



# A physics-informed machine learning approach for predicting dynamic behavior of reacting flows with application to hydrogen jet flames

Pushan Sharma <sup>a</sup>\*, Wai Tong Chung <sup>a</sup>, Matthias Ihme <sup>a,b,c</sup>

<sup>a</sup> Department of Mechanical Engineering, Stanford University, Stanford, CA 94305, USA

<sup>b</sup> Department of Photon Science, SLAC National Accelerator Laboratory, Menlo Park, CA 94025, USA

<sup>c</sup> Department of Energy Science and Engineering, Stanford University, Stanford, CA 94305, USA



## ARTICLE INFO

Dataset link: <https://blastnet.github.io/>

**Keywords:**

Physics-informed machine learning  
Transient dynamics  
Mixture-of-experts  
Spatiotemporal prediction  
Reacting flow  
Hydrogen flame

## ABSTRACT

Traditional data-driven modeling approaches suffer from large error accumulation over time, divergence from expected physical behavior, and poor generalizability for out-of-distribution samples. To address this, we present Physics-informed hybrid Multiscale and Partitioned Network (PiMAPNet), a physics-informed machine learning (ML) strategy for generating multi-scale and multi-physics predictions by integrating low-resolution physics-based models with neural networks. Motivated by prior work on hybrid methods that combine coarse-grain simulations with ML, PiMAPNet employs state-space decomposition on the hydrodynamic (velocity and pressure) and thermochemical (temperature and species mass fractions) quantities for improved predictions of multiphysical processes. In addition, the ML model utilizes a mixture-of-experts (MoE) architecture that partitions the thermochemical state-space and employs a separate ML model to represent a specific region within this partition. We demonstrate this ML framework on a reacting hydrogen/air jet flame configuration. Results demonstrate that both the purely data-driven ML model and a traditional PIML approach could not represent the entire state-space, which resulted in unphysical behavior in long-term predictions. In contrast, the MoE-based PiMAPNet achieves higher accuracy and demonstrates improved robustness over extended time windows and out-of-distribution scenarios. Through our analysis, we show that PiMAPNet offers faster inference speed than a numerical simulation with comparable accuracies in multiple physical quantities.

### Novelty and Significance Statement

This study introduces a novel physics-informed machine learning framework that enhances the predictive accuracy for chemically reacting flows by integrating low-resolution physics-based models with neural networks. The novelty of the framework lies in its specialized treatments for hydrodynamic and thermochemical variables. Additionally, the thermochemical state-space is partitioned to effectively capture the evolution of different regions within the state-space. The significance of our work is its ability to deliver highly accurate and robust predictions over extended time periods and for out-of-distribution scenarios. Furthermore, the separate treatment of different physical processes enables this framework to be extendable to other multiphysics systems, such as plasma physics or multiphase flows, making it a valuable tool for researchers across various domains in computational physics.

## 1. Introduction

High-fidelity simulations are essential in understanding physical processes and designing practical combustion systems [1]. In particular, with today's heightened environmental awareness, high-fidelity simulations play a crucial role in developing cleaner and more efficient combustion technologies, reducing emissions, and transitioning towards sustainable fuel utilization. However, managing high computational costs associated with resolving all spatial and temporal scales

within multi-physical processes is still a challenge [2,3]. These challenges become especially pronounced in combustion simulations where detailed chemical kinetics introduces issues related to dimensionality, stiff chemistry and the consideration of the interaction between turbulence and chemistry, or in capturing transient events, such as ignition, blow-out dynamics, and flame-wall coupling.

One approach for lowering computational costs involves the employment of dimensionality reduction methods, such as Proper Orthogonal Decomposition (POD) [4] or Principal Component Analysis

\* Corresponding author.

E-mail addresses: [ps729@stanford.edu](mailto:ps729@stanford.edu) (P. Sharma), [wtchung@stanford.edu](mailto:wtchung@stanford.edu) (W.T. Chung), [mihme@stanford.edu](mailto:mihme@stanford.edu) (M. Ihme).

(PCA) [5], for developing reduced order models (ROMs) from large systems of differential equations that govern flow physics. Recent studies [6] have explored ROMs with deep-learning based dimensionality reduction, where non-linear methods have been shown to be more effective when compared with their aforementioned linear counterparts. Surrogate modeling via Physics-informed Machine Learning (PIML) [7–10] offers an alternative pathway towards reducing the computational cost of flow simulations by combining low-cost physical calculations and machine learning (ML) predictions. Without the consideration of physical information, the use of purely ML-based regression within numerical predictions has been shown to result in significant out-of-distribution errors [8,11,12]. PIML methods ameliorate these drawbacks in ML through the incorporation of well-known physical knowledge (such as conservation principles or governing equations) within ML models. This treatment has been shown to enhance ML training efficiency and improve model generalizability in predictions over long temporal durations [8].

The Physics-Informed Neural Network (PINN) [13] is a well-known example of a PIML approach. In a PINN, the network solution is constrained during training, during which the ML training loss function is combined with the residuals of the governing PDEs [14], in order to improve extrapolation in unseen test conditions. PINNs have motivated the use of similar physics-informed loss functions in ML-based super-resolution models, which can model closure terms for under-resolved simulations of both turbulent non-reacting [15–17] and reacting flows [12,18,19]. Other popular PIML methods include TurbulentFlowNet (TFNet) [20] and Fourier Neural Operator (FNO) [21], which employ special model architectures that can account for the multi-scale and spectral nature of turbulence, respectively, thereby enabling them to outperform their conventional counterparts in flow physics problems. A few other studies [22–25] also explore the integration of coarse PDE solutions with ML models in order to improve the accuracy and stability for high-resolution spatiotemporal predictions of non-reacting canonical flows. A detailed description of other PIML methods for spatiotemporal modeling is provided in several review articles [7,8,11,26].

In reacting flows, ML models have been proposed for solving a range of problems including modeling combustion chemistry [27–29], treating stiff chemical systems [30–33], improving combustion manifolds [34–37], and selecting combustion models [38]. However, the progress on the development of ML-based models for spatiotemporal predictions of reacting flows remains relatively nascent compared to their non-reacting counterparts [9]. An et al. [39] devised separate CNN models as surrogates for Unsteady Reynolds-averaged Navier-Stokes (URANS) and large-eddy simulation (LES) to capture the spatiotemporal behavior of hydrogen combustion including detailed chemistry. Wu et al. [40] performed a similar study with a different ML architecture, showing that the injection of Gaussian noise during ML training improves the stability of the model predictions. Both of these purely data-driven ML studies provided better statistical accuracy for URANS simulations compared to its LES counterpart, concluding that detailed and small-scale structures within the finer mesh in high-fidelity simulations pose significant challenges for predictions over long temporal durations.

In several reacting flow studies, ML models have demonstrated improved accuracy when learning tasks are partitioned and assigned to different submodels, as demonstrated in problems such as adaptive chemistry [41] and combustion emission modeling [42,43]. In these efforts, the composition space is divided using a variety of techniques (including mixture-of-experts (MoEs) [33,36], self-organizing maps [42,44,45], k-means clustering [42,43,46] or PCA [41,46,47]). This division simplifies the learning process, which in turn allows the model accuracy to improve significantly.

This study aims to address challenges associated with long-term spatiotemporal predictions of reacting flow simulations by combining a PIML method with suitable treatment of the composition space using

an MoE approach. Specifically, the objective of this study is to develop a cost-effective Physics informed hybrid Multiscale and Partitioned Network (PiMAPNet) for predicting the spatiotemporal behavior of reacting flow simulations. Here, we focus on the stability and generalizability of PiMAPNet and compare its accuracy with a traditional PPNN and a physics-agnostic ML model in the context of unsteady combustion simulations. To this end, we evaluate the model behavior on a hydrogen flame ignition and flame stabilization in a vitiated co-flow burner configuration.

The remainder of this manuscript has the following structure. The mathematical formulation of the numerical solver and details of the PiMAPNet architecture are presented in Section 2. The configuration of the hydrogen flame is discussed in Section 3. Results, presented in Section 4, first examine the stability of PIML architecture in the context of spatiotemporal reactive flow predictions, followed by the importance of MoE for partitioning the state-space. Then, accuracy, generalizability and computational cost of PiMAPNet are analyzed for samples within and out of training distributions. The study finishes with concluding remarks in Section 5.

## 2. Methods

Our objective is to reduce the computational cost of numerical simulations of unsteady reacting flows. To that end, we consider PiMAPNet, a PIML approach that combines affordable coarse grid simulations with state-space partitioned ML framework. In this section, we first discuss the governing equations and the numerical solver, followed by the details of the ML architecture and its training procedure.

### 2.1. Physical model

The physical model [48–50] numerically solves the conservation laws for mass, momentum, total energy, and chemical species:

$$\partial_t \rho + \nabla \cdot (\rho \mathbf{u}) = 0 , \quad (1a)$$

$$\partial_t (\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} , \quad (1b)$$

$$\partial_t (\rho e_t) + \nabla \cdot [\mathbf{u}(\rho e_t + p)] = -\nabla \cdot \mathbf{q} + \nabla \cdot [\boldsymbol{\tau} \cdot \mathbf{u}] , \quad (1c)$$

$$\partial_t (\rho Y_k) + \nabla \cdot (\rho \mathbf{u} Y_k) = -\nabla \cdot \mathbf{j}_k + \rho \dot{\omega}_k , \quad (1d)$$

with density  $\rho$ , velocity vector  $\mathbf{u}$ , pressure  $p$ , specific total energy  $e_t$ , stress tensor  $\boldsymbol{\tau}$ , and heat flux vector  $\mathbf{q}$ .  $Y_k$ ,  $\mathbf{j}_k$ , and  $\dot{\omega}_k$  are the mass fraction, diffusion flux, and source term for chemical species  $k$  with  $k = 1$  to  $N_s - 1$ , where  $N_s$  is the number of species. The heat flux, diffusion flux, and stress tensor are given by:

$$\mathbf{q} = -\kappa \nabla T , \quad (2a)$$

$$\mathbf{j}_k = -\rho D_k \nabla Y_k , \quad (2b)$$

$$\boldsymbol{\tau} = \rho v \left[ (\nabla \mathbf{u}) + (\nabla \mathbf{u})^T - \frac{2}{3} (\nabla \cdot \mathbf{u}) \mathbf{I} \right] , \quad (2c)$$

where  $\kappa$ ,  $D_k$  and  $v$  are the mixture thermal conductivity, mixture-averaged diffusion coefficient of species  $k$ , and kinematic viscosity, respectively. The total energy,  $e_t$ , is related to the sensible internal energy ( $e$ ) and the kinetic energy:

$$e_t = e + \frac{1}{2} \mathbf{u} \cdot \mathbf{u} , \quad (3)$$

where  $e$  takes the form:

$$e = h - RT = \sum_{k=1}^{N_s} Y_k \left[ \Delta h_k^0(T_{ref}) + \int_{T_{ref}}^T c_{p,k}(T') dT' \right] - RT , \quad (4)$$

with specific isobaric heat capacity  $c_{p,k}$  and heat of formation  $\Delta h_k^0$  of species  $k$ , and  $R$  is the mixture-averaged gas constant. Since  $e$  is a non-linear function of  $T$ , an iterative method is used to compute  $T$  given  $e$ . The system of equations is closed with the ideal gas law.

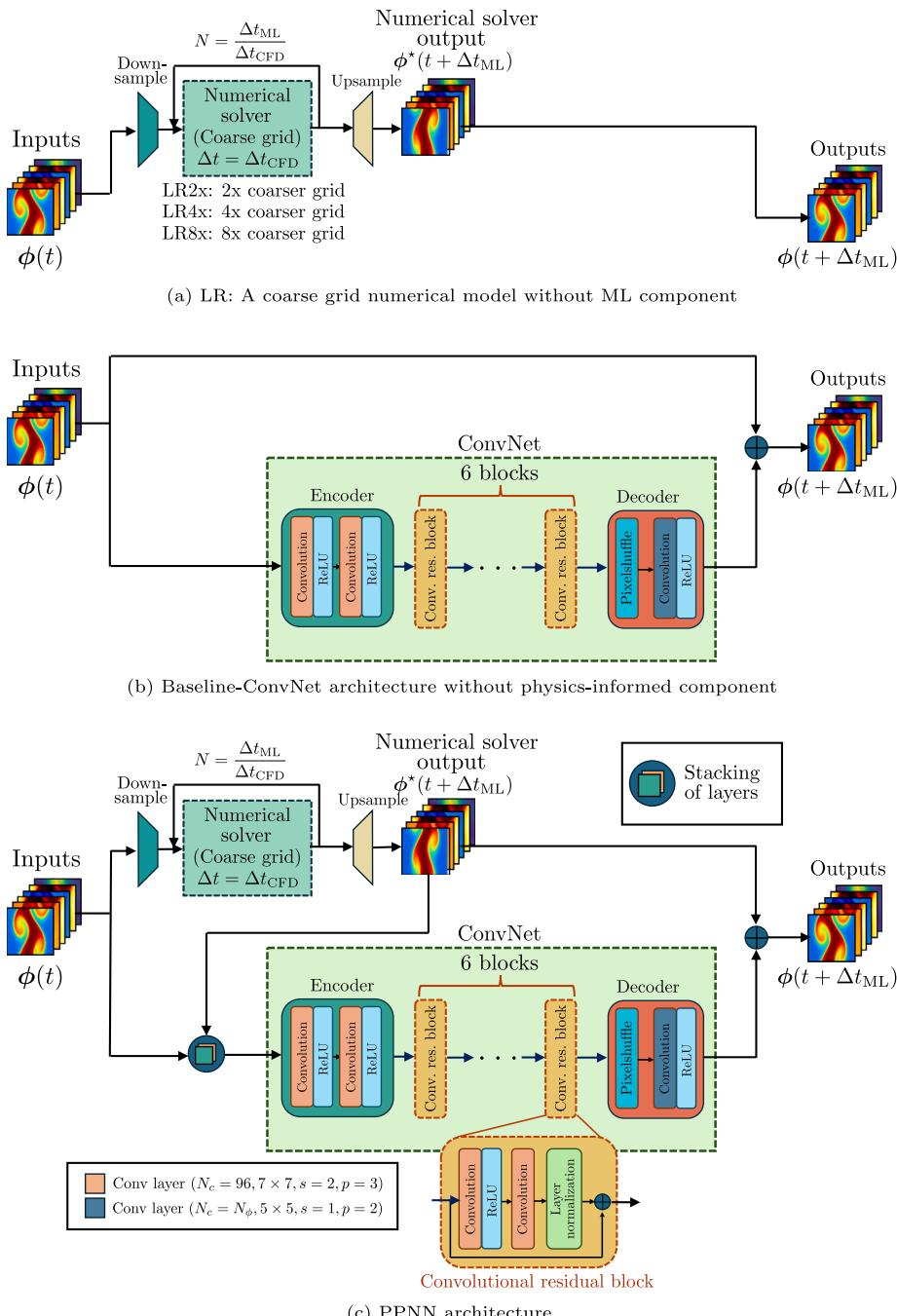


Fig. 1. Schematics of the ML models considered in this study: (a) LR, (b) Baseline-ConvNet, (c) PPNN, and (d) PiMAPNet.

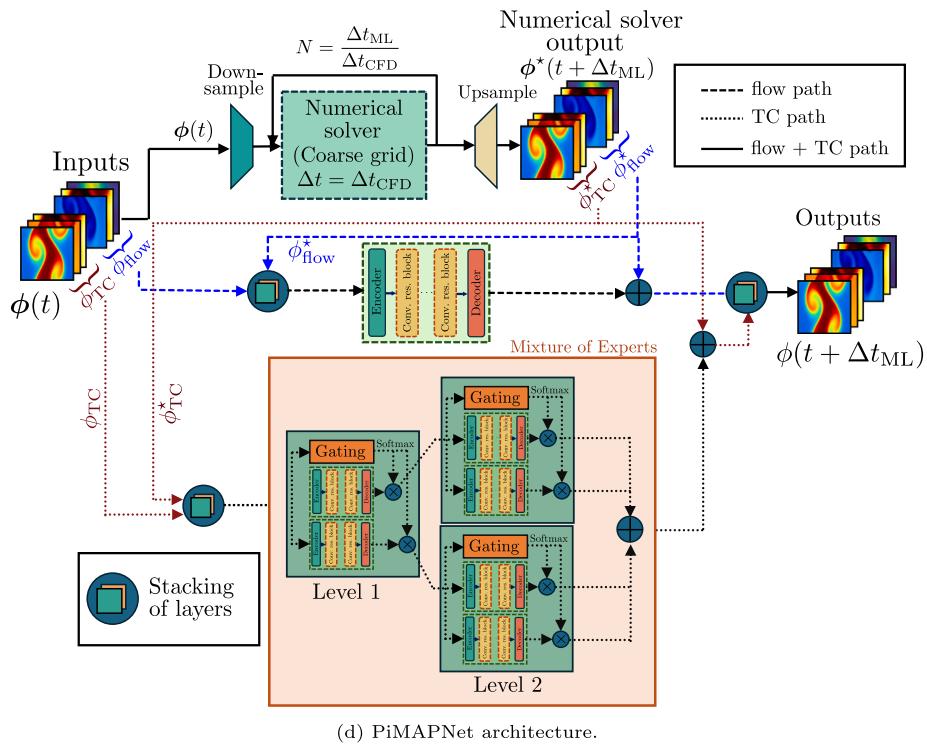
A finite-volume approach is used for the discretization of the system of equations in Eq. (1). The convective fluxes are discretized using a sensor-based hybrid scheme, where a high-order, non-dissipative scheme is combined with a low-order scheme to describe interfaces and flow field discontinuities [51]. A central scheme, which is 4th-order accurate on uniform meshes, is used along with a 2nd-order accurate ENO scheme. We apply a 2nd-order accurate simpler balanced-splitting scheme [52] to separate the convection, diffusion, and reaction operators. The stiff chemical source terms are integrated in time using a semi-implicit 4th-order accurate Rosenbrock-Krylov scheme [52]. For all other non-stiff operators, we utilize a strong stability preserving 3rd-order Runge-Kutta (SSP-RK3) scheme [53].

## 2.2. ML architecture

In this study, we propose a new PiMAPNet model architecture. This model combines a coarse grid numerical solver with a ML component that treats hydrodynamics and thermochemical quantities separately, and partitions the thermochemical state space using MoE for better predictive accuracy. In this section, we provide the details of these individual components first, followed by the overall PiMAPNet architecture. The input state vector to the model,  $\phi$ , includes all the quantities of interest:

$$\phi = [\rho, \rho\mathbf{u}, p, T, \rho\mathbf{Y}]^T, \quad (5)$$

where  $\mathbf{Y} \in \mathbb{R}^{N_s}$  is the species mass fraction vector,  $\mathbf{u} \in \mathbb{R}^{N_d}$  is the velocity vector with  $N_d = 2$  as we consider a 2D configuration, which



(d) PiMAPNet architecture.

Fig. 1. (continued).

will be discussed in Section 3. The cardinality of  $\phi$  is  $N_\phi = N_s + N_d + 2$ . For  $\phi$ , the governing equations in Eq. (1) can be simplified in the form:

$$\frac{\partial \phi}{\partial t} = \mathcal{F}(\phi(t)), \quad (6)$$

where  $\mathcal{F}$  includes the convective and diffusion fluxes and the source terms.

### 2.2.1. Numerical solver on coarse grid

The first model, shown in Fig. 1(a), only involves low-resolution (LR) solution of the governing equations (Eq. (1)) on a coarse grid. The flowfield  $\phi$  is downsampled by interpolation on a LR grid at time  $t$ , i.e.,  $\phi_{LR}(t) = \mathbb{P}\phi(t)$ , where  $\mathbb{P}$  is the interpolation matrix. The downsampled flowfield is the input to the numerical solver, which is advanced until  $t + \Delta t_{ML}$  with a sub timestep of  $\Delta t_{CFD}$ . The CFD timestep,  $\Delta t_{CFD}$ , is constrained by the Courant–Friedrichs–Lowy (CFL) condition. Thus, the number of sub timesteps within the numerical solver is  $N_{skip} = \frac{\Delta t_{ML}}{\Delta t_{CFD}}$ . Finally, the updated flowfield at  $t + \Delta t_{ML}$ ,  $\phi_{LR}^*(t + \Delta t_{ML})$ , is upsampled to  $\phi^*(t + \Delta t_{ML})$  by interpolation on the high-resolution grid, i.e.,  $\phi^*(t + \Delta t_{ML}) = \mathbb{P}^{-1}\phi_{LR}^*(t + \Delta t_{ML})$ , such that  $\mathbb{P}\mathbb{P}^{-1} = I$ , where  $I$  is the identity matrix. Here, we choose bilinear interpolation for downsampling and upsampling of the flowfields. Three different levels of coarse grid simulations are performed with a 2× ('LR2x'), 4× ('LR4x'), and 8× ('LR8x') coarser grid resolutions compared to the fine-grid required to resolve all scales.

### 2.2.2. Baseline-ConvNet

The second model is a CNN-based ML model, referred as 'Baseline-ConvNet' hereafter, is used as a next step spatiotemporal flowfield predictor without any physics-informed component, as shown in Fig. 1(b). In Baseline-ConvNet,  $\phi(x, t)$  is advanced to  $t + \Delta t_{ML}$ , where  $\Delta t_{ML}$  is one ML timestep, using ConvNet by:

$$\phi(x, t + \Delta t_{ML}) = \phi(x, t) + \text{ConvNet}(\phi(x, t)). \quad (7)$$

The ConvNet is an encoder-decoder type CNN [25], as shown in Fig. 1(b). In the present study, the encoder includes two convolutional

layers with  $7 \times 7$  kernel, zero padding and a stride of 2 along with ReLU activation function. The encoder is followed by 4 convolutional residual blocks, each of which contains a convolution filter with  $7 \times 7$  kernel, 96 channels and zero padding of 3, ReLU activation and a layer normalization. Finally, the decoder part includes pixelshuffle with an upscale factor of 4 and a convolutional layer of  $5 \times 5$  kernel and zero padding of 2. The total number of trainable parameters in the ConvNet is approximately 27 million.

### 2.2.3. PPNN

The third model is a PPNN architecture, inspired by Liu et al. [25]. This architecture (Fig. 1(c)) combines a coarse grid numerical solver and 'ConvNet' as a trainable ML component.  $\phi$  at a given time  $t$  is used as input to the physics-informed component of the PIML architecture, where the governing PDEs (Eq. (1)) are solved on a coarser grid with a sub timestep  $\Delta t_{CFD}$ . The output of the coarse grid simulation at the end of one ML timestep is denoted by  $\phi_{LR}^*(t + \Delta t_{ML})$ . The output is then upsampled to the high-resolution grid using the method described in Section 2.2.1. The upsampled fields are denoted by  $\phi^*(t + \Delta t_{ML})$ . Typically,  $\Delta t_{ML}$  is significantly larger than  $\Delta t_{CFD}$ , which accelerates the spatiotemporal prediction compared to traditional CFD approaches. In the following stage, the input of the ML component involves both  $\phi(t)$  and  $\phi^*(t + \Delta t_{ML})$ . The overall advancement of the flowfield from  $\phi(t)$  to  $\phi(t + \Delta t_{ML})$  in PPNN can be written as:

$$\phi_{LR}(t) = \mathbb{P}\phi(t), \quad (8a)$$

$$\phi_{LR}^*(t + \Delta t_{ML}) = \text{LR}(\phi_{LR}(t)) \quad [\text{substeps of } \Delta t_{CFD}], \quad (8b)$$

$$\phi^*(t + \Delta t_{ML}) = \mathbb{P}^{-1}\phi_{LR}^*(t + \Delta t_{ML}), \quad (8c)$$

$$\phi(t + \Delta t_{ML}) = \phi^*(t + \Delta t_{ML}) + \text{ConvNet}(\phi(t), \phi^*(t + \Delta t_{ML})), \quad (8d)$$

where  $\phi(t + \Delta t_{ML})$  denotes the final prediction of the PIML model after one ML timestep. During inference,  $\phi(t + \Delta t_{ML})$  is used as input for the next timestep. The PPNN architecture involves a single ConvNet, which is optimized using the mean square error (MSE) loss function, which is evaluated on fine grids.

In essence, PPNN can be interpreted as a solver with explicit forward Euler scheme with a timestep of  $\Delta t_{ML}$ . The ML component of it learns the residual between the ground truth (GT) and the LR output at the updated timestep,  $\phi^*(t + \Delta t_{ML})$ . For a simplified version of the governing equation in Eq. (6), the ConvNet learns the following equation based on forward Euler scheme:

$$\text{ConvNet}(\phi(t), \phi^*(t + \Delta t_{ML})) \approx [F(\phi(t)) - \mathbb{P}^{-1}F(\phi_{LR}(t))] \Delta t_{ML}. \quad (9)$$

Physically, the ConvNet in Eq. (9) learns to correct the interpolation error at the initial time  $t$ , the additional numerical diffusion invoked by the coarser resolution and the subgrid scale stress and flux terms, if the flow is turbulent.

#### 2.2.4. PiMAPNet

The PPNN architecture (Fig. 1(c)) was shown to be more robust in simulations of incompressible flows, involving next-step velocity and pressure predictions [25]. However, for reacting flow simulations, the thermochemical state introduces additional complexity as different regions of the state space exhibit different physical and chemical behavior. For instance, in a flame, the chemical composition and their rates of change may exhibit significant variability across manifold regions, changing from inert or equilibrium states to regions of auto-ignition, thereby encompassing a broad spectrum of timescales [3]. Despite the nonlinear nature of a neural network, a single ConvNet cannot fully capture the entire dynamics of the thermochemical manifold. To address this, we propose a flexible ML model architecture that allows the input space to be decomposed based on the distinct physical processes involved. In this study, for a reacting flow simulation, we decompose the state vector,  $\phi$ , into hydrodynamic quantities or flow variables ( $\phi_{flow}$ ) and thermochemical variables ( $\phi_{TC}$ ):

$$\phi = (\phi_{flow}, \phi_{TC}) ; \phi_{flow} = [\rho, \rho\mathbf{u}, p]^T \quad \text{and} \quad \phi_{TC} = [T, \rho Y]^T. \quad (10)$$

Other choices of decompositions are equally feasible. The overall model architecture is shown in Fig. 1(d). The evolution of  $\phi$  from time  $t$  to  $t + \Delta t_{ML}$  in a PiMAPNet follows the same steps as Eq. (8), except the last step. In PiMAPNet, the flow variables,  $\phi_{flow}$ , are trained by a single ConvNet. In addition, we partition the thermochemical manifold and use separate ConvNets to represent different regions of the manifold, which improves the learning process, thus the prediction accuracy [36]. This partitioning of  $\phi_{TC}$  is achieved by using MoE, where each ‘expert’ is a separate ConvNet. For a general case with  $N_{expt}$  experts in an MoE, let  $\hat{\phi}_{TC,n}$  be the prediction of the  $n$ th expert, and  $\pi_n \in [0, 1]$  represents the prior probability of choosing expert  $n$  for the current input sample. Then the overall prediction will be:

$$\hat{\phi}_{TC} = \sum_{n=1}^{N_{expt}} \pi_n \hat{\phi}_{TC,n}. \quad (11)$$

The prior probabilities,  $\pi_n$ , are determined by a gating network, which is comprised of a single-layer feed-forward neural network featuring a sigmoidal activation function in the hidden layer and a standard softmax activation function in the output layer [36]. With this, in PiMAPNet, Eq. (8d) is modified as:

$$\phi_{flow}(t + \Delta t_{ML}) = \phi_{flow}^*(t + \Delta t_{ML}) + \text{ConvNet}(\phi_{flow}(t), \phi_{flow}^*(t + \Delta t_{ML})) \quad (12a)$$

$$\phi_{TC}(t + \Delta t_{ML}) = \phi_{TC}^*(t + \Delta t_{ML}) + \text{MoE}(\phi_{TC}(t), \phi_{TC}^*(t + \Delta t_{ML})). \quad (12b)$$

Similarly, the ConvNet and MoE in PiMAPNet learn the following.

$$\text{ConvNet}(\phi_{flow}(t), \phi_{flow}^*(t + \Delta t_{ML})) \approx [F(\phi_{flow}(t)) - \mathbb{P}^{-1}F(\phi_{LR,flow}(t))] \Delta t_{ML} \quad (13a)$$

$$\text{MoE}(\phi_{TC}(t), \phi_{TC}^*(t + \Delta t_{ML})) \approx [F(\phi_{TC}(t)) - \mathbb{P}^{-1}F(\phi_{LR,TC}(t))] \Delta t_{ML} \quad (13b)$$

Although the flow and thermochemical variables are treated separately in PiMAPNet, the coupling between these variables is captured in the LR component of PiMAPNet. To explore whether this coupling could be explicitly included in the ML components, we considered the full state vector ( $\phi$ ) as the input for both the ConvNet and the MoE. However, that did not improve the prediction accuracy compared to using  $\phi_{flow}$  for ConvNet and  $\phi_{TC}$  for MoE. Additionally, using these decomposed inputs reduces the training cost and memory requirement.

Ideally, in a MoE, each expert should conditionally perform inference on one part of the thermochemical manifold. However, the output of the gating network typically provides an intermediate value between 0 and 1, not allowing a hard boundary between two experts. To ensure a hard boundary between experts or conditional computations, the training process for each MoE involves the following steps:

- Both the experts and the gating network are trained at the same time, updating both  $\hat{\phi}_{TC,n}$  and  $\pi_n$ . The loss function is given by [54]:

$$L = -\log \left[ \sum_{n=1}^{N_{expt}} \pi_n \exp \left( -\frac{1}{2} \|\phi_{TC}(x, t) - \hat{\phi}_{TC,n}(x, t)\|_2^2 \right) \right]. \quad (14)$$

At the end of this stage, the prior probabilities ( $\pi_n$ ) for each expert are obtained.

- Similar to the concept of switch transformers [55], for each input in  $\phi_{TC}$ , instead of using all experts, we route to the expert with maximum  $\pi$ . In that case, the overall output at this stage will be modified from Eq. (11) as

$$\hat{\phi}_{TC} = \hat{\phi}_{TC,M} \quad \text{with } M = \text{argmax}_m(\pi_m). \quad (15)$$

The objective here is to train each expert separately on areas of the manifold where it is considered the best expert. As the gating network is frozen in this step ( $\pi$  is not being updated), the loss function in this stage is the MSE between  $\phi_{TC}$  and  $\hat{\phi}_{TC}$ , obtained from Eq. (15). This approach divides the regions among experts with distinct boundaries, avoiding overlap and ensuring specialization.

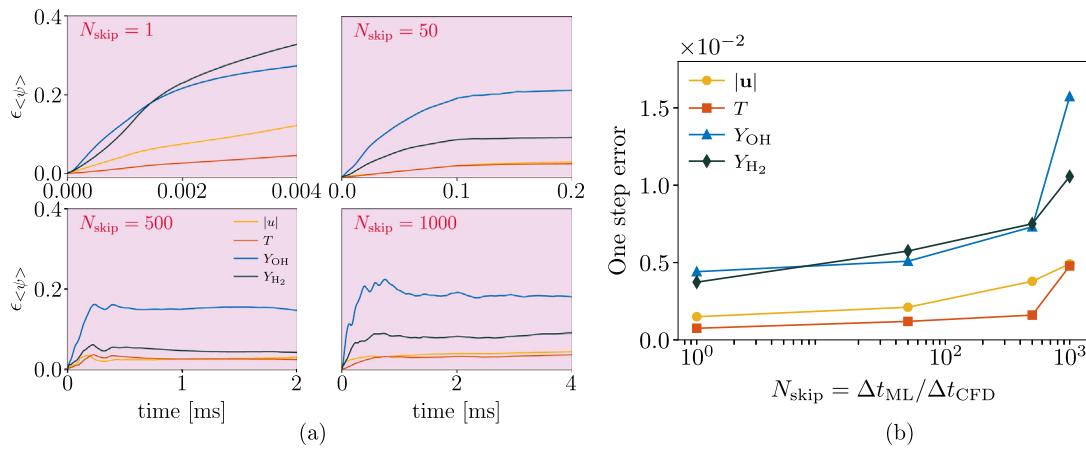
For this study, we choose the number of experts in each MoE to be  $N_{expt} = 2$ . Additionally, we utilize a hierarchical splitting technique, wherein the thermochemical state-space is partitioned at multiple levels as shown in Fig. 1(d). In this study, we employed a two-level MoE framework. Thus, the hierarchical MoE architecture consists of one MoE in the first level, in which the manifold is split between two experts. The second level consists of two MoEs with a total of 4 experts (Fig. 1(d)), where the regions of the manifold assigned at the first level are further divided. At each level, the gating network determines the assignment of regions to experts. All loss functions are evaluated on fine grids.

If the gating network consistently assigns the same expert to handle inputs within its region, determined by Eq. (15), the region remains undivided, and the output from the preceding MoE level is retained rather than undergoing further partitioning. The PiMAPNet architecture can readily be extended to a larger  $N_{expt}$  to partition the thermochemical state-space of a more complex reacting flow configuration.

### 3. Case study: hydrogen jet flame

#### 3.1. Simulation configuration

Spatially developing hydrogen jet flames, similar to the study by Yoo et al. [56], are simulated, but in a 2D configuration, which is representative of the original study as the flow remains transient, involves complex chemistry, and the flame exhibits lift-off at higher jet Reynolds numbers. The central jet with  $D = 1.92$  mm consists of a mixture of 65% hydrogen and 35% nitrogen by volume with an inlet temperature of 400 K. The jet is surrounded by a co-flow of heated



**Fig. 2.** (a) Propagation of error in mean velocity, temperature, OH and  $\text{H}_2$  mass fractions in physical time for different ML timestep size,  $N_{\text{skip}}$ , (b) Average one timestep error for different values of  $N_{\text{skip}}$ .

**Table 1**  
Summary of all models considered in the present study.

Model name	Description
LR2x	Numerical solver with 2x coarser resolution (Fig. 1(a))
LR4x	Numerical solver with 4x coarser resolution (Fig. 1(a))
LR8x	Numerical solver with 8x coarser resolution (Fig. 1(a))
Baseline-ConvNet	Baseline convolutional neural network with MSE loss (Fig. 1(b)) (No physics-informed component)
PPNN	PIML with LR4x and one ConvNet (Fig. 1(c))
PiMAPNet2x	PIML with LR2x, one ConvNet for $\phi_{\text{flow}}$ and MoEs for $\phi_{\text{TC}}$ (Fig. 1(d))
PiMAPNet4x	PIML with LR4x, one ConvNet for $\phi_{\text{flow}}$ and MoEs for $\phi_{\text{TC}}$ (Fig. 1(d))
PiMAPNet8x	PIML with LR8x, one ConvNet for $\phi_{\text{flow}}$ and MoEs for $\phi_{\text{TC}}$ (Fig. 1(d))

air at 1100 K and 1 bar pressure. The heated co-flow autoignites the hydrogen-air mixture at a downstream location of the burner exit. The location of ignition and lift-off length (LOL) are determined by the residence time, which depends on the bulk jet velocity ( $U_j$ ), and the ignition delay time of the mixture [56].

The mean inlet axial velocity profile,  $U_{\text{inlet}}$ , is given by

$$U_{\text{inlet}}(x=0, y) = U_{\text{cf}} + \frac{U_j - U_{\text{cf}}}{2} \left[ \tanh\left(\frac{D}{\delta} \left[\frac{y}{D} - \frac{1}{2}\right]\right) + \tanh\left(\frac{D}{\delta} \left[\frac{y}{D} + \frac{1}{2}\right]\right) \right], \quad (16)$$

where  $U_{\text{cf}}$  is the mean co-flow velocity and the shear-layer thickness,  $\delta$ , is specified to be  $0.1D$ . The jet Reynolds number,  $Re_j$ , defined as  $U_j D / v_j$  is varied between 5000 to 11,000 in this study.  $U_{\text{cf}}$  is chosen as 10% of  $U_j$ . The size of the computational domain is  $12.5D \times 15.625D$ . A detailed hydrogen-air chemical mechanism, consisting of 9 species and 21 elementary reactions [57], is employed in this study. The time step ( $\Delta t_{\text{CFD}}$ ) is set to  $10^{-8}$  s. For the highest  $Re_j$  considered in this study, a grid spacing of  $15 \mu\text{m}$  is necessary to resolve all scales and the flame structure [56]. An equidistant and uniform grid is used in this study, resulting in  $N_x = 1600$  grid points in the axial direction and  $N_y = 2000$  grid points in the spanwise direction.

### 3.2. Training dataset

The objective of this study is to train PiMAPNet to learn the parametric variations of  $\phi$  as a function of  $Re_j$ . In the training set, we use three different  $Re_j$ , i.e., 5000, 6000 and 7500. For each  $Re_j$ , the training data is collected after  $t = t_0$ , when the flow is past auto-ignition and becomes statistically stationary. Selecting such training data increases

both statistical and pointwise accuracy, as well as longtime stability of the models [58].

In the present study, GT is generated with a grid size of  $N_x \times N_y$ . Three different levels of coarser grids (LR2x, LR4x, and LR8x) are used in the physics-informed part of the PiMAPNet architecture. During training of each of the models, each component of  $\phi$  is normalized with its corresponding mean and standard deviation, and a 80:20 split is used for training and validation purposes. The ML models were initialized using Xavier initialization [59] and trained with an initial learning rate of 0.0001, halved every 200 epochs for adaptive fine-tuning. We employed the Adam optimizer [60] and conducted training using a batch size of 16. All models are implemented in PyTorch [61] and trained on a NVIDIA A100 GPU. For a succinct discussion in the rest of this study, we summarize all models for spatiotemporal predictions considered in Table 1. A total of 5 models ('Baseline-ConvNet', 'PPNN', 'PiMAPNet2x', 'PiMAPNet4x' and 'PiMAPNet8x') are trained, resulting in a total computational cost of approximately 100 GPU-hours. The details of the training cost of PiMAPNets are provided in Appendix A. The training data for all the models consists of 200 snapshots from each  $Re_j$ . As a result, for different choice of ML timestep size, the training data covers different physical time in the reacting flow simulation, which affects the prediction accuracy. In the following section, we perform a hyperparameter study for the optimal choice of ML timestep size in the context of the current simulation configuration.

### 3.3. Effect of ML timestep size on prediction accuracy

The selection of ML timestep ( $N_{\text{skip}} = \Delta t_{\text{ML}}/\Delta t_{\text{CFD}}$ ) can be associated with a trade-off between accuracy and speedup in ML-based flow predictions. As such we perform a hyperparameter search for a suitable ML timestep size. In this search, we vary  $N_{\text{skip}}$  between 1 and 1000, training PiMAPNet4x at each  $N_{\text{skip}}$  across three  $Re_j = 5000, 6000$  and 7500. For each configuration, we collect  $N_{\text{ML}} = 200$  number of snapshots spanning a total physical time of  $N_{\text{ML}} \Delta t_{\text{ML}}$  or  $N_{\text{ML}} N_{\text{skip}} \Delta t_{\text{CFD}}$  in the training trajectories. The trained models for each  $N$  are then evaluated on  $Re_j = 10,000$  over  $N_{\text{ML}} = 400$  steps. To quantify the deviation from GT, we define the normalized root-mean-squared error (NRMSE) at time  $t$  as:

$$\epsilon_{\psi}(t) = \frac{\|\hat{\psi}(\mathbf{x}, t) - \psi(\mathbf{x}, t)\|_2}{\|\psi(\mathbf{x}, t)\|_2}, \quad \text{for } \psi \in \phi, \quad (17)$$

where  $\hat{\psi}$  is the prediction of the ML model for any quantity of interest  $\psi \in \phi$ .

Fig. 2a shows the propagation of prediction errors in mean velocity, temperature, OH and  $\text{H}_2$  mass fractions over physical time. It is important to note that the total physical time covered for a fixed

$N_{ML}$  increases with  $N_{skip}$ , since larger timesteps span longer temporal windows. Fig. 2b indicates the one-step prediction error in mean quantities for different values of  $N_{skip}$ . This error metric is calculated by taking a single ML timestep from each of the 400 test snapshots and computing the mean error across these predictions. For  $N_{skip} = 1$ , where the timestep size is very small ( $10^{-8}$  s in this case), we observe the minimum one-step error across all variables (Fig. 2b). As  $N_{skip}$  increases, one-step error increases monotonically for each variable. This results in the slower error growth rate in the short time prediction window for  $N_{skip} = 1$  compared to other values of  $N_{skip}$ . However, when viewed for long time rollout in Fig. 2a, the error grows very rapidly for  $N_{skip} = 1$  due to two compounding factors. Firstly, the large number of ML prediction steps for  $N_{skip} = 1$  leads to rapid error accumulation, which causes the predicted distributions to increasingly deviate from the input training distribution, a well-documented failure mode in ML-based predictors [62]. Secondly, the  $N_{skip} = 1$  configuration suffers from poor sample efficiency, as 200 ML steps in training cover an insufficient time window to adequately capture the complex large timescale dynamics of the reacting flow case. For velocity and temperature predictions, we observe comparable error magnitudes across  $N_{skip} = 50,500$ , and 1000. However, the species mass fraction predictions demonstrate optimal accuracy at  $N_{skip} = 500$ . At larger timesteps ( $N_{skip} = 1000$ ), although solutions maintain stability, they exhibit marginally higher errors compared to  $N_{skip} = 500$ . Despite the reduced number of ML steps, predicting finer structures becomes more challenging for a large value of  $N_{skip}$  as the accumulated error from the LR simulation increases. These results indicate a U-shaped relationship between ML timestep size and prediction accuracy, consistent with the observations in Stachenfeld et al. [62]. On one hand, since ML-based predictors are not constrained by CFL criteria and can learn efficient updates that capture relevant dynamics on larger timescales, which allows faster and efficient simulations. On the other hand, training ML predictors with very large timesteps can result in the inability to capture dynamics at smaller temporal scales. Based on these findings from this hyperparameter search, we adopt  $N_{skip} = 500$  for analyzing the capabilities of PiMAPNet in this work. However, the optimality of this value cannot be guaranteed due to the large computational costs associated with performing extensive hyperparameter searches, which is a known challenge in all ML studies [9].

## 4. Results

The principal aim of this study is to develop a PIML model to accelerate high-resolution simulations without compromising accuracy and generalizability. Employing a purely data-driven ML model for accelerating conventional CFD approaches often sacrifices accuracy. Integrating ML models with coarse grid simulations minimizes the computational expense of the fine-grid numerical solver while enhancing the accuracy compared to low-resolution simulations. To that end, we first investigate flames at  $Re_j = 5000$ . We assess the accuracy of the prediction of PiMAPNet and compare against the Baseline-ConvNet, LR and PPNN. More importantly, a well-trained ML model should generalize to new out-of-distribution conditions, e.g., a different  $Re_j$  in this case. To examine this, we perform simulations at  $Re_j = 10,000$ , and compare accuracy and computational cost between different models.

### 4.1. Comparison of PiMAPNet with baseline-ConvNet and PPNN

We compare the prediction accuracy of the Baseline-ConvNet model, PPNN and PiMAPNet4x for  $Re_j = 5000$ . Fig. 3(a) shows the instantaneous temperature fields after  $N_{ML} = 50$  and 100, both of which are part of the training. The evolution of errors in temperature and OH mass fraction are shown in Fig. 3(b). For this analysis, only PiMAPNet4x is considered to compare the accuracy with Baseline-ConvNet and PPNN to quantify the benefit of the proposed model architecture for reacting flow simulations. The other PiMAPNet models

with different levels of low-resolution simulations will be analyzed later in Section 4.3.

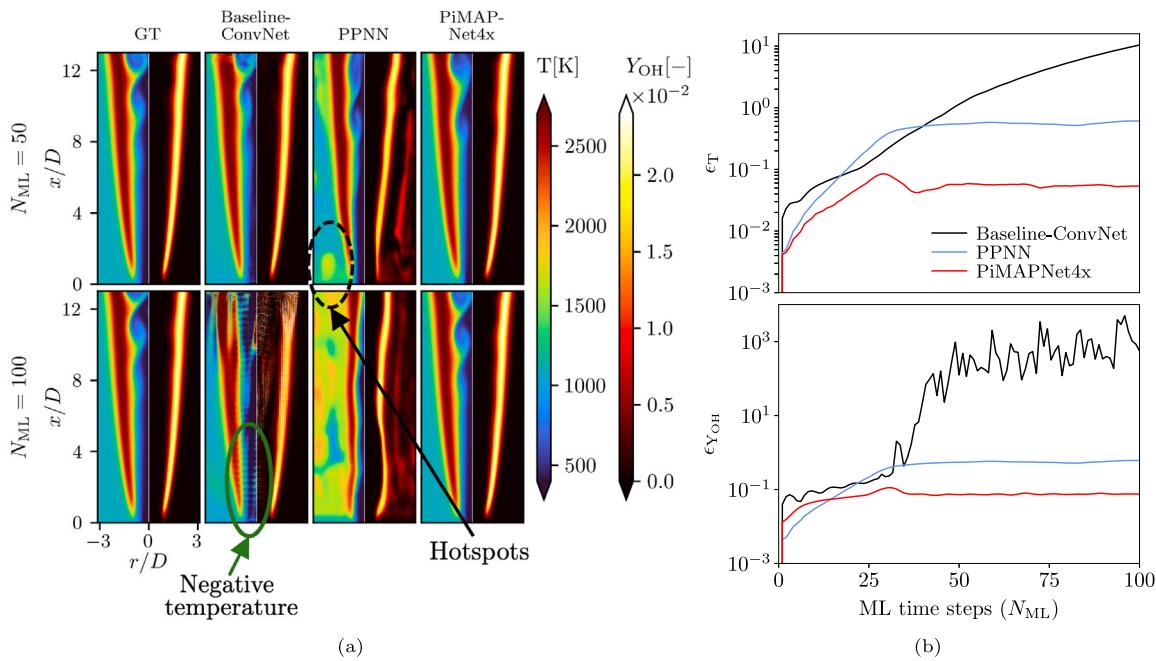
From Fig. 3(a), it can be observed that within  $N_{ML} = 100$ , the Baseline-ConvNet predictions significantly differ from the reference. Although the Baseline-ConvNet is trained on data for  $Re_j = 5000$ , the ML model still introduces a small bias at every time step. Over time, the ML model cannot correct the errors, thus the bias accumulates and the flow solution diverges for the Baseline-ConvNet. Eventually, the predictions exhibit unphysical behavior, such as negative temperature fields (Fig. 3(a)). Consequently, the errors in both temperature and OH mass fraction are also high, as observed in Fig. 3(b).

In case of PPNN, it can be seen from Fig. 3 that although the error accumulation is reduced compared to the Baseline-ConvNet (Fig. 3(b)), the temperature and OH fields exhibit spurious developments of hotspots away from the flame (Fig. 3(a)). These spurious reactive zones cause divergence from the reference solution. In contrast, PiMAPNet4x serves to mitigate the accumulation of bias effectively compared to both Baseline-ConvNet and PPNN. Specifically, the error in temperature prediction is reduced by 45% after  $N_{ML} = 100$  in case of PiMAPNet4x from an error of approximately 50% in case of PPNN and an even higher reduction in the error from Baseline-ConvNet. To better understand the reason for the generation of spurious reaction zones in PPNN and how these are addressed with PiMAPNet, we investigate the importance of the partition of the thermochemical manifold, which is discussed in Section 4.2.

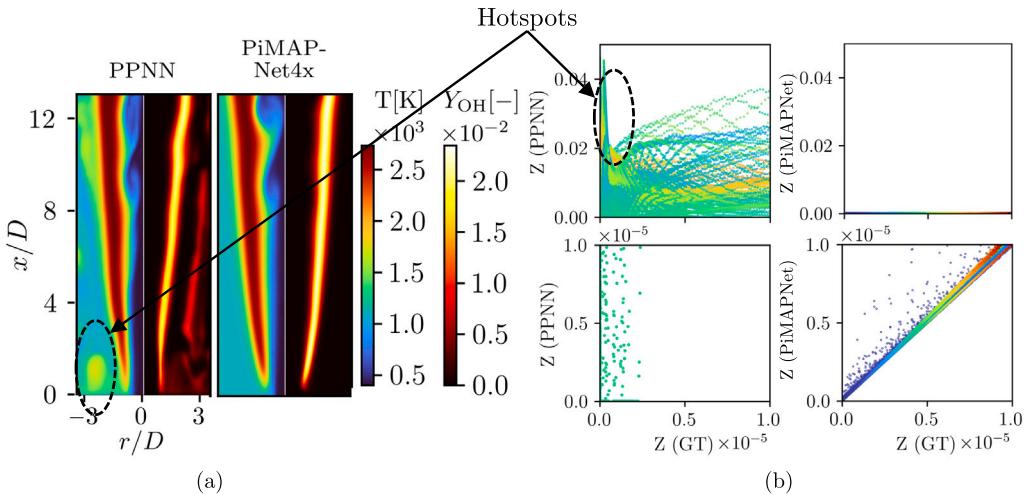
### 4.2. Importance of partition of thermochemical manifold

To investigate the source of the hotspots in the predictions of PPNN, we compare the mixture fraction ( $Z$ )-scatter plots in Fig. 4b. The mixture fraction is evaluated using Bilger's definition [63]. In the scatter plots, PPNN and PiMAPNet predictions are compared for very small values of  $Z$ . Such small  $Z$  values correspond to the non-reacting co-flow compositions. PPNN encounters challenges in predicting these extremely small  $Z$  values, leading to significant errors. Particularly, the change in compositions over time corresponding to such small  $Z$  values should be close to zero. However, the predicted mixture fraction from PPNN corresponding to a GT value of  $\mathcal{O}(10^{-5})$  are two orders of magnitude larger, as shown in Fig. 4b. The larger  $Z$  values indicate artificial mixing of fuel into the co-flow region by the ML model, which, under oxygen and high-temperature conditions, triggers spurious reactive zones in the coarse-grid numerical solution. This differs to the Baseline-ConvNet predictions (Fig. 3(a)), where ML prediction errors accumulate throughout the entire computational domain over time. Fig. 5(a) shows the probability distribution of temperature for GT, PPNN and PiMAPNet. The highest probability is observed near 1100 K for all three cases, which corresponds to a large number of co-flow compositions in the computational domain. While capturing the full temperature distribution with a single ConvNet model, PPNN captures the main reactive zone reasonably accurately as can be seen from the temperature contours and the distribution (temperature higher than 2200 K). However, the overall temperature distribution significantly differs from GT, especially between 1200 to 1600 K corresponding to spurious hotspots in the co-flow stream.

In contrast, PiMAPNet accurately predicts even the smallest target outputs by partitioning the thermochemical manifold and assigning separate ML models to different regions. Fig. 5(b) provides a visual representation of this partition in the physical domain. Fig. 5(c) illustrates the corresponding manifold partition in the  $T - Z$  space. Referring to the MoE model architecture (Fig. 1(d)), in level 1, the thermochemical manifold ( $\phi_{TC}$ ) is divided into two experts. Expert 1, corresponding to the blue region in Fig. 5(c), handles the high-temperature fuel-lean compositions in the flame, while all other compositions are managed by expert 2. In level 2, the first MoE attempts to further divide the blue region but fails, routing all compositions to one expert based on maximum prior probability. Consequently, predictions for the blue



**Fig. 3.** (a) Instantaneous temperature (left panel) and OH mass fraction (right panel) fields at two different time instants corresponding to  $N_{ML} = 50$  and  $N_{ML} = 100$  (both part of training), (b) Error propagation in time for the prediction of temperature and OH mass fraction for Baseline-ConvNet, PPNN and PiMAPNet4x in case of  $Re_j = 5000$ .



**Fig. 4.** (a) Instantaneous temperature (left panel) and OH mass fraction (right panel) fields after  $N_{ML} = 50$  corresponding to PPNN and PiMAPNet4x. The PPNN case exhibits hotspots in the co-flow region. (b) Comparison of mixture fraction ( $Z$ ) between GT and the predictions by PPNN and PiMAPNet for small target outputs.

region are solely derived from level 1. In contrast, the second MoE in level 2 successfully divides all the remaining compositions into the orange and red regions. The orange region represents the non-reacting co-flow compositions (Fig. 5(b)). The red region corresponds to the compositions in the fuel stream, two mixing lines (fuel-co-flow and fuel-product) as well as the fuel rich compositions. Overall, this partitioning is consolidated into three experts, denoted by ‘Expert 1’, ‘Expert 2’, and ‘Expert 3’ for clarity. Similar to Expert 1 in level 1, Expert 2 and Expert 3 in level 2 cannot be divided any further based on the gating network outputs. However, for a more complex configuration, the partitions may need to be extended further as part of training.

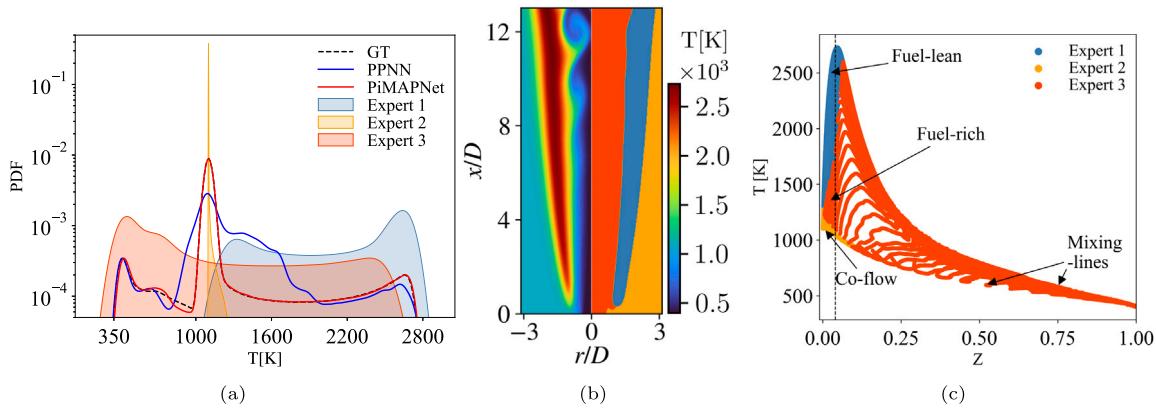
As opposed to PPNN, where a single ConvNet is responsible to learn the evolution of the full spectrum of compositions, in PiMAPNet, a single ConvNet (Expert 2) is dedicated to learning the dynamics of the region characterized by small  $Z$  values in the co-flow region. As a result, the prediction error associated with co-flow compositions remains small enough not to become chemically active in the coarse

grid numerical solver and the overall temperature distribution from the prediction of the PiMAPNet closely mimics the GT distribution.

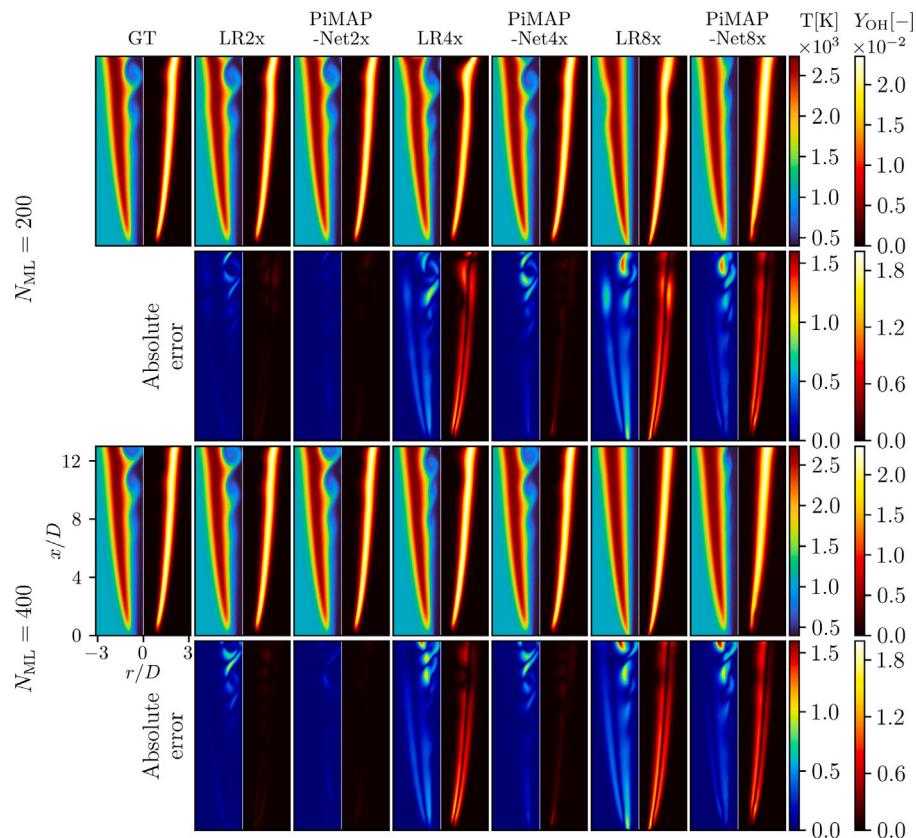
It is important to note that PiMAPNet’s use of multiple ConvNets results in a larger model size compared to both Baseline-ConvNet and PPNN. To evaluate whether these models can achieve similar predictive performance as PiMAPNet when scaled up, we increase their model sizes by 3x and repeat the analysis in Appendix B. While the larger models exhibited a reduction in prediction errors, they eventually diverge in a manner consistent with the behavior observed in Section 4.1.

#### 4.3. Analysis of PiMAPNet

In this section, we analyze PiMAPNet further considering long-term rollout, generalizability for out-of-training-distribution inputs, and computational gains compared to monolithic numerical CFD.



**Fig. 5.** (a) PDF of temperature corresponding to GT, PPNN and PiMAPNet. The shaded plots correspond to the PDF of the temperature of regions corresponding to each expert separately. (b) Temperature field (left panel) and the partition in the physical domain (right panel). (c) Partition of the thermochemical manifold in  $T - Z$  space for  $Re_j = 5000$  at time  $t_0 + 50\Delta t_{ML}$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



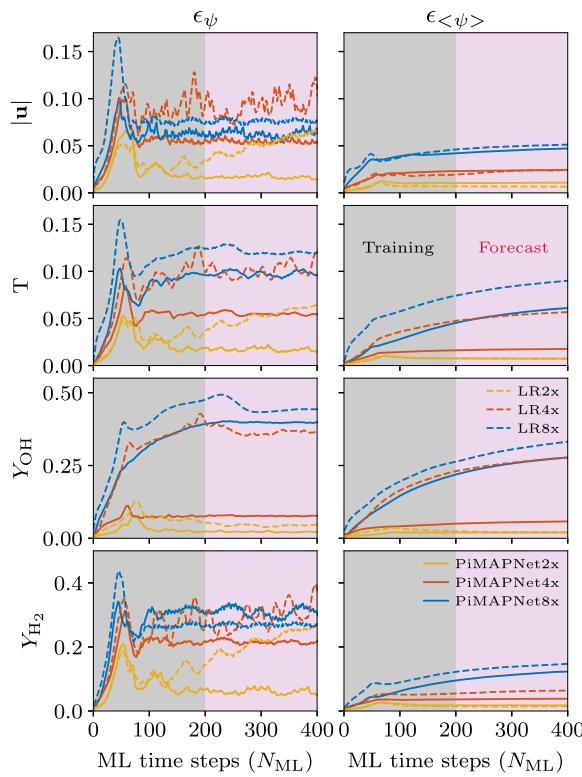
**Fig. 6.** Instantaneous temperature (left panel) and OH mass fraction (right panel) fields and absolute error in prediction of temperature and OH mass fractions at two different time instants corresponding to  $N_{ML} = 200$  (top) and  $N_{ML} = 400$  (bottom) for  $Re_j = 5000$ . The first column shows GT. The following columns correspond to predictions of LR and their PiMAPNet counterparts.

#### 4.3.1. Accuracy

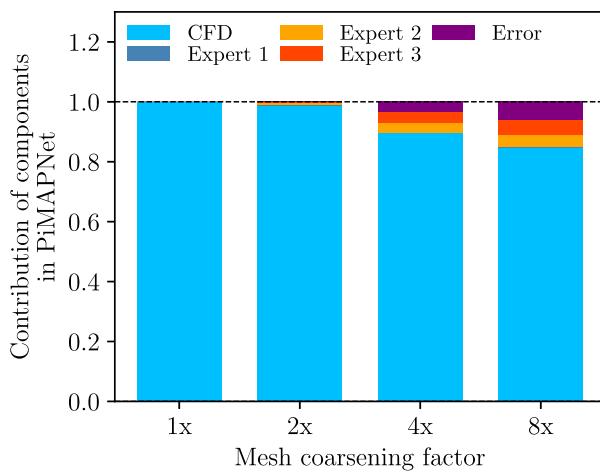
First, we examine the accuracy of PiMAPNet with different levels of coarse-grid simulations ( $2\times$ ,  $4\times$  and  $8\times$ ) in the physics-informed component. The coarse-grid simulations are compared with the PiMAPNet predictions. Fig. 6 shows the instantaneous temperature fields at two different time instances:  $N_{ML} = 200$  (part of training) and  $N_{ML} = 400$  (inference beyond the training time step). The first column shows the GT, and the following six columns correspond to the prediction of LR2x, PiMAPNet2x, LR4x, PiMAPNet4x, LR8x and PiMAPNet8x, respectively. The first row for each time instant depicts the temperature fields, and the second row shows the absolute error of each prediction compared to GT. The accuracy of the numerical simulation decreases as the

grid resolution reduces from  $2\times$  to  $8\times$ , clearly visible in the absolute error fields in Fig. 6. For all cases, their corresponding PiMAPNet counterpart reduces the pointwise absolute error. Furthermore, for ‘LR4x’ and ‘LR8x’, the grid resolution cannot capture the unsteadiness of the flame, and the vortices are damped out in the coarse simulation. However, for PiMAPNet4x, the ML model significantly improves the flow characteristics.

To quantify the improvement observed from PiMAPNet, the evolution of NRMSE (Eq. (17)) of pointwise error and error in mean quantities are plotted in Fig. 7 for  $Re_j = 5000$ . The data up to  $N_{ML} = 200$  is included in the training of the model, which is shown by the gray shaded area in Fig. 7. For all cases, after an initial error accumulation,



**Fig. 7.** Error propagation in time for the prediction of magnitude of velocity (first row), temperature (second row), OH (third row) and  $\text{H}_2$  (fourth row) mass fractions for  $\text{Re}_j = 5000$ . The columns indicate the pointwise error in the left panel and statistical errors (mean) in the right panel. The gray and purple shaded regions correspond to the steps included in the training of PiMAPNet and forecasting (extrapolation), respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)



**Fig. 8.** Contribution of each component of the PiMAPNet architecture in the prediction of mean temperature, normalized by the GT mean temperature at  $\text{Re}_j = 5000$ .

the error plateaus even beyond the training period (purple-shaded region). The first row shows the error propagation of velocity magnitude, which is part of  $\phi_{\text{flow}}$ . At  $\text{Re}_j = 5000$ , as the flow is statistically stationary, the mean velocity is well captured by the LR simulations itself, over which the improvements by PiMAPNet is marginal. The effect of PiMAPNet on  $\phi_{\text{flow}}$  is more prominent for higher  $\text{Re}_j$ , which is discussed later. However, for the thermochemical quantities, the PiMAPNet predictions are significantly more accurate than all LR simulations as the ML model captures the scales, unresolved by LR simulations. In

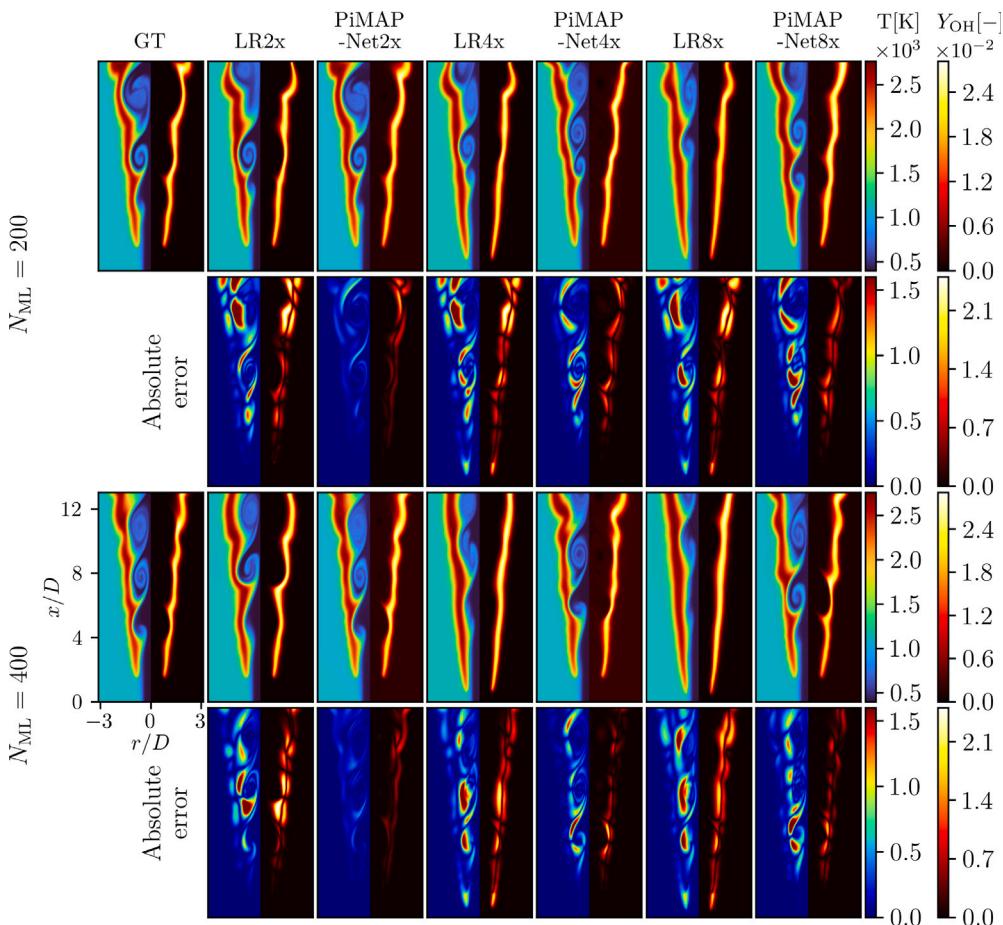
particular, at steady state in ‘LR4x’, the mean OH mass fraction error drops from 25% to 5%, whereas the maximum error in mean temperature is observed to be below 3% for PiMAPNet4x. In comparison, PiMAPNet8x exhibits a larger error than the other PiMAPNet models. Although PiMAPNet8x helps mitigate the numerical dissipation and vortex damping issues seen in LR8x, its improvements are still less effective than those achieved by PiMAPNet4x. This reduction in performance with lower grid resolution aligns with recent findings [12,64] showing similar trends when examining how input resolution affects the accuracy of physics-informed ML models. Overall, the PiMAPNet framework overcomes the typical error accumulation problem observed in using a Baseline-ConvNet model, exhibiting stability and robustness in terms of long-term prediction accuracy.

These results show that PiMAPNet provides better prediction accuracy compared to Baseline-ConvNet, PPNN and LR. To assess the contributions of coarse grid solver and three experts in PiMAPNet to the overall solution, we analyze the mean temperature field over  $N_{\text{ML}} = 200$ , and the contribution of each component is plotted in Fig. 8. The contribution of the coarse grid numerical solver is denoted by ‘CFD’. The ‘Error’ component is the difference between the overall PiMAPNet prediction and GT. At ‘1x’ resolution, CFD component is equivalent to GT prediction without any contribution of MoE. At ‘2x’ resolution, the contribution of ‘CFD’ is 98.5% compared to GT in Fig. 8, resulting in a contribution of 1.5% from the experts and ‘Error’. As the mesh coarsening factor increases, the CFD component deviates further from GT, resulting in decreasing contribution from CFD, and an increasing contribution from each expert in MoE towards GT. Even with the correction from the MoE, the ‘Error’ component increases as well with mesh coarsening, which is consistent with the observations in Fig. 7. For instance, at ‘8x’ resolution, the CFD contribution drops to approximately 89%, with overall MoE contributions of 6% and an ‘Error’ of 5%. The contributions are observed to be qualitatively similar for other quantities of interest as well.

#### 4.3.2. Generalization

In order to be impactful, a well-learned ML model should accurately predict reacting flows outside the training samples. To investigate the generalizability of the PiMAPNet architecture, we perform ML predictions for  $\text{Re}_j = 10,000$  with numerical solvers on 2x, 4x and 8x coarser grids. Fig. 9 shows the instantaneous temperature fields at two time instances for  $\text{Re}_j = 10,000$ , both of which are inference or forecast of the trained models. The observations are similar to the previous case. For the 2x coarser case, the numerical solver is accurate and the PiMAPNet framework does not visibly improve the fields. However, for 4x and 8x coarser cases, PiMAPNet prediction errors are significantly smaller than their numerical solver counterparts. Additionally, as  $\text{Re}_j$  increases from 5000 to 10 000, the flame starts to lift with increasing jet velocity. The flame LOL for the cases of LR2x and PiMAPNet2x are the same as the GT. However, the LOL decreases as the grid resolution reduces to ‘4x’ and ‘8x’. On the contrary, PiMAPNet4x and PiMAPNet8x accurately predict the LOL at all grid resolutions (Fig. 9).

The propagation of NRMSE of mean and RMS of velocity magnitude, temperature, OH and  $\text{H}_2$  mass fractions are plotted in Fig. 10. Unlike  $\text{Re}_j = 5000$ , in this case, as the flow starts to become unsteady, errors in both mean and RMS of velocity are high for LR simulations. For all three levels of coarser resolution simulations, PiMAPNet improves the prediction accuracy of both  $\phi_{\text{flow}}$  and  $\phi_{\text{TC}}$  quantities. Clearly, a single ConvNet for  $\phi_{\text{flow}}$  keeps the solution stable, while used with MoE framework for thermochemical quantities, even for out-of-distribution samples at  $\text{Re}_j = 10,000$ . However, the discrepancy from the GT solution for each quantity increases as the resolution of the physics-informed component of PiMAPNet decreases from 2x to 8x. For instance, at statistically steady state, the errors in the mean temperature are 1.5%, 3% and 7% for PiMAPNet2x, PiMAPNet4x and PiMAPNet8x, respectively.



**Fig. 9.** Instantaneous temperature (left panel) and OH mass fraction (right panel) fields and absolute error in prediction of temperature and OH mass fractions at two different time instants corresponding to  $N_{\text{ML}} = 200$  (top) and  $N_{\text{ML}} = 400$  (bottom) for  $\text{Re}_j = 10,000$ . The first column shows GT. The following columns correspond to predictions of LR and their PiMAPNet counterparts.

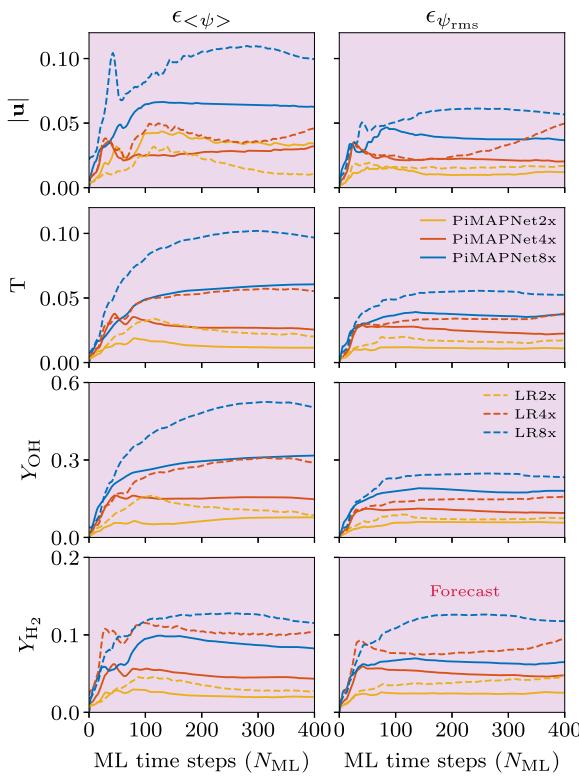
To illustrate the predictions in flame structure, the conditional temperature profiles with mixture fractions is plotted at different axial locations for  $\text{Re}_j = 10,000$  in Fig. 11. The plots for PiMAPNet2x and PiMAPNet4x closely mirror the GT solution. For PiMAPNet8x, the flame structure shows significant deviations from the GT, particularly near the stoichiometric  $Z$ . Specifically, at downstream locations far from the inlet, the PiMAPNet8x model predicts the maximum temperature position to shift towards richer mixtures, accompanied by an additional anomaly at the same  $Z$  location but at a lower temperature. This discrepancy can likely be attributed to artificial mixing between fuel-lean and fuel-rich mixtures introduced by PiMAPNet8x. It should also be noted that the presence of MoE helps to capture these flame structures accurately even after  $400\Delta t_{\text{ML}}$ , whereas for PPNN, the solution becomes unphysical within a few ML timesteps due to the spurious hotspots, as discussed in Section 4.1.

To quantify the difference in predicted LOL between LR simulations and PiMAPNet, we plot the LOL with varying  $\text{Re}_j$  between 5000 and 11,000 in Fig. 12. The gray and purple shaded regions in the plot represent the  $\text{Re}_j$  within and out-of-distributions, respectively. As  $\text{Re}_j$  increases, the LOL increases as well. We compare the LOL from the high-resolution simulation (GT) with the prediction from LR2x, LR4x, LR8x and their PiMAPNet counterparts. LOL is defined as the axial distance from the nozzle exit to 2% of the maximum OH mass fraction [65]. LR2x and all PiMAPNet simulations predict the LOL very well compared to GT. However, LR4x and LR8x predictions diverge significantly from the high-resolution solutions. This shows the importance of having an ML model that captures the missing physics in the low resolution simulations and improve the accuracy to the level of a high resolution numerical solver.

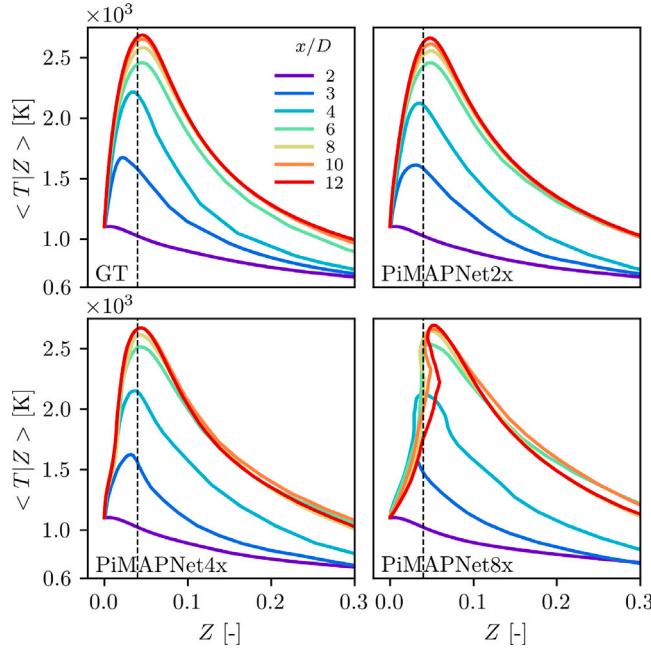
In addition to testing PiMAPNet's generalization to out-of-distribution  $\text{Re}_j$ , we explore two other aspects of its generalization: grid resolution and fuel composition. In Appendix C, we evaluate PiMAPNet's accuracy across different grid resolutions by applying PiMAPNet2x and PiMAPNet4x to LR8x simulations and comparing their predictions with PiMAPNet8x. Furthermore, to examine its ability to handle varying fuel compositions, we reduce the hydrogen content in the fuel stream from 65% to 50%. Both analyses are performed at  $\text{Re}_j = 10,000$ .

#### 4.3.3. Computational efficiency

The primary objective of using PiMAPNet is to gain stable and robust jet flame predictions with good accuracy at appreciable computational cost reductions. We compare the computational gains achieved by PiMAPNet2x, PiMAPNet4x, PiMAPNet8x with respect to the numerical simulation cost of the GT generation in Fig. 13(a). The ordinate is the ratio of each of the PiMAPNet model prediction and the computational cost of GT data generation. Each of the PiMAPNet plots is divided into the cost of the corresponding LR simulation, denoted by 'CFD', and the cost of predictions from the MoE-based ML component, labeled as 'PiMAPNet'. The cost is measured for  $\text{Re}_j = 10,000$  over  $N_{\text{ML}} = 400$ . Due to the fast inference speed of neural networks, the PiMAPNet predictions show significant speedup compared to the high-resolution numerical solvers. The inference cost of the ML part is approximately 15% of the GT data generation, which is independent of the grid resolution in the LR simulation. In contrast, as mesh resolution decreases from  $2\times$  to  $8\times$ , the cost of the CFD component keeps decreasing, as expected. Combining these two costs, the speedups

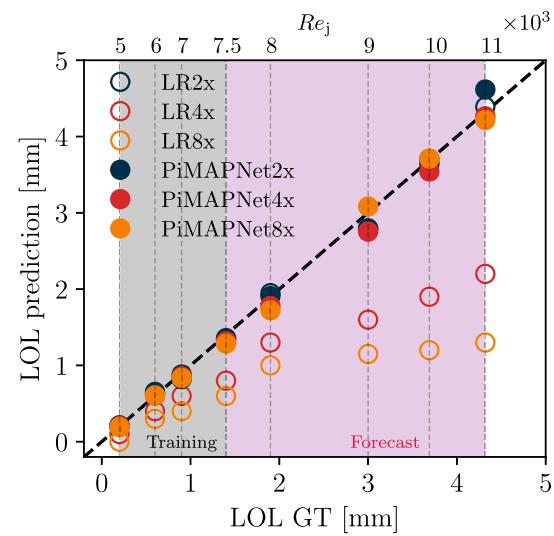


**Fig. 10.** Error propagation in time for the prediction of magnitude of velocity (first row), temperature (second row), OH (third row) and  $H_2$  (fourth row) mass fractions for  $Re_j = 10,000$ . The columns indicate the statistical errors (mean and RMS), respectively.



**Fig. 11.** Conditional mean temperature profiles with respect to mixture fraction over time at different axial locations for  $Re_j = 10,000$  corresponding to GT and three PiMAPNet models.

are approximately 3 $\times$ , 10 $\times$  and 45 $\times$  for PiMAPNet2x, PiMAPNet4x and PiMAPNet8x, respectively. While these speedups are significant, an accurate comparison also considers speedup at equivalent accuracy levels [66]. Fig. 13(b) shows the mean NRMSE error in temperature and  $H_2$  mass fractions for LR and PiMAPNet simulations with varying



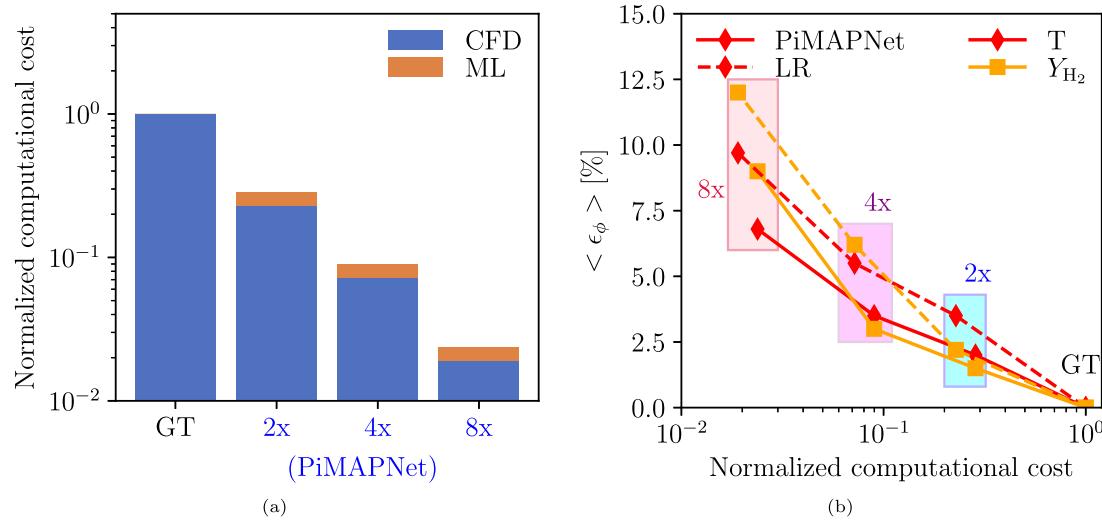
**Fig. 12.** Comparison of predicted LOL by different models compared to GT at varying  $Re_j$  between 5000 and 11,000. The gray and purple regions correspond to  $Re_j$  within and out-of-distribution, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

normalized computational cost. The reference points at a normalized cost of unity and zero error correspond to the GT. As the normalized cost decreases with decreasing grid resolution, the error increases for both LR and PiMAPNet. Nevertheless, the overall errors observed in PiMAPNet predictions remain within 10% even for the lowest grid resolution. More importantly, at a given error level, PiMAPNet shows lower normalized cost compared to LR. For example, at a 5% error in temperature and  $H_2$  mass fraction, PiMAPNet achieves approximately 3 $\times$  speedup over LR predictions. Similarly, at given normalized cost, PiMAPNet exhibits higher accuracy than LR. The magnitude of the overall speedups will increase further when considering longer model rollouts and numerous repeated model queries across a wide range of  $Re_j$ , which are often necessary in applications involving design optimization and inverse problems.

## 5. Conclusions

We introduce PiMAPNet, a physics-informed ML approach designed for spatiotemporal forecasting in multiphysics simulations. This model architecture leverages outputs from a coarser grid simulation and integrates them into a CNN-based architecture, establishing a physics-constrained ML model. To account for the accumulation of spurious hotspots observed in the prediction of thermochemical state space by a single ML model, we introduce the MoE approach in PiMAPNet, where multiple neural networks are trained, each specialized to represent a specific region of the manifold. This methodology was evaluated on a lifted hydrogen jet flame with finite rate chemistry with varying jet Reynolds numbers.

The analysis across various regions in the thermochemical state-space shows that the MoE strategy effectively decomposes the manifold in a physically meaningful way, distinctly isolating fuel-rich, fuel-lean and mixing compositions. Employing this strategy shows that the predicted solution remains stable and physical unlike purely data-driven models and a single-architecture PIML model. Furthermore, PiMAPNet demonstrates robust generalization capabilities, effectively maintaining reasonable accuracy compared to ground truth even for out-of-distribution Reynolds numbers. In particular, the error in mean temperature prediction is below 5% even with a 4 $\times$  coarser grid simulation for an out-of-distribution  $Re_j$ .



**Fig. 13.** (a) Comparison of the computational cost for PiMAPNet predictions with different low-resolution simulations relative to the high resolution simulation (GT). (b) The mean NRMSE error in temperature and  $H_2$  mass fractions with varying normalized computational cost for LR and PiMAPNet predictions.

Performance analysis shows the model achieves up to 45 $\times$  speedup than the fine-grid numerical solver. Additionally, we show that PiMAPNet shows approximately 3 $\times$  faster inference speed compared to low-resolution numerical simulations at similar accuracy level in multiple thermophysical quantities. The distinct treatment of different physical variables allows this framework to be easily adapted to other multi-physics systems making it a valuable tool for researchers across various domains in computational physics.

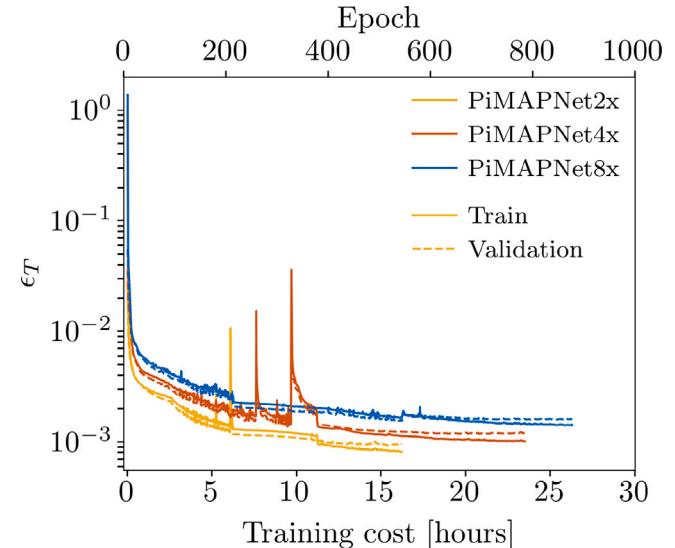
Despite the significant improvements achieved by the PiMAPNet model, it is crucial to acknowledge the potential opportunities for enhancement. This study employs a uniform mesh, which simplifies the use of CNNs. However, to accommodate complex and unstructured meshes, alternatives such as graph neural networks [67] could be employed. A comprehensive analysis of the effect of  $N$  or  $\Delta t_{ML}$  on PiMAPNet predictions is essential in a future study. This study considers statistically stationary data for the training and prediction of PiMAPNet, which helps in long-term stability. Although it is reasonable as a first attempt to spatiotemporal reacting flow prediction using a PIML architecture, the model may require further extensions to capture transient phenomena such as auto-ignition. Other future extensions of the PiMAPNet framework will focus on (i) applying the model to more realistic configurations in 3D geometry, and (iii) implementing adaptive selection of ML models for different regions of the thermochemical manifold to further enhance overall model efficiency [38].

#### CRediT authorship contribution statement

**Pushan Sharma:** Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Wai Tong Chung:** Writing – review & editing, Supervision, Software, Methodology, Conceptualization. **Matthias Ihme:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, 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.



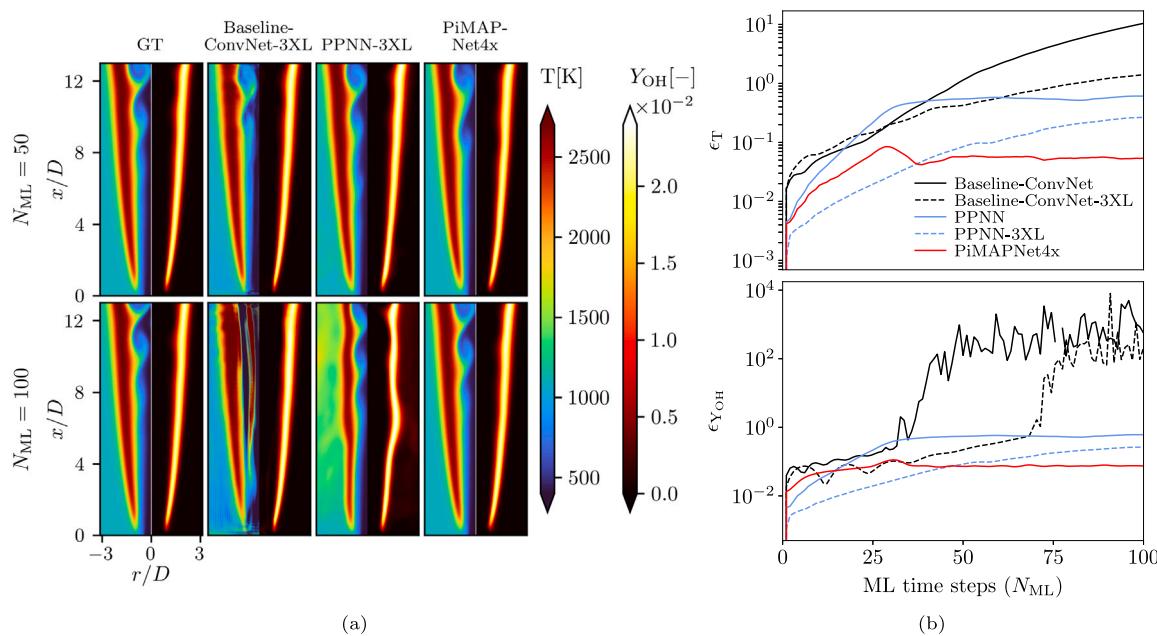
**Fig. A.14.** Computational cost of training PiMAPNets with different levels of coarse grid simulations on an NVIDIA A100 GPU.

#### Acknowledgments

This work was supported by the Department of Energy Office of Energy Efficiency Renewable Energy (EERE) with award number DE-EE0008875. This research used resources of the National Energy Research Scientific Computing Center, a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

#### Appendix A. Cost of training PiMAPNet

**Fig. A.14** illustrates the mean squared errors (MSE) in temperature over the number of training epochs, alongside the corresponding training time in hours. PiMAPNet2x achieves convergence in approximately 600 epochs with training and validation errors of around 0.1%, while PiMAPNet8x requires about 900 epochs and reaches a higher error of 0.4%. This variation in training and validation errors also translate into the prediction phase, where PiMAPNet8x predictions miss some features of the flow physics, such as the unsteadiness observed in case



**Fig. B.15.** (a) Instantaneous temperature (left panel) and OH mass fraction (right panel) fields at two different time instants corresponding to  $N_{\text{ML}} = 50$  and  $N_{\text{ML}} = 100$  with 3x larger model size than Fig. 3, (b) Error propagation in time for the prediction of temperature and OH mass fraction for Baseline-ConvNet(3x), PPNN(3x) and PiMAPNet4x in case of  $\text{Re}_j = 5000$ .

of  $\text{Re}_j = 5000$  (Fig. 7), and exhibits larger error than PiMAPNet2x. The total training time for all PiMAPNet models is approximately 70 h on a single NVIDIA A100 GPU, with the cost ranging between 15 to 30 h for individual model, depending on the LR simulation resolution. For each model, 200 snapshots are used for each of the three  $\text{Re}_j$  for training. Collecting this data for each  $\text{Re}_j$  took approximately 20 h in 8 AMD Milan nodes with 128 CPU cores, resulting in 160 node-hours or  $\sim 20500$  CPU-hours. Since the GT data is available publicly through BLASTNet [12], future ML models trained on these data would not incur the data generation cost.

#### Appendix B. PPNN and Baseline-ConvNet with 3x larger model size

The PiMAPNet architecture utilizes a single ConvNet for the flow variables,  $\phi_{\text{flow}}$ , and employs multiple ConvNets to represent distinct regions of the thermochemical manifold, leveraging a MoE approach. This design results in a significantly larger overall model size for PiMAPNet compared to both PPNN and Baseline-ConvNet, each of which uses only one ConvNet for the input vector,  $\phi$ . To ensure a fair comparison, we increased the model size of PPNN and Baseline-ConvNet by a factor of three to approximate the model size of PiMAPNet, denoted as ‘Baseline-ConvNet-3XL’ and ‘PPNN-3XL’ in Fig. B.15. We then evaluate the predictive accuracy of these two larger models against PiMAPNet4x for  $\text{Re}_j = 5000$ .

Fig. B.15(a) presents instantaneous temperature and OH mass fraction fields at  $N_{\text{ML}} = 50$  and 100, including absolute error plots against ground truth and NRMSE progression for temperature and OH mass fraction in Fig. B.15(b). Increasing model size reduces overall errors for both Baseline-ConvNet and PPNN. However, similar to observations in Fig. 3(a), Baseline-ConvNet-3XL eventually diverges, though the divergence is delayed compared to its smaller counterpart. In contrast, PPNN-3XL shows a distinct divergence pattern. In the short-term window, the prediction accuracy is better than PiMAPNet4x. Despite similarities in the size of both ML models, this behavior can be explained by their differences in the ML architecture. PPNN-3XL consists of a dense ML architecture, while PiMAPNet4x consists of a MoE architecture that combines the prediction of smaller dense networks

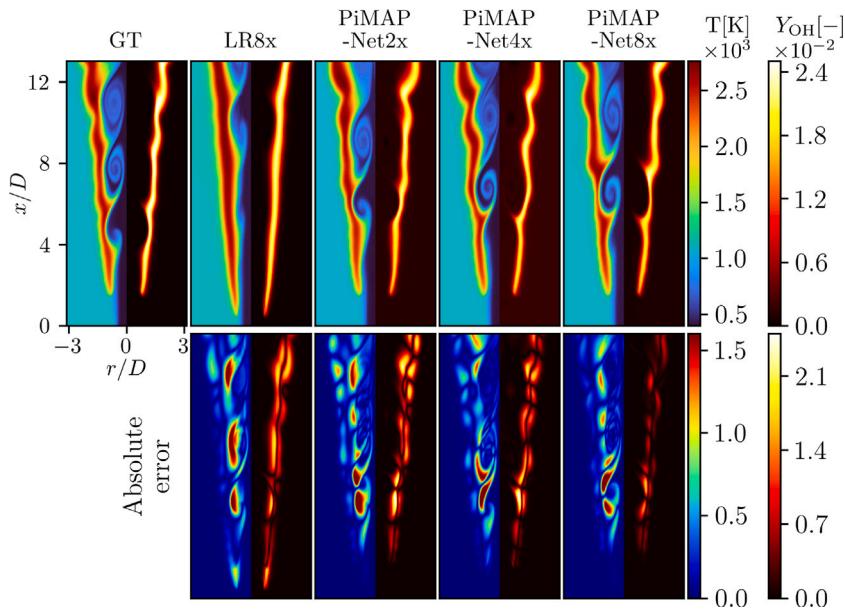
that are specialized to treat different regimes in the flowfield. Hence, PPNN-3XL is more expressive due to the larger dense ML module, and can capture phenomena in finer resolutions [12], resulting in lower errors. However, error reduction alone does not necessarily correlate with physically meaningful predictions. Even at a short time window, PPNN-3XL introduces spurious temperature fluctuations in the coflow, although at a lower magnitude. These spurious hotspots developing away from the flame eventually lead to unphysical results and deviation from the reference solution in long-term rollout. In contrast, the MoE architecture in PiMAPNet allows each of the smaller dense ML component to focus on regions with vastly different physics and offers better stability for long-term rollout compared to both the larger baseline and PPNN models. This suggests that the issues observed in Section 4.1 for PPNN and the baseline model are independent of model size, emphasizing the necessity of the MoE framework for physically accurate, long-term predictions.

#### Appendix C. Generalization of PiMAPNet

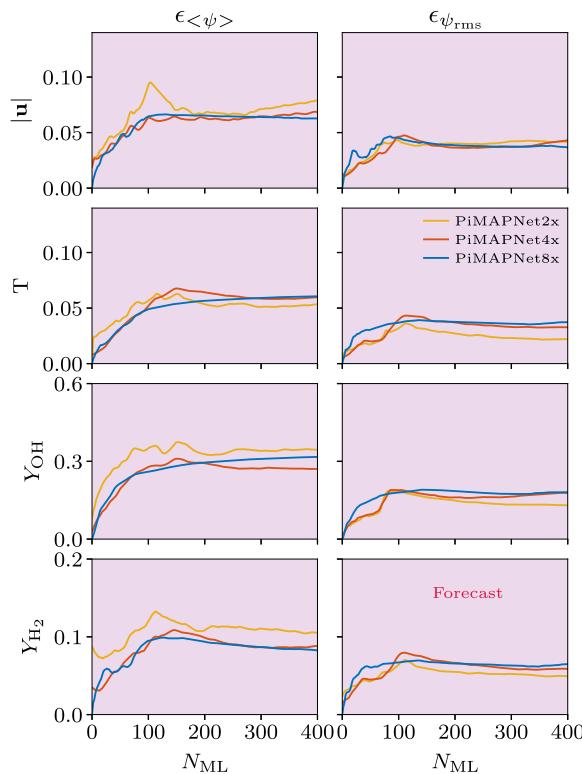
As discussed in Section 4.3.2, well-trained ML models must generalize effectively to out-of-training samples. We examined the generalizability of PiMAPNet by testing its predictions at Reynolds numbers beyond the training range, such as  $\text{Re}_j = 10,000$ . In this section, we further explore PiMAPNet’s generalization capabilities in two additional areas: (1) grid resolution, and (2) fuel composition, where we modify the hydrogen percentage in the fuel jet from its original value of 65% to 50%.

##### C.1. Generalization: grid resolution

To assess the accuracy of PiMAPNet predictions across different grid resolutions, we employ PiMAPNet2x and PiMAPNet4x to predict the evolution of the hydrogen combustion configuration on an 8 $\times$  coarser grid. The goal is to compare the performance of PiMAPNet2x and PiMAPNet4x against PiMAPNet8x in terms of predictive accuracy. Fig. C.16 shows the instantaneous temperature and OH mass fractions at  $N_{\text{ML}} = 400$  corresponding GT, PiMAPNet2x, PiMAPNet4x and PiMAPNet8x. The LR8x solution is also shown for a comprehensive



**Fig. C.16.** Instantaneous temperature (left panel) and OH mass fraction (right panel) fields and absolute error in prediction of temperature and OH mass fraction after  $N_{ML} = 400$ . The first column corresponds to GT. The second column corresponds to numerical simulation on  $8\times$  coarser grid. The next three columns shows the PiMAPNet2x, PiMAPNet4x and PiMAPNet8x predictions with LR8x as the low resolution simulation.



**Fig. C.17.** Error propagation in time for the prediction of magnitude of velocity, temperature, OH and  $H_2$  mass fractions for  $Re_j = 10,000$  for PiMAPNet2x, PiMAPNet4x and PiMAPNet8x while using LR8x as the low resolution simulation.

comparison. The second row shows the absolute errors at the same instant compared to GT. The propagation of NRMSE for all three PiMAPNet predictions are shown in Fig. C.17. From both contour and error plots, a few conclusions can be drawn. All three PiMAPNet models exhibit stable long-term prediction accuracy. The damped vortices in LR8x due to artificial dissipation are accurately captured

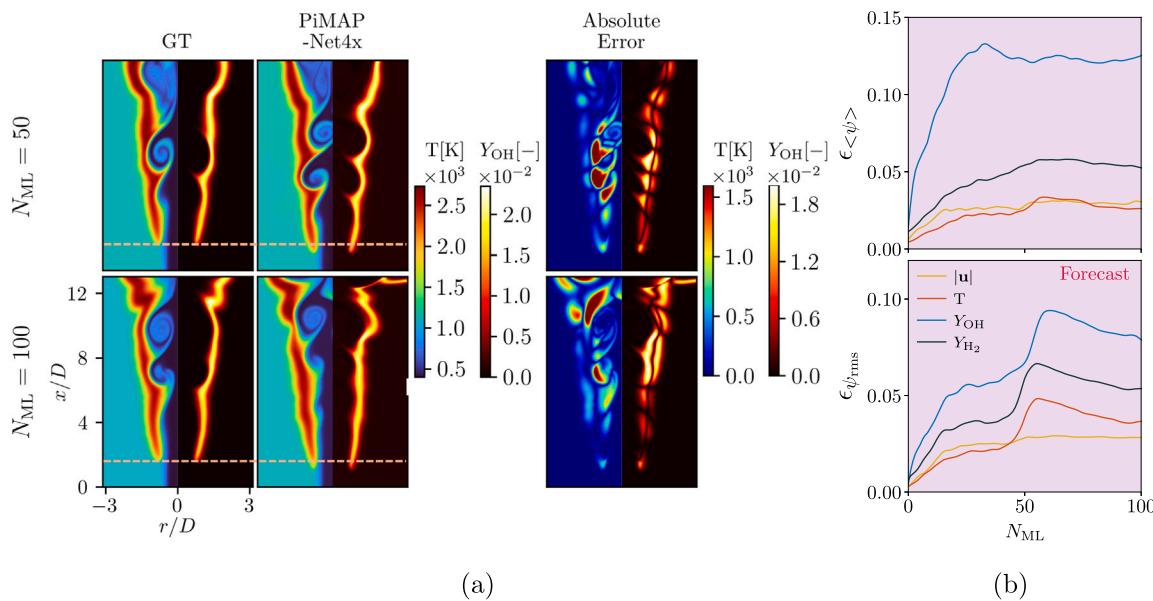
in all PiMAPNet predictions. For LR8x, the lift-off height is underestimated. However, similar to PiMAPNet8x, both PiMAPNet2x and PiMAPNet4x have predicted the lift-off height accurately. In terms of error, PiMAPNet2x incurs slightly larger errors compared to PiMAPNet4x and PiMAPNet8x, especially visible for mean  $H_2$  and OH mass fractions in the first panel of Fig. C.17. The propagation of error in RMS quantities are very similar for all ML models. Overall, the errors remain reasonably small and do not accumulate over time, which demonstrate that PiMAPNet generalizes well across different grid resolutions.

### C.2. Generalization: fuel composition

The other important aspect of generalization in the context of reacting flow simulations is the fuel composition. The PiMAPNet models are trained with 65% hydrogen in the fuel stream. To assess the performance of PiMAPNet with varying fuel compositions, we reduce the hydrogen content to 50% in the fuel stream at  $Re_j = 10,000$  and used PiMAPNet4x to predict the evolution of the reacting jet for  $N_{ML} = 100$  steps. Fig. C.18a shows the instantaneous fields at two different time instants along with the absolute errors compared to GT. The propagation of NRMSE of velocity magnitude, temperature, OH and  $H_2$  mass fractions are plotted in Fig. C.18b. Similar to previous observations, the errors initially increase, but them remains stable after a few initial ML time steps. The maximum observed error in mean temperature approximately 5% as opposed to 3% in case 65% hydrogen. The error for OH mass fractions is relatively higher, but the magnitudes are similar to previous observations (Fig. 10). However, the lift-off height in the ML prediction is underpredicted by 0.5 mm, as shown by the dashed lines in Fig. C.18a. It should be noted this predictions are out-of-distribution for both Reynolds number and fuel composition. Enriching the training data with more than a single fuel composition is expected to improve the model predictions significantly. Overall PiMAPNet predictions are stable and robust for long-time rollout and out-of-distribution samples.

### Data availability

The fine-grid lifted hydrogen jet flame data are available as training data along with the PiMAPNet model architecture in BLASTNet database (<https://blastnet.github.io/>).



**Fig. C.18.** (a) Instantaneous temperature (left panel) and OH mass fraction (right panel) fields and absolute error in prediction of temperature and OH mass fraction after  $N_{ML} = 50$  and 100. The first column corresponds to GT. The second column is the PiMAPNet4x predictions and the third column corresponds to the absolute error. The dashed lines in the contour plots show the discrepancy in the prediction of LOL, (b) Error propagation in time for the prediction of magnitude of velocity, temperature, OH and H<sub>2</sub> mass fractions for  $Re_j = 10,000$  for PiMAPNet4x with 50% H<sub>2</sub> in the fuel stream.

## References

- [1] A.R. Masri, Challenges for turbulent combustion, *Proc. Combust. Inst.* 38 (1) (2021) 121–155.
- [2] S.B. Pope, Small scales, many species and the manifold challenges of turbulent combustion, *Proc. Combust. Inst.* 34 (1) (2013) 1–31.
- [3] H. Wu, Y.C. See, Q. Wang, M. Ihme, A Pareto-efficient combustion framework with submodel assignment for predicting complex flame configurations, *Combust. Flame* 162 (11) (2015) 4208–4230.
- [4] N. Aubry, P. Holmes, J.L. Lumley, E. Stone, The dynamics of coherent structures in the wall region of a turbulent boundary layer, *J. Fluid Mech.* 192 (1988) 115–173.
- [5] J.C. Sutherland, A. Parente, Combustion modeling using principal component analysis, *Proc. Combust. Inst.* 32 (1) (2009) 1563–1570.
- [6] P. Pant, R. Doshi, P. Bahl, A. Barati Farimani, Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations, *Phys. Fluids* 33 (10) (2021).
- [7] S.L. Brunton, B.R. Noack, P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* 52 (2020) 477–508.
- [8] P. Sharma, W.T. Chung, B. Akoush, M. Ihme, A review of physics-informed machine learning in fluid mechanics, *Energies* 16 (5) (2023) 2343.
- [9] M. Ihme, W.T. Chung, A.A. Mishra, Combustion machine learning: Principles, progress and prospects, *Prog. Energy Combust. Sci.* 91 (2022) 101010.
- [10] M. Ihme, W.T. Chung, Artificial intelligence as a catalyst for combustion science and engineering, *Proc. Combust. Inst.* 40 (1–4) (2024) 105730.
- [11] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.* 3 (6) (2021) 422–440.
- [12] W.T. Chung, B. Akoush, P. Sharma, A. Tamkin, K.S. Jung, J. Chen, J. Guo, D. Brouzet, M. Talei, B. Savard, A. Poludnenko, M. Ihme, Turbulence in focus: Benchmarking scaling behavior of 3D volumetric super-resolution with BLASTNet 2.0 data, *Adv. Neur. Inf. Process. Syst.* 36 (2024).
- [13] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686–707.
- [14] M. Raissi, Z. Wang, M.S. Triantafyllou, G.E. Karniadakis, Deep learning of vortex-induced vibrations, *J. Fluid Mech.* 861 (2019) 119–137.
- [15] C.M. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, K. Kashinath, M. Mustafa, H.A. Tchelepi, P. Marcus, M. Prabhat, A. Anandkumar, MeshfreeFlowNet: A physics-constrained deep continuous space-time super-resolution framework, SC20: Int. Conf. High Perform. Comput. Netw. Storage Anal. (2020) 1–15.
- [16] X. Sun, W. Cao, Y. Liu, L. Zhu, W. Zhang, High Reynolds number airfoil turbulence modeling method based on machine learning technique, *Comp. Fluid.* 236 (2022) 105298.
- [17] Y. Guan, A. Subel, A. Chattopadhyay, P. Hassanzadeh, Learning physics-constrained subgrid-scale closures in the small-data regime for stable and accurate LES, *Phys. D* 443 (2023) 133568.
- [18] M. Bode, M. Gauding, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, H. Pitsch, Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows, *Proc. Combust. Inst.* 38 (2) (2021) 2617–2625.
- [19] S. Liu, H. Wang, Z. Sun, K.K. Foo, G.J. Nathan, X. Dong, M.J. Evans, B.B. Dally, K. Luo, J. Fan, Reconstructing soot fields in acoustically forced laminar sooting flames using physics-informed machine learning models, *Proc. Combust. Inst.* 40 (1–4) (2024) 105314.
- [20] R. Wang, K. Kashinath, M. Mustafa, A. Albert, R. Yu, Towards physics-informed deep learning for turbulent flow prediction, in: 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 1457–1466.
- [21] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, 2020, arXiv preprint arXiv:2010.08895.
- [22] J. Pathak, M. Mustafa, K. Kashinath, E. Motheau, T. Kurth, M. Day, Using machine learning to augment coarse-grid computational fluid dynamics simulations, 2020, arXiv preprint arXiv:2010.00072.
- [23] K. Um, R. Brand, Y.R. Fei, P. Holl, N. Thurey, Solver-in-the-loop: Learning from differentiable physics to interact with iterative PDE-solvers, *Adv. Neural Inf. Process. Syst.* 33 (2020) 6111–6122.
- [24] D. Kochkov, J.A. Smith, A. Alieva, Q. Wang, M.P. Brenner, S. Hoyer, Machine learning-accelerated computational fluid dynamics, *Proc. Natl. Acad. Sci.* 118 (21) (2021) e2101784118.
- [25] X.-Y. Liu, M. Zhu, L. Lu, H. Sun, J.-X. Wang, Multi-resolution partial differential equations preserved learning framework for spatiotemporal dynamics, *Commun. Phys.* 7 (1) (2024) 31.
- [26] H. Wang, Y. Cao, Z. Huang, Y. Liu, P. Hu, X. Luo, Z. Song, W. Zhao, J. Liu, J. Sun, S. Zhang, L. Wei, Y. Wang, T. Wu, Z.-M. Ma, Y. Sun, Recent advances on machine learning for computational fluid dynamics: A survey, 2024, arXiv preprint arXiv:2408.12171.
- [27] R. Ranade, S. Alqahtani, A. Farooq, T. Echekki, An ANN based hybrid chemistry framework for complex fuels, *Fuel* 241 (2019) 625–636.
- [28] W. Ji, S. Deng, Autonomous discovery of unknown reaction pathways from data by chemical reaction neural network, *J. Phys. Chem. A* 125 (4) (2021) 1082–1092.
- [29] K. Wan, S. Hartl, L. Vervisch, P. Domingo, R.S. Barlow, C. Hasse, Combustion regime identification from machine learning trained by Raman/Rayleigh line measurements, *Combust. Flame* 219 (2020) 268–274.
- [30] B.A. Sen, S. Menon, Turbulent premixed flame modeling using artificial neural networks based chemical kinetics, *Proc. Combust. Inst.* 32 (1) (2009) 1605–1611.
- [31] O. Owoyele, P. Pal, ChemNODE: A neural ordinary differential equations framework for efficient chemical kinetic solvers, *Energy AI* 7 (2022) 100118.
- [32] A. Chatzopoulos, S. Rigopoulos, A chemistry tabulation approach via rate-controlled constrained equilibrium (RCCE) and artificial neural networks (ANNs), with application to turbulent non-premixed CH<sub>4</sub>/H<sub>2</sub>/N<sub>2</sub> flames, *Proc. Combust. Inst.* 34 (1) (2013) 1465–1473.

- [33] A. Almeldein, N. Van Dam, Accelerating chemical kinetics calculations with physics informed neural networks, *J. Eng. Gas Turb. Power* 145 (9) (2023) 091008.
- [34] M. Ihme, C. Schmitt, H. Pitsch, Optimal artificial neural networks and tabulation methods for chemistry representation in LES of a bluff-body swirl-stabilized flame, *Proc. Combust. Inst.* 32 (1) (2009) 1527–1535.
- [35] Y. Zhang, S. Xu, S. Zhong, X.-S. Bai, H. Wang, M. Yao, Large eddy simulation of spray combustion using flamelet generated manifolds combined with artificial neural networks, *Energy AI* 2 (2020) 100021.
- [36] O. Owoyele, P. Kundu, P. Pal, Efficient bifurcation and tabulation of multi-dimensional combustion manifolds using deep mixture of experts: An a priori study, *Proc. Combust. Inst.* 38 (4) (2021) 5889–5896.
- [37] R. Mishra, S. Mayilvahanan, D. Jarrahbashi, Hybrid unsupervised cluster-wise regression approach for representing the flamelet tables, *Energy Fuels* 37 (4) (2023) 3056–3070.
- [38] W.T. Chung, A.A. Mishra, N. Perakis, M. Ihme, Data-assisted combustion simulations with dynamic submodel assignment using random forests, *Combust. Flame* 227 (2021) 172–185.
- [39] J. An, H. Wang, B. Liu, K.H. Luo, F. Qin, G.Q. He, A deep learning framework for hydrogen-fueled turbulent combustion simulation, *Int. J. Hydrg. Energy* 45 (35) (2020) 17992–18000.
- [40] S. Wu, H. Wang, K.H. Luo, A robust autoregressive long-term spatiotemporal forecasting framework for surrogate-based turbulent combustion modeling via deep learning, *Energy AI* (2023) 100333.
- [41] G. D'Alessio, A. Cuoci, A. Parente, Feature extraction and artificial neural networks for the on-the-fly classification of high-dimensional thermochemical spaces in adaptive-chemistry simulations, *Data-Centric Eng.* 2 (2021) e2.
- [42] M. Liukkonen, T. Hiltunen, E. Hälikkä, Y. Hiltunen, Modeling of the fluidized bed combustion process and NO<sub>x</sub> emissions using self-organizing maps: An application to the diagnosis of process states, *Env. Model. Softw.* 26 (5) (2011) 605–614.
- [43] W. Yu, F. Zhao, W. Yang, H. Xu, Integrated analysis of CFD simulation data with K-means clustering algorithm for soot formation under varied combustion conditions, *Appl. Therm. Eng.* 153 (2019) 299–305.
- [44] J.A. Blasco, N. Fueyo, C. Dopazo, J.-Y. Chen, A self-organizing-map approach to chemistry representation in combustion applications, *Combust. Theor. Model.* 4 (1) (2000) 61–76.
- [45] L.L. Franke, A.K. Chatzopoulos, S. Rigopoulos, Tabulation of combustion chemistry via artificial neural networks (ANNs): Methodology and application to LES-PDF simulation of sydney flame I, *Combust. Flame* 185 (2017) 245–260.
- [46] H.-T. Nguyen, P. Domingo, L. Vervisch, P.-D. Nguyen, Machine learning for integrating combustion chemistry in numerical simulations, *Energy AI* 5 (2021) 100082.
- [47] M.R. Malik, A. Coussement, T. Echekki, A. Parente, Principal component analysis based combustion model in the context of a lifted methane/air flame: Sensitivity to the manifold parameters and subgrid closure, *Combust. Flame* 244 (2022) 112134.
- [48] P.C. Ma, H. Wu, T. Jaravel, L. Bravo, M. Ihme, Large-eddy simulations of trans-critical injection and auto-ignition using diffuse-interface method and finite-rate chemistry, *Proc. Combust. Inst.* 37 (3) (2019) 3303–3310.
- [49] W.T. Chung, A.A. Mishra, M. Ihme, Interpretable data-driven methods for subgrid-scale closure in LES for transcritical LOX/GCH<sub>4</sub> combustion, *Combust. Flame* 239 (2022) 111758.
- [50] P.C. Ma, Y. Lv, M. Ihme, An entropy-stable hybrid scheme for simulations of transcritical real-fluid flows, *J. Comput. Phys.* 340 (2017) 330–357.
- [51] Y. Khalighi, F. Ham, J. Nichols, S. Lele, P. Moin, Unstructured Large Eddy Simulation for Prediction of Noise Issued from Turbulent Jets in Various Configurations, *AIAA Paper* 2011-2886, 2011.
- [52] H. Wu, P.C. Ma, M. Ihme, Efficient time-stepping techniques for simulating turbulent reactive flows with stiff chemistry, *Comput. Phys. Comm.* 243 (2019) 81–96.
- [53] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability-preserving high-order time discretization methods, *SIAM Rev.* 43 (1) (2001) 89–112.
- [54] R.A. Jacobs, M.I. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, *Neural Comput.* 3 (1) (1991) 79–87.
- [55] W. Fedus, B. Zoph, N. Shazeer, Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity, *J. Mach. Learn. Res.* 23 (120) (2022) 1–39.
- [56] C.S. Yoo, R. Sankaran, J. Chen, Three-dimensional direct numerical simulation of a turbulent lifted hydrogen jet flame in heated coflow: Flame stabilization and structure, *J. Fluid Mech.* 640 (2009) 453–481.
- [57] J. Li, Z. Zhao, A. Kazakov, F.L. Dryer, An updated comprehensive kinetic model of hydrogen combustion, *Int. J. Chem. Kinetics* 36 (10) (2004) 566–575.
- [58] A. Alieva, S. Hoyer, M. Brenner, G. Iaccarino, P. Norgaard, Toward accelerated data-driven Rayleigh-Bénard convection simulations, *Eur. Phys. J. E* 46 (7) (2023) 64.
- [59] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: 13th International Conference on Artificial Intelligence and Statistics, 2010.
- [60] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [61] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., PyTorch: An imperative style, high-performance deep learning library, *Adv. Neural Inf. Process. Syst.* 32 (2019).
- [62] K. Stachenfeld, D.B. Fielding, D. Kochkov, M. Cranmer, T. Pfaff, J. Godwin, C. Cui, S. Ho, P. Battaglia, A. Sanchez-Gonzalez, Learned coarse models for efficient turbulence simulation, 2021, arXiv preprint [arXiv:2112.15275](https://arxiv.org/abs/2112.15275).
- [63] R. Bilger, The structure of turbulent nonpremixed flames, *Symp. (Int.) Combust.* 22 (1) (1989) 475–488.
- [64] Z. Li, H. Zheng, N. Kovachki, D. Jin, H. Chen, B. Liu, K. Azizzadenesheli, A. Anandkumar, Physics-informed neural operator for learning partial differential equations, *ACM/ JMS J. Data Sci.* 1 (3) (2024) 1–27.
- [65] J. Guo, D. Brouzet, W.T. Chung, M. Ihme, Analysis of ducted fuel injection at high-pressure transcritical conditions using large-eddy simulations, *Int. J. Engine Res.* 25 (2) (2024) 305–319.
- [66] N. McGreivy, A. Hakim, Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations, *Nat. Mach. Intell.* (2024) 1–14.
- [67] H. Gao, M.J. Zahr, J.-X. Wang, Physics-informed graph neural Galerkin networks: A unified framework for solving PDE-governed forward and inverse problems, *Comp. Methods Appl. Mech.* 390 (2022) 114502.