al., 2022). Um et al. (2020) and Belbute-Peres et al. (2020) showed the advantages of training neural networks with differentiable physics to correct the numerical errors that arise in the discretization of PDEs. Sirignano et al. (2020) and Kochkov et al. (2021) additionally correct for closure error by integrating neural network with a differentiable CFD simulator. These approaches demonstrate the capabilities of neural networks to correct errors in fast, under-resolved simulation.

3 METHODOLOGY

We consider two different sets of PDEs with their associated numerical solver, which we denote as the *incomplete* PDE $\mathcal{P}_i$ and the *complete* PDE $\mathcal{P}_c$. By evaluating $\mathcal{P}_c$ on an input state ϕ_t at time t, we can compute the points of the phase space sequences; $\phi_{t+\Delta t} = \mathcal{P}_c(\phi_t)$. Without loss of generality, we assume a fixed time-step Δt and denote a state $\phi_{t+\Delta t}$ at next time instance as ϕ_{t+1}. The main goal of this study is to learn a correction function that models the effects of the unknown physical system on the incomplete PDE solver to obtain the complete PDE solutions. We train a correction model $C(\phi;\theta) : \mathcal{X} \rightarrow \mathcal{X}$ that maps $\mathcal{P}_i(\phi_t)$ to ϕ_{t+1}, defined over a finite time interval $[0, T]$ and a spacial domain $\Omega \subset \mathbb{R}^2$. The learning objective is to find the best possible correction function $C(\phi;\theta)$ given the weights θ and an input flow state, ϕ. The model parameters θ are estimated from the complete PDE solution trajectories $(\phi_0, \phi_1, .., \phi_T)$. The learned predictions obtained after repeatedly applying the corrector C are denoted by $(\tilde{\phi_0}, \tilde{\phi_1}, .., \tilde{\phi_T})$.

3.1 PARTIAL DIFFERENTIAL EQUATIONS FOR REACTIVE FLOWS

The physical system we investigate is a laminar premixed methane-air flame with a single step chemistry. It is governed by the following Navier-Stokes equations Poinsot & Veynante (2005)

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_i}{\partial x_i} = 0 && \rho C_p(\frac{\partial T}{\partial t} + \frac{\partial u_i T}{\partial x_i}) = \dot{\omega}'_T + \frac{\partial}{\partial x_i}(\lambda \frac{\partial T}{\partial x_i}) \\ &\frac{\partial \rho u_j}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_i} = -\frac{\partial p}{\partial x_j} + \frac{\partial \tau_{ij}}{\partial x_i} && \frac{\partial \rho Y_k}{\partial t} + \frac{\partial \rho u_i Y_k}{\partial x_i} = \dot{\omega}_k + \frac{\partial}{\partial x_i}(\rho D_k \frac{\partial Y_k}{\partial x_i}) \end{aligned} \tag{2}$$

where ρ, u_i, p, T, Y_k denote the density, velocity, pressure, temperature, and species mass fractions of species k, respectively. τ, D_k, λ are the viscous tensor, the diffusion coefficient of species k and mixture thermal conductivity. In addition, C_p denotes the mixture specific heat capacity. $\dot{\omega}_k$ and $\dot{\omega}'_T$ are the species reaction rate and heat release rate, respectively.

For simplicity, a one-step chemistry is used to model the chemical reactions. We consider a chemical system of the type $\upsilon'_F F + \upsilon'_O O \rightarrow$ products, where υ'_F and υ'_O are the stoichiometric coefficients corresponding to fuel and oxidizer. The simplifications proposed by Williams (1985) and Mitani (1980) are used to model the reaction rates $\dot{\omega}_k$ for each species. The fuel consumption rate $\dot{\omega}_F$ is assumed to have the Arrhenius form.

$$\dot{\omega}_F = \upsilon'_F W_F B_1 T^{\beta_1} (\frac{\rho Y_F}{W_F})^{n_F} (\frac{\rho Y_O}{W_O})^{n_O} \exp(-\frac{E_a}{RT}) \tag{3}$$

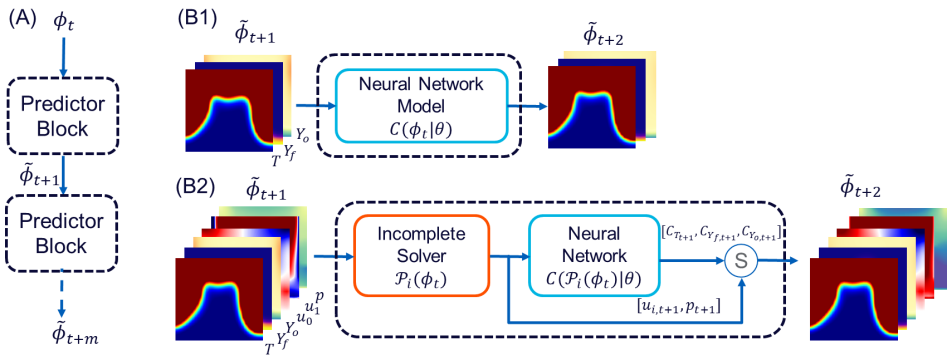

Figure 2: (A) Multi-step training framework helps to learn the dynamics of complete PDE solver over longer rollouts. (B1) Details of the input flow state and predictor block used in purely data-driven approach and (B2) hybrid NN-PDE approach.

The heat release source term $\dot{\omega}'_T$ and the fuel source term $\dot{\omega}_F$ are linked by $\dot{\omega}'_T = -Q\dot{\omega}_F$, where Q is the heat of reaction per unit mass. Following Poinsot & Veynante (2005), the parameters corresponding to the methane-air flame are chosen:$B_1 = 1.0810^7$ uSI; $\beta_1 = 0$; $E_a = 83600$ J/mole; $n_F = 1$; $n_O = 0.5$; $Q = 50100$ kJ/kg; $C_p = 1450$ J/(kgK).

3.2 Problem formulation

The incomplete PDE solver solves the set of equations 2 without the source terms and reaction rates $\dot{\omega}_k$ and $\dot{\omega}'_T$. The complete PDE solver solves the full equations in equation 2. The neural network model, denoted by $C(\mathcal{P}_i(\phi)|\theta)$, will correct the incomplete/non-reacting flow states $\mathcal{P}_i(\phi)$ to obtain the complete/reacting states $\mathcal{P}_c(\phi)$ as shown in figure 2 (C). It will learn to model the effects of the unknown chemistry using the neural network parameters θ and input flow state, $\phi = [u_i, p, T, Y_f, Y_o]$. Temperature and mass fraction fields are directly affected due to the unknown chemical reaction, therefore we consider a problem where multiple fields are corrected and predicted simultaneously. As seen from equation 2, and equation 3, the temperature and species mass fractions fields are strongly coupled. This increases the complexity of the prediction problem significantly. A slight error in one of the fields will propagate into the other fields, making the predictions diverge very quickly. In the following, a subscript $C_s(\circ)$ will denote that the neural network C generates the field s, e.g., C_T generating the temperature field.

$$\tilde{\phi}_{t+1} = [u_{i,t+1}, p_{t+1}, C_T(\mathcal{P}_i(\phi_t)|\theta), C_{Y_f}(\mathcal{P}_i(\phi_t)|\theta), C_{Y_o}(\mathcal{P}_i(\phi_t)|\theta)] \tag{4}$$

where $\tilde{\cdot}$ indicates the new state or the corrected state. $u_{i,t+1}$ and p_{t+1} are the time-advanced velocity and pressure field, respectively, predicted using the incomplete PDE solver. As outlined above, C_T denotes the neural network model generating a corrected temperature field given the predicted state at time $t+1$. In short, equation 4 is denoted as $\tilde{\phi}_{t+1} = C_S(\mathcal{P}_i(\phi_t)|\theta)$, where $S = \{T, Y_f, Y_o\}$ are the corrected states.

3.3 Neural Network Models

In this work, we study a *hybrid NN-PDE* approach of augmenting a neural network model with an incomplete PDE solver. For the hybrid NN-PDE approach, the neural network is integrated and trained in a loop with the incomplete solver using stochastic gradient descent for m time steps, as shown in figure 2 (A,C). Here, the number of temporal look-ahead steps, m, is an important hyper-parameter of the training process. Higher m provides the network with longer-term feedback at training time through the gradient roll-outs. The gradients for larger m give the model improved feedback on how the time dynamics of the incomplete solver affect the input states, and hence which corrections need to be inferred by the model.

A *purely data-driven* model (PDD) is used as a baseline to compare the results of the hybrid NN-PDE approach. A purely data-driven approach in this setting employs a neural network model to learn the complete flow states $\mathcal{P}_c(\phi)$ given an input flow state ϕ^S where $\phi^S = [T, Y_f, Y_o]$, as shown in figure

2 (B). The new state predicted by the trained neural network model is given by $\tilde{\phi}^S_{t+1} = C(\phi^S_t|\theta)$. For hybrid NN-PDE approach and PDD approach, neural network part of the predictor block in figure 2 consists of a fully convolutional neural network model with 5 ResBlocks. Each ResBlock consists of a convolutional layer with LeakyRelu activation followed by another convolutional layer. Each layer performs convolution of filter size 5×5 producing a vector of 32 features, bypassing the input every 2 convolutions. These skip connections, alleviate the vanishing gradient problem He et al. (2016). The network receives the input states as shown in figure 2. The output of the neural network model is used to obtain the corrected state $\tilde{\phi}_{t+1}$ as specified above. We constrain the mass fraction fields Y_k to contain physical values in the range $Y_f \in [0, 0.05]$ and $Y_o \in [0, 1]$. All models are trained for 100 epochs with a batch size of 3 and a learning rate of 0.0001.

We additionally compare against the *Fourier neural operator* (FNO) of Li et al. (2020a) as an example of a state-of-the-art neural operator method. We adapt the FNO formulation proposed in Li et al. (2020a) to suit our application. The new state predicted by the trained model is given by $\tilde{\phi}^S_{t+1} = C(\phi^S_t|\theta)$ where $\phi^S = [T, Y_f, Y_o]$ and $C(\circ)$ represents the Fourier neural operator. Detailed description of all models are provided in the appendix. All three models use an L_2 based loss to train the neural networks, which is evaluated for m steps as

$$\mathcal{L}(\theta) = \sum_{n=t+1}^{t+m+1} \sum_{\phi=\{T,Y_f,Y_o\}} ||\tilde{\phi}_n, \mathcal{P}_c(\phi_n)||_2 \tag{5}$$

3.4 Differentiable PDE solver

The central component of the structure of the hybrid NN-PDE approach is the differentiable solver, which allows us to embed a PDE solver for the incomplete model into the training of a neural network. The differentiable solver acts as a set of additional layers in the network, which do not contain parameters and hence are not trainable. Instead, they provide derivatives of the outputs of the simulation with respect to its inputs and parameters. Finite differences can be used to compute the gradients of the PDE solver, but they are computationally very expensive for high-dimensional PDEs. Differentiable solvers resolve this issue by solving the adjoint problem Pontryagin (1987) via analytic derivatives.

Here, we use the differentiable PDE solver from the *phiflow* framework of Holl et al. (2019) in combination with *Tensorflow*. The marker-and-cell method Harlow & Welch (1965) is adopted to represent temperature, pressure, density, species mass fraction fields in a centered grid, and velocities in a staggered grid. Basic differential operators such as gradient, divergence, curl, and Laplace operators are implemented in TensorFlow using basic mathematical tensor operations. These differential operators act on a 9 point stencil of grid points and the corresponding derivatives are straight-forward to compute. Advection is implemented with a semi-Lagrangian step, while the Poisson problem for the pressure and its gradient are solved implicitly. Efficient analytic derivatives for all these operations are then combined via back-propagation Werbos (1990).

4 Numerical Experiments

We consider a case of a planar 2D premixed methane-air flame propagating in a quiescent mixture (**Planar-v0**) and two cases of the transient evolution of initially planar laminar premixed flame into a Bunsen-type flame under different inlet velocity conditions (**uniform-Bunsen, nonUniform-Bunsen**). We obtain the target data by considering the source terms as defined by equation 3 with the parameters mentioned in section 3.1.

Planar-v0 For the most basic case, the planar 2D flame model setup, we consider the reacting Navier-Stokes equations described in section 3 with zero inlet velocity i.e. $U = 0$. We consider a square domain of size 0.05 m $\times$ 0.05 m with 32×32 resolution and closed boundary conditions. So, the domain is resolved with a uniform grid with cell size of 1.5625 mm. The simulation is initialized using a steep transition between a premixed methane-air mixture and burnt gases. Our training data consists of 6 simulations of 300 steps created by varying the equivalence ratio $E_{\text{train}} = \{1.0, 0.9, 0.8, 0.7, 0.6\}$, where the equivalence ratio (E) is the stoichiometric

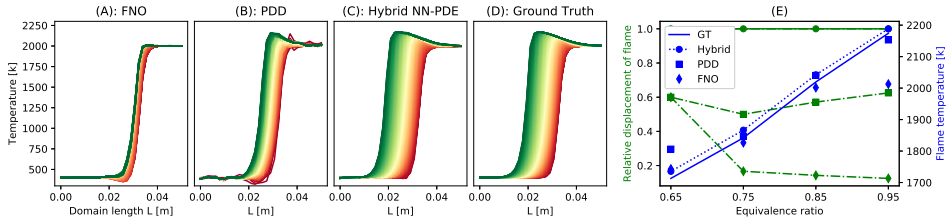

Figure 3: 1D cut of the planar flame simulation over 300 steps. The initial state is plotted in red, target state in green. Hybrid NN-PDE approach predicts physically accurate results over longer rollouts with correct flame temperature and relative displacement of flame front across different equivalence ratios.

mixture (φ) of fuel (Y_f) and oxidizer (Y_o) mass fractions; $E = \varphi \frac{Y_f}{Y_o}$. The test dataset contains $E_{\text{test}} = \{0.95, 0.85, 0.75, 0.65\}$.

uniform-Bunsen In contrast to the planar case, the premixed methane-air mixture is now fed with constant velocity. The boundary conditions upstream, at $x = 0$, are $T_0 = 800K, v \in \mathbb{R}, u = 0$. The target simulation contains a heat release rate term, and in this simulation, the initial temperature field evolves into a Λ-shaped flame at the end of the 300th time step. Training and testing datasets are created by varying the equivalence ratio and inlet velocity amplitude U: $E_{\text{train}} = \{1.0, 0.9, 0.8\}$ and $U_{\text{train}} = \{0.45, 0.4, 0.3\}$. The test dataset uses $E_{\text{test}} = \{0.95, 0.85\}$ $U_{\text{test}} = \{0.43, 0.375, 0.325\}$. The flame length and flame temperature significantly vary depending on the inlet velocity and equivalence ratio provided. Further details on the boundary conditions used and training data visualizations can be found in the appendix.

nonUniform-Bunsen As a third case we consider the transient evolution of a premixed methane-air flame with the non-uniform inlet velocity. The boundary conditions upstream, at $x = 0$, are $T_0 = 800K, v \in \mathbb{R}^n, u = 0$. The inlet velocity is randomly fluctuated between $v \in [0.2, 0.65]^n$. We experiment with two different domain sizes, 32×32 (**nonUniform-Bunsen32**) and 100×100 (**nonUniform-Bunsen100**), as shown in figure 4. The larger domain size used is closer to the practical reactive flow domain used in CFD applications Jaensch et al. (2017) with highly resolved flame. These inlet velocity conditions generate complex flame shapes which increase the difficulty of the prediction problem. We consider simulation sequences with 500 steps as it takes longer time for the flame to reach the steady-state solution.

5 Results

We demonstrate the capabilities of the proposed learning approach to represent the complete PDE description with tasks of increasing difficulty. We additionally study its ability to generalize to unseen operating conditions such as equivalence ratios, simultaneous variations in constant inlet velocity and equivalence ratio, and non-uniform inlet velocity profiles. As baselines, we compare against a purely data-driven approach with exactly the same neural network model and look-ahead steps, and the modified version of Fourier neural operator of Li et al. (2020a) to include look-ahead steps as discussed in section 3.3.

5.1 Qualitative and Quantitative Analysis

For Planar-v0 case, we visualize the 1D transverse cut of the simulation domain over 300 timesteps in figure 3. The hybrid NN-PDE approach successfully captures the propagation of the flame. The purely data-driven approach and FNO fail to capture the correct flame propagation over longer simulation steps. In figure 3 (E), we compare two physical quantities of this reactive flow case, the flame temperature and relative displacement of the flame front, across different equivalence ratios. It can be seen that the hybrid NN-PDE model (blue circles) accurately predicts the flame temperature for different test cases (solid blue line). The relative displacement of the flame front is computed as $\frac{|\tilde{x}_{300} - \tilde{x}_0|}{|x_{300} - x_0|}$, where x_t is the position along the flame normal at time t on the $1200K$ isotherm of the ground truth simulation, and $\tilde{x}$ denotes the predicted position. For all test cases, the hybrid approach

Table 1: Top: Mean and standard deviation of absolute percentage error for all the testsets; Bottom: mean and standard deviation of squared error over 300 time steps of all testsets

		Baseline FNO	Baseline NN	Hybrid NN-PDE	Hybrid NN-PDE-dt
MAPE	Planar-v0	8.27 $\pm$ 5.51%	6.33 $\pm$ 3.05%	1.40 $\pm$ 0.65%	**1.21** $\pm$ 0.55%
	uniform-Bunsen	15.57 $\pm$ 8.72%	7.58 $\pm$ 3.73%	**0.72** $\pm$ 0.37%	1.11 $\pm$ 0.52%
	nonUniform-Bunsen32	12.30 $\pm$ 7.98%	12.48 $\pm$ 11.31%	**2.04** $\pm$ 1.39%	2.46 $\pm$ 1.61%
	nonUniform-Bunsen100	-	-	**3.23** $\pm$ 3.76%	4.14 $\pm$ 4.99%
MSE	Planar-v0	88316 $\pm$ 124394	34491 $\pm$ 43966	**1122** $\pm$ 1300	2295 $\pm$ 2687
	uniform-Bunsen	220817 $\pm$ 167024	72563 $\pm$ 56919	**721** $\pm$ 1007	1893 $\pm$ 2389
	nonUniform-Bunsen32	184860 $\pm$ 185694	130219 $\pm$ 142094	**9647** $\pm$ 13903	17569 $\pm$ 24392
	nonUniform-Bunsen100	-	-	**23293** $\pm$ 37451	26862 $\pm$ 47639

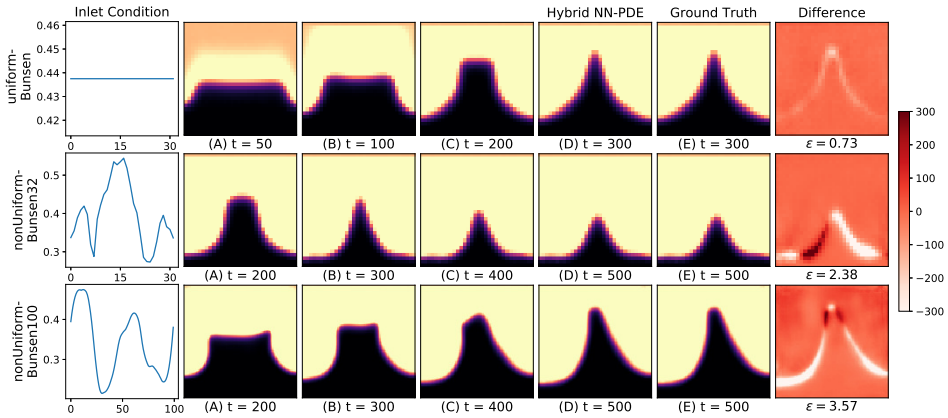

Figure 4: Left : Non-uniform inlet velocity profile used. (A-D) Temperature field prediction by hybrid NN-PDE approach over 500 steps given the inlet velocity profile. (E) Ground truth data Right : Difference between ground truth data and hybrid NN-PDE output at t = 500 for 32 $\times$ 32 and 100 $\times$ 100 testcases.

very accurately predicts the flame displacement over 300 steps, while the other approaches yield significant errors. Table 1 compares the mean absolute percentage errors (MAPE) and mean squared errors (MSE) of temperature field for all the cases considered in section 4. For Planar-v0 scenario, FNO and PDD approach yields MAPE of 8.27% and 6.33% respectively. Hybrid NN-PDE model trained with 32 look-ahead steps reduces the error to 1.4%.

Figure 4 shows the temporal evolution of the hybrid NN-PDE approach predicted over 300 time steps for uniform-Bunsen case. It shows the results for a test case with $U = 0.4375$ and $E = 0.95$. It predicts a symmetric flame with accurate flame height and achieves very low MAPE of 0.73%, when compared with the ground truth data. As shown in table 1, amongst baseline models PDD performs better with an error of 7.58% compared to FNO model which yields an error of 15.57%. A neural network model combined with an incomplete PDE solver for 32 look-ahead steps yields a significantly lower error of 0.72%. This means, hybrid approach reduces the errors by a factor of 10 over baselines considered.

Finally we study a complex scenario of nonUniform-Bunsen flame with 32 $\times$ 32 and 100 $\times$ 100 resolution. Figure 4 shows snapshots of the hybrid NN-PDE model predictions for both the cases of nonUniform-Bunsen. The trained model accurately predicts the flame simulation over long roll-outs of 500 steps. The hybrid NN-PDE approach outperforms the purely data-driven approach and FNO with an improvement of $\sim 75\%$. The purely data-driven approach predicts the flame dynamics with larger errors for all the cases considered. The MSE values show this trend even more clearly. The large standard deviations of the MSE numbers indicate that the predictions made by the baseline approaches contain substantial deviations from target values. What is striking in these numbers is that, despite using the same training data and neural network architecture, the hybrid approach

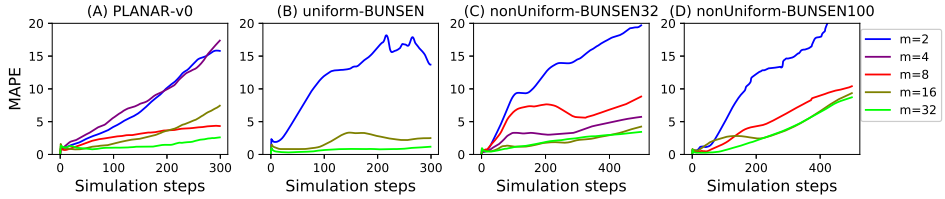

Figure 5: MAPE of temperature field over all testsets. Models trained with higher look-ahead steps m accurately predict the temporal evolution of dynamics for longer duration across all cases considered.

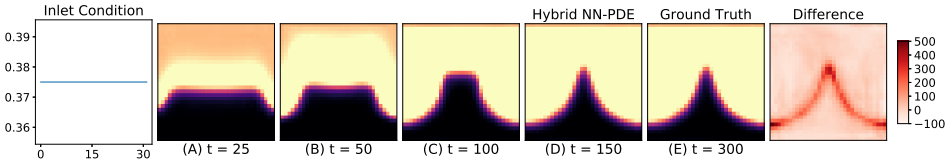

Figure 6: Left : Constant inlet velocity profile used. (A-D) uniform-Bunsen case trained and inferred at double time-step using hybrid NN-PDE approach. It achieves same flame shape at t = 150 for which the complete PDE solver requires 300 timesteps.

outperforms the purely data-driven approach thanks to its learned collaboration with the incomplete PDE solver. It accurately reproduces the complete PDE behavior.

We evaluate the effect of varying look-ahead steps on the performance of the hybrid model. We show the comparison of models trained with $m = \{2, 4, 8, 16, 32, 64\}$ for different cases in figure 5. Models with smaller m (2 or 4) do not learn to accurately correct the fields over long time and quickly diverge from the target simulation. Using larger m improves the quality of prediction drastically as the model learns the correction via the gradients over longer simulation steps. For Planar-v0 and uniform-BUNSEN cases, iterating the NN and PDE solver for 32 time-steps improves the accuracy by 68.7 % and 61.8 % respectively compared to the $m = 8$ model. For nonUniform-Bunsen, model performance improves by 71.9 % by using $m = 64$ instead of $m = 8$ model.

5.2 Relaxing temporal stiffness in the PDE solver

Traditionally, the source terms and reaction rate terms involved in modeling the fast chemistry of reacting flows require the use of very small time-steps in simulations due to the stiffness involved in solving the chemical mechanism. The incomplete PDE solver used in the hybrid NN-PDE approach, does not contain the source terms and reaction rate terms. Therefore, the time scales associated with the chemical reactions play a less important role in maintaining numerical stability, and it becomes possible for the solver to employ larger time steps. To illustrate this advantage, we train the neural network model with a timestep that is twice as large as the largest timestep Δt_c required for the complete PDE solver to yield a stable result.

We mimic the setup described in sections 4 and with the same test datasets, but now the incomplete PDE solver uses a timestep of $2\Delta t_c$. Figure 6 shows that the results obtained from the models trained with a larger time-step at $\{25, 50, 100, 150\}$ simulation steps are in good agreement with the target data for the difficult uniform-Bunsen case. The flame dynamics are predicted accurately while using less simulation steps. This highlights the capabilities of learned, hybrid PDE solvers, which can produces these solutions without the stability problems exhibited by the complete solver, while at the same time being more accurate than pure data-driven predictions.

5.3 Flame shape control

We finally test the applicability of the neural network model in conjunction with the reactive flow solver to obtain the desired flame shape. We consider an inverse problem of predicting an inlet velocity field that yields a target flame shape. We use the predictor-corrector approach from Holl

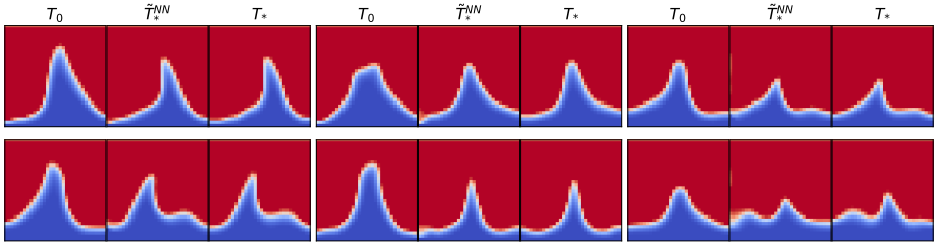

Figure 7: Temperature field predictions achieved by the neural network model T_*^{NN} from given initial flame shape T_0 to achieve target flame shape T_* by controlling inlet flow velocity.

et al. (2019) with differentiable physics loss function and refined control sequence. We consider the setup of nonUniform-Bunsen simulation with 32 $\times$ 32 resolution. The goal is to achieve a target temperature field with the desired flame shape from a given initial flame shape by controlling the inlet velocity field. We assume that the temperature and mass-fraction fields are observable and only the partial velocity field is controllable. Figure 7 compares the target flame shapes achieved using the neural network model with the target flame shapes (T_*) for different testcases. The neural network model is able to control the velocity of fuel-air mixture flow to obtain the desired flame shape.

6 Conclusions

This paper demonstrates that neural networks can be employed for learning the completion of partial PDE solvers in the context of a multi-physics system. The trained neural network model works alongside a partial solver to accurately account for and predict the effects of unknown physics. The ability of the presented hybrid NN-PDE approach to predict the long-term temporal evolution of multi-physics systems was demonstrated on the specific case of reacting flows in different configurations. When compared with a purely data-driven approach and state-of-the-art FNO approach, qualitative and quantitative performance of the hybrid NN-PDE model is found to be clearly superior. The incomplete PDE description helps the neural network model recover the target simulation with a significantly improved accuracy. This hybrid NN-PDE model is able to predict the correct evolution of the important features of the multi-physics system (in our case, the flame interface and flame shape) for longer simulation steps in all scenarios and is able to generalize to other (initial and boundary) conditions of the system (as demonstrated by variations in the equivalence ratio and inlet velocity forcing). We also show that such hybrid approach may have the added benefits of allowing to relax numerical requirements linked to the potential stiffness of the unknown physics.

As future work, the proposed hybrid NN-PDE solver framework can be further utilized to predict and control the dynamics of such multi-physical systems. We note that the proposed approach does not seek to replace the classical physical simulation approaches. However, it is an important step in the direction of harnessing the capabilities of neural network models using the knowledge of partial differential equations for multi-physical systems.

References

Ekhi Ajuria Illarramendi, Antonio Alguacil, Michaël Bauerheim, Antony Misdariis, Benedicte Cuenot, and Emmanuel Benazera. Towards an hybrid computational strategy based on deep learning for incompressible flows. In *AIAA AVIATION 2020 FORUM*, pp. 3058, 2020.

Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019.

Filipe de Avila Belbute-Peres, Thomas Economon, and Zico Kolter. Combining differentiable pde solvers and graph neural networks for fluid flow prediction. In *International Conference on Machine Learning*, pp. 2402–2411. PMLR, 2020.

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B Kovachki, and Andrew M Stuart. Model reduction and neural networks for parametric pdes. *arXiv preprint arXiv:2005.03180*, 2020.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 6572–6583, 2018.

Gideon Dresdner, Dmitrii Kochkov, Peter Norgaard, Leonardo Zepeda-Núñez, Jamie A Smith, Michael P Brenner, and Stephan Hoyer. Learning to correct spectral methods for simulating turbulent flows. *arXiv preprint arXiv:2207.00556*, 2022.

Kai Fukami, Koji Fukagata, and Kunihiko Taira. Super-resolution reconstruction of turbulent flows with machine learning. *Journal of Fluid Mechanics*, 870:106–120, 2019.

Vadim N Gamezo, Alexei M Khokhlov, Elaine S Oran, Almadena Y Chtchelkanova, and Robert O Rosenberg. Thermonuclear supernovae: Simulations of the deflagration stage and their implications. *Science*, 299(5603):77–81, 2003.

Samuel Greydanus, Misko Dzamba, and Jason Yosinski. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32:15379–15389, 2019.

Jiequn Han, Arnulf Jentzen, and E Weinan. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

Francis H Harlow and J Eddie Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids*, 8(12):2182–2189, 1965.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Philipp Holl, Vladlen Koltun, and Nils Thuerey. Learning to control pdes with differentiable physics. In *ICLR*, 2019.

Mark Z Jacobson. *Fundamentals of atmospheric modeling*. Cambridge university press, 1999.

S Jaensch, M Merk, EA Gopalakrishnan, S Bomberg, T Emmert, RI Sujith, and W Polifke. Hybrid CFD/low-order modeling of nonlinear thermoacoustic oscillations. *Proceedings of the Combustion Institute*, 36(3):3827–3834, 2017.

Eugenia Kalnay. *Atmospheric modeling, data assimilation and predictability*. Cambridge university press, 2003.

Byungsoo Kim, Vinicius C Azevedo, Nils Thuerey, Theodore Kim, Markus Gross, and Barbara Solenthaler. Deep fluids: A generative network for parameterized fluid simulations. In *Computer Graphics Forum*, pp. 59–70. Wiley Online Library, 2019.

Omar M Knio, Habib N Najm, and Peter S Wyckoff. A semi-implicit numerical scheme for reacting flow: Ii. stiff, operator-split formulation. *Journal of Computational Physics*, 154(2):428–467, 1999.

Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21), 2021.

Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9(5):987–1000, 1998.

Corentin J Lapeyre, Antony Misdariis, Nicolas Cazard, Denis Veynante, and Thierry Poinsot. Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame*, 203:255–264, 2019.

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

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020b.

Tim C Lieuwen. *Unsteady Combustor Physics*. Cambridge University Press, 2012.

Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*, pp. 3208–3216. PMLR, 2018.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

Tohru Mitani. Propagation velocities of two-reactant flames. *Combustion Science and Technology*, 21(3-4):175–177, 1980.

Habib N Najm, Peter S Wyckoff, and Omar M Knio. A semi-implicit numerical scheme for reacting flow: I. stiff chemistry. *Journal of Computational Physics*, 143(2):381–402, 1998.

Ravi G Patel, Nathaniel A Trask, Mitchell A Wood, and Eric C Cyr. A physics-informed operator regression framework for extracting data-driven continuum models. *Computer Methods in Applied Mechanics and Engineering*, 373:113500, 2021.

Jaideep Pathak, Mustafa Mustafa, Karthik Kashinath, Emmanuel Motheau, Thorsten Kurth, and Marcus Day. Using machine learning to augment coarse-grid computational fluid dynamics simulations. *arXiv preprint arXiv:2010.00072*, 2020.

Thierry Poinsot and Denis Veynante. *Theoretical and numerical combustion*. RT Edwards, Inc., 2005.

Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. CRC press, 1987.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Chae M Rhie and Wei-Liang Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA journal*, 21(11):1525–1532, 1983.

David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2):1–96, 2022.

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

Kim Stachenfeld, Drummond Buschman Fielding, Dmitrii Kochkov, Miles Cranmer, Tobias Pfaff, Jonathan Godwin, Can Cui, Shirley Ho, Peter Battaglia, and Alvaro Sanchez-Gonzalez. Learned simulators for turbulence. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=msRBojTz-Nh`.

Nils Thuerey, Konstantin Weißenow, Lukas Prantl, and Xiangyu Hu. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 58(1):25–36, 2020.

Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating eulerian fluid simulation with convolutional networks. In *International Conference on Machine Learning*, pp. 3424–3433. PMLR, 2017.

Kiwon Um, Robert Brand, Fei Yun, Philipp Holl, and Nils Thuerey. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. In *Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 6111–6122, 2020.

Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1457–1466, 2020.

Gerhard Wanner and Ernst Hairer. *Solving ordinary differential equations II*, volume 375. Springer Berlin Heidelberg New York, 1996.

Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

Adam A Wheeler, William J Boettinger, and Geoffrey B McFadden. Phase-field model for isothermal phase transitions in binary alloys. *Physical Review A*, 45(10):7424, 1992.

Forman A Williams. Combustion theory, (1985). *Cummings Publ. Co*, 31, 1985.

Yuan Yin, Vincent Le Guen, Jérémie Dona, Emmanuel de Bézenac, Ibrahim Ayed, Nicolas Thome, and Patrick Gallinari. Augmenting physical models with deep networks for complex dynamics forecasting. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12):124012, 2021.

Ali Girayhan Özbay, Arash Hamzehloo, Sylvain Laizet, Panagiotis Tzirakis, Georgios Rizos, and Björn Schuller. Poisson cnn: Convolutional neural networks for the solution of the poisson equation on a cartesian mesh. *Data-Centric Engineering*, 2:e6, 2021. doi: 10.1017/dce.2021.7.

A Appendix

You may include other additional sections here.