FIREMARK: A BENCHMARK DATASET FOR DATA-DRIVEN FORECASTING OF LAMINAR FLAME DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

High-fidelity computational fluid-dynamics (CFD) simulations of reacting flows are computationally expensive, limiting their usage in real-time applications. In this context, machine learning (ML) methods offer a promising approach for predicting dynamic flow evolution. However, evaluating them remains an open challenge due to the limited availability of benchmarks that combine high temporal resolution with a diverse range of exogenous inputs. In this work, we introduce FireMark, an open benchmark for evaluating data-driven methods to forecast reacting flows evolution. The benchmark contains multiple simulations of a non-premixed laminar methane/nitrogen coflow flame subjected to time-varying inlet velocities. FireMark includes both thermodynamic and chemical fields, obtained through CFD simulations with a detailed chemical kinetic mechanism. The benchmark is designed to assess the ability of forecasting models to reproduce the evolution of a laminar non-premixed flame and to generalize to unseen input signals. Training data consist of sine-sweep forcing signals at two amplitudes, whereas the test set comprises fixed-frequency sinusoids and step functions, with amplitudes both within and outside the training range. To establish reference performance and demonstrate the benchmark capabilities, we evaluate representative classical reduced-order and ML-based forecasting approaches using a set of quantitative metrics, including measures of physical consistency. A linear or bilinear autoregressive model on a reduced basis is more accurate than every neural forecaster on the same latent space and reproduces the heat-release response best; several neural forecasters fall below the accuracy of a constant initial state predictor; and the Fourier neural Operator (FNO), which has the lowest field error, reproduces the integrated heat release worse than that constant forecast. These results provide a baseline performance against which future forecasting methods can be compared. FireMark is publicly available and provides a standardized framework for the reproducible evaluation of data-driven models for reacting-flow dynamics.

1 INTRODUCTION

The study of combustion systems remains highly relevant today, supporting the ongoing energy transition toward zero-emission propulsion and power generation. Computational fluid dynamics (CFD), a branch of fluid mechanics, allows for the accurate study of reacting flows and is widely used in both research and industrial applications. The computational cost of CFD simulations is substantial in terms of both computing resources and time, and this is further amplified in combustion, where the coupling between thermodynamics and chemical kinetics must be resolved. ML-based reduced-order models (ROMs) have therefore shown great potential to accelerate CFD, enabling applications such as digital twinning and real-time control. Developing and validating such forecasting models requires benchmarks that capture the temporal evolution of combustion systems under varying inputs, a resource that still remains largely unavailable for reacting flows.

A broad range of benchmarks have been developed to support the development of scientific ML models applied to CFD. However, existing resources typically offer either fine spatial flame resolution with a rich set of features or fine temporal resolution. To the best of authors' knowledge, there is no public dataset that simultaneously combines spatially resolved flame fields, long and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

finely sampled trajectories, and multiple dynamic inputs. Moreover, as highlighted by Ihme et al. (2022), a systematic assessment of well-defined benchmark cases for data-driven methodologies in combustion is still missing, and this is even more true for forecasting of reactive flow dynamics.

This work addresses this need by introducing FireMark, a benchmark for data-driven forecasting ROMs of reacting flows. Specifically, our contributions are the following.

- **Dataset.** Eight simulations of a forced laminar non-premixed flame, generated using sinusoidal and step excitations spanning a broad range of frequencies and amplitudes. Each simulation provides long trajectories of eleven thermochemical and flow fields on a fixed spatio-temporal discretizations, together with the corresponding input excitation formulas.
- **Evaluation protocol.** The eight simulations are partitioned in training and test sets such that the models are evaluated on fixed excitation frequencies and amplitudes both seen and unseen during training, assessing the generalization and robustness of the considered baselines. For a fair comparison, each baseline is tuned using Bayesian hyperparameter optimization using the same number of optimization trials per learnable component, and the test evaluation results report statistics over three training seeds.
- **Findings.** Within the same latent representation family, a linear autoregressive model beats every neural sequence forecaster in field error, and closed-form linear and bilinear models on a linear basis give the most accurate heat-release response. Several neural forecasters, including recurrent ones, do worse than simply keeping the initial steady field, and the neural operator, although it has the lowest field error, predicts the integrated heat release less accurately than that constant forecast.

2 Related Works

A broad range of benchmark datasets has been developed to support scientific machine learning (SciML) for CFD, yet none combines fine spatial and temporal resolution with multiple dynamic inputs for reacting flows.

Large turbulence databases such as Johns Hopkins Turbulence Database (JHTDB) (Perlman et al., 2007; Li et al., 2008) offers long, finely sampled trajectories, but only for non-reactive flows with few variables, and are designed for turbulence analysis or modal decomposition rather than forecasting. General-purpose benchmarks such as PDEBench (Takamoto et al., 2022), CFDBench (Luo et al., 2023), APEBench (Koehler et al., 2024), and The Well (Ohana et al., 2024) provide long, multi-condition PDE trajectories, but again for non-reactive systems with a limited variable set. Combustion-specific resources, such as the TNF flame library (`https://tnfworkshop.org`) and the Sandia flame databases (Barlow & Frank, 1998), instead offer a rich thermochemical description, but are largely statistically stationary or single-shot, lacking the long, densely sampled trajectories required for forecasting. BLASTNet (Chung et al., 2022; 2023) extends this with a 28 TB multi-variable collection of reacting and non-reacting simulations, but its use cases center on spatial super-resolution from single snapshots, not on temporal forecasting under a dynamic input. A systematic benchmark for data-driven forecasting of flow dynamics in combustion is still missing: existing resources are limited in temporal resolution, spatial/chemical richness, or input diversity, but never combine all three, which is precisely the gap FireMark addresses.

Alongside these datasets, a range of reduced-order and machine-learning methodologies have been developed to model flow dynamics. Classical approaches typically combine dimensionality reduction with a reduced dynamical model: Proper Orthogonal Decomposition (POD) remains a standard linear compression technique, while Dynamic Mode Decomposition (DMD) and Operator Inference (OpInf) are widely used for temporal evolution (Swischuk et al., 2020; McQuarrie et al., 2021; Towne et al., 2023), and input–output formulations such as autoregressive with exogenous input (ARX) combined with POD (Wang et al., 2018) identify low-order dynamical responses from broadband excitation data. For nonlinear dimensionality reduction, convolutional and variational autoencoders (CAE, VAE) improve latent-space representations, paired with nonlinear forecasters such as long short-term memory (LSTM) networks (Mohan & Gaitonde, 2018) and Transformers (Hemmasian & Barati Farimani, 2023). More broadly, neural operators such as the Fourier neural operator (FNO) (Li et al., 2021) have become popular in general-purpose PDE benchmarks (Takamoto et al., 2022; Luo et al., 2023), reflecting a shift from explicit reduced-order dynamics toward end-to-end

field-to-field prediction. We adopt several of these established methods as baselines in Section 5, evaluated here for the first time on a forced reacting-flow forecasting task.

3 DATASET DESCRIPTION

The dataset consists of multiple high-fidelity CFD simulations of an axisymmetric, time-varying, non-premixed laminar methane/nitrogen coflow flame subjected to time-dependent perturbations of the fuel-inlet velocity. In addition to the simulation results, the dataset provides the corresponding computational mesh, enabling the spatial structure of the simulated flow field to be interpreted together with its temporal evolution.

The configuration considered in FireMark has previously been investigated both numerically and experimentally by Mohammed et al. (1998) and D'Alessio et al. (2020). The computational configuration and mesh adopted in the present dataset are based on the setup reported by D'Alessio et al. (2020).

The fuel consists of a nitrogen-diluted methane mixture containing 65% CH_4 and 35% N_2 on a molar basis, while the oxidizer is air. Both streams are supplied at ambient temperature and atmospheric pressure. The CH_4/N_2 mixture enters through a circular nozzle with an internal diameter of 4 mm and a wall thickness of 0.38 mm. The coflow air enters through an annular region with an internal diameter of 50 mm. The computational domain is two-dimensional and axisymmetric, with dimensions of 54 mm in the radial direction and 120 mm in the axial direction. The domain is discretized using a Cartesian mesh containing approximately 21,000 cells.

The simulations were performed using `laminarSMOKE` (Cuoci et al., 2013), a CFD solver for laminar reacting flows with detailed chemical kinetic mechanisms developed within the OpenFOAM and OpenSMOKE++ frameworks (Cuoci et al., 2009; 2015). The solver employs an operator-splitting approach in which each time step is advanced through a transport and a chemical sub-step. The transport sub-step accounts for convection and diffusion, whereas the chemical sub-step advances the chemical source terms. A reduced version of the detailed C1C3 high-temperature chemical kinetic mechanism developed by the CRECK Modeling group is used in all simulations (Ranzi et al., 2012). The mechanism contains 82 species and 1,698 reactions.

The transient behavior of the flame is generated through time-dependent perturbations of the fuel-inlet velocity. The radial fuel velocity is prescribed as

$$v_r(r,t) = v_{\max}\left(1 - \frac{r^2}{R^2}\right)\phi(t), \tag{1}$$

where r is the radial coordinate, R is the internal radius of the fuel nozzle, $v_{\max} = 70$ cm/s is the maximum baseline fuel velocity, and $\phi(t)$ is a dimensionless time-dependent excitation function. The coflow air is injected at a constant velocity of 35 cm/s (Mohammed et al., 1998; D'Alessio et al., 2020).

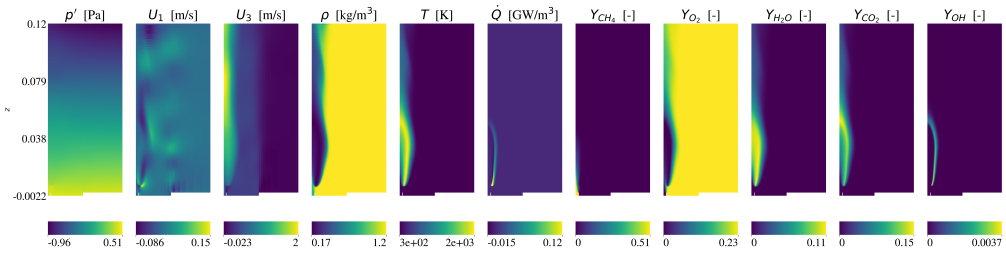

Figure 1: The eleven fields of one snapshot of the test case with a 10 Hz sine of amplitude $A = 0.5$, at $t = 0.275$ s. Each panel shows the 2D computational domain, with the fuel nozzle at the bottom left and the axis of symmetry along the left edge; white pixels lie outside the domain. p' is the deviation of the pressure from its spatial mean. The color scale of $\dot{Q}$ is clipped at its 99.5th percentile, since its peak is confined to a few cells.

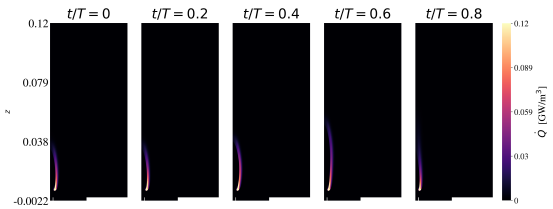

Figure 2: Heat-release rate $\dot{Q}$ over one forcing period T of the same test case (10 Hz, $A = 0.5$), starting at $t = 0.2$ s, on a shared color scale clipped at the 99.5th percentile. The forcing periodically stretches the flame downstream and lets it contract again.

3.1 Excitation Signals

A central feature of FireMark is the use of time-dependent inlet perturbations to generate a diverse set of transient flame responses. The dataset contains three classes of excitation signals: sinusoidal signals, sine-sweep signals, and step signals. For sinusoidal excitation, the inlet perturbation is defined as

$$\phi(t) = 1 + A\sin(2\pi f t), \tag{2}$$

where A is the dimensionless perturbation amplitude and f is the excitation frequency. For sine-sweep excitation, the frequency varies continuously in time according to

$$\phi(t) = 1 + A\sin\left(2\pi \int_0^t f(\tau), d\tau\right), \tag{3}$$

where $f(t)$ varies continuously from 1 Hz to 80 Hz and back to 1 Hz over a total duration of 2 s. These excitations provide a broad range of forcing frequencies within a single simulation and are therefore used to characterize the transient response of the flame over a wide frequency range. For step excitation, the perturbation is defined as

$$\phi_{\text{step}}(t) = \begin{cases} 1, & t < t_0, \\ 1 + A, & t \geq t_0, \end{cases} \tag{4}$$

where $t_0 = 0$ is the time at which the perturbation is applied. The complete set of excitation signals included in FireMark is summarized in Table 1 and shown in Appendix A. It comprises two sine-sweep simulations, two step simulations, and four fixed-frequency sinusoidal simulations.

Table 1: Excitation signals included in the FireMark dataset.

Excitation	Duration	Frequency	Amplitudes A
Sine sweep	2 s	1–80–1 Hz	0.2, 0.4
Step	1 s	–	0.3, 0.5
Sine	1 s	10 Hz	0.3, 0.5
Sine	1 s	40 Hz	0.3, 0.5

The simulations are sampled with a temporal resolution of 0.5 ms. The resulting high temporal resolution is intended to enable the study of transient flame dynamics and to support the training and evaluation of data-driven forecasting methods.

3.2 Dataset Variables

For each simulation, the dataset provides the temporal evolution of eleven features, chosen to give a balanced description of the flame's fluid-dynamic and chemical state. The former are the primary state variables of the reacting flow, and comprise pressure P, the velocity components U_x and U_z, density ρ, temperature T, and volumetric heat release rate $\dot{Q}_{vol}$. The latter are the mass fractions of five species selected for their distinct role in the flame: CH_4 as the fuel, O_2 as main component of the air coflow, H_2O as a main combustion product, CO_2 as a key pollutant, and OH as a radical that is challenging to model. The diluent N_2 is omitted, as it is chemically inert and is not relevant in

this case. Figure 1 shows one snapshot of all fields, and Figure 2 the evolution of $\dot{Q}$ over a forcing period.

The complete dataset, including the CFD fields and corresponding mesh information, is publicly available through the FireMark repository in Zenodo by Anonymous (2026).

3.3 Benchmark Organization and Data Split

The eight simulations listed in Table 1 are organized into training (two simulations) and test (six simulations) sets to assess both forecast accuracy and generalization to unseen forcing conditions. The two sine-sweep simulations, with amplitudes $A = 0.2$ and $A = 0.4$, constitute the training data. Fixed-frequency sinusoidal and step excitations are reserved for testing. This split is designed to train the models across a broad frequency range and to test their performance at specific forcing frequencies and amplitudes, both in within and outside the intervals observed during training. The formulation for the precise forecasting task is defined in Section 4.

Preprocessing. Each field value is standardized with a single global (mean,std) pair computed across all cells and snapshots of the training trajectories; the same statistics are applied to the test data, and all evaluation metrics are computed after inverting this scaling in physical units. Since the mesh is planar, Cartesian and fixed, each cell center is mapped to exactly one pixel of a regular $H \times W$ image without interpolation; pixels with no corresponding cell are masked and ignored by every loss and metric. The physical cell volumes are retained for volume integrals such as the total heat-release rate.

4 Problem formulation

Forced dynamical system. Let $\mathbf{x}_t \in \mathbb{R}^{F \times N}$ denote the flame state at time $t \cdot \Delta t$, with $\Delta t = 0.5$ ms, collecting the $F = 11$ fields of the dataset (P, U_x, U_z, ρ, T, $\dot{Q}$ and the mass fractions of CH_4, O_2, H_2O, CO_2, OH) on the $N \approx 21{,}000$ cells of the mesh. The only exogenous input is the scalar excitation, which we express as its deviation from the unforced flow, $u_t = \phi(t\,\Delta t) - 1$, so that $u_t = 0$ corresponds to the steady flame. The CFD solver defines an unknown forced map $\mathbf{x}_{t+1} = \mathcal{F}\big(\mathbf{x}_{\le t},\, u_{\le t+1}\big)$, and the forecasting task is to learn a surrogate of $\mathcal{F}$ from data.

Forecasting model. A model predicts the next state from a finite window of past states and inputs,

$$\hat{\mathbf{x}}_{t+1} = f_\theta\big(\mathbf{x}_{t-N_x}, \ldots, \mathbf{x}_t;\; u_{t+1-N_i}, \ldots, u_{t+1}\big), \tag{5}$$

where N_x and N_i are the state and input memories. The input at the predicted time, u_{t+1}, is included: the excitation is prescribed, so it is known in advance.Most baselines follow an encode–forecast–decode structure: an encoder E maps each state to a latent $\mathbf{z}_t = E(\mathbf{x}_t) \in \mathbb{R}^r$ with $r \ll F \cdot N$, a forecaster g_θ advances the latent according to Eq. 5, and a decoder returns $\hat{\mathbf{x}}_{t+1} = D(\hat{\mathbf{z}}_{t+1})$. Models that act directly on the fields correspond to $E = D = \mathrm{id}$ (identity projection). Convolutional architectures act on the $H \times W$ images.

Test protocol: closed-loop rollout. Every test trajectory starts from the steady flame. The history window is filled with the initial (steady) snapshot, $\mathbf{x}_{-N_x} = \cdots = \mathbf{x}_0$, with $u_t = 0$ for $t \le 0$. The model is then applied recursively for the whole trajectory (≈ 2000 steps): each prediction $\hat{\mathbf{x}}_{t+1}$ is fed back as the newest state, while the true excitation u_{t+1} is supplied at every step. No reference snapshot is observed after $t = 0$, and all F fields are predicted on every cell. Because models are trained on sine sweeps only and tested on steps and fixed-frequency sines, including amplitudes outside the training range (Table 1), the task measures generalization to unseen excitations rather than interpolation between training conditions. A rollout that produces non-finite values is recorded as diverged.

Training and model selection. Training uses the two sine-sweep trajectories only. Each is cut in time into 20 blocks; 30% of each block, is held out for validation, so that both partitions cover the full frequency range of the sweep, and no window crosses a block edge. Neural forecasters minimize

a combination of one-step and unrolled errors,

$$\mathcal{L}(\theta) = (1 - w)\,\ell_1(\theta) + \frac{w}{h}\sum_{k=1}^{h}\ell_k(\theta), \tag{6}$$

where $\ell_k(\theta)$ is the error of the k-th recursive prediction of a training window, h is the training horizon, and $w \in [0, 1]$ is the rollout weight. The error is computed on the latents[1] or, when an autoencoder is fine-tuned jointly with the forecaster, on the decoded fields. The linear (ARX) and nonlinear (NARX) autoregressive models are fitted in closed form for the one-step increment $\mathbf{z}_{t+1} - \mathbf{z}_t$, i.e. $h = 1$. All hyperparameters, including N_x, N_i, h and the latent size r, are selected on the validation blocks using recursive rollouts of 150 steps; the test trajectories are never used for training or model selection. The selected configuration is then trained with three random seeds on the same split, with early stopping on the validation blocks, and each seed is tested; a fully deterministic model (POD with ARX or NARX) is trained once.

5 BASELINE MODELS

The baselines follow the encode–forecast–decode structure of Eq. 5. The chosen baselines were selected to cover diverse model families for both compression and forecasting, including linear and nonlinear representations, linear, quadratic and neural forecasters, and models with and without temporal memory. The descriptions of the compressors and forecasters are provided respectively in Sec.5.1 and 5.2.

For each complete model (pairs comnpressor-forecaster or neural operators) its hyperparameters are optimized through Bayesian Optimization with Optuna Akiba et al. (2019) (the search ranges are listed in Appendix B).

5.1 REDUCED-ORDER MODELS

POD. POD of the standardized snapshots, with all fields of a snapshot stacked into one vector. The POD basis is computed using a randomized SVD of the centered training snapshot matrix, and $\mathbf{z}_t$ holds the coefficients of the first r modes.

CAE. The encoder applies a stack of strided 2-D convolutions with SiLU activations and doubling channel widths to the field image, followed by a linear map to the r-dimensional latent; the decoder mirrors it with transposed convolutions. Like POD, the CAE works on deviations from the per-pixel training mean, and the decoder output is scaled by the per-pixel standard deviation computed on the training set, so that pixels that never vary during training are reconstructed exactly. The CAE is first trained on reconstruction and then, fine-tuned jointly with the forecaster on the loss of Eq. 6. A Kullback–Leibler term weighted by $\beta > 0$ turns the CAE into a variational CAE (CVAE) (Kingma & Welling, 2014).

Vision-transformer autoencoder (ViTAE). A hybrid vision-transformer (Dosovitskiy et al., 2021) introduces self-attention between the CAE feature maps and the latent space. The encoder and decoder follow symmetric designs, combining convolutional and transformer layers. In the encoder, convolutions first produce a feature map, to which a learnable positional embedding is added. Then, each resulting spatial location is treated as a transformer token. The decoder reverses this process, applying transformer layers followed by transposed convolutions to reconstruct the original input dimension. Similarly to the CVAE, a parameter $\beta > 0$ enables a Variational version of the hybrid ViTAE.

Identity. $E = D = \mathrm{id}$, used by the operator models below.

[1]The error in the latent space is computed between the predicted latent $\hat{\mathbf{z}}_{t+1}$ and the true one $\mathbf{z}_{t+1} = E(\mathbf{x}_{t+1})$. Note that, when computing the loss in the latent space, the encoder's parameters are frozen, thus the loss does not depends on them.

5.2 FORECASTING MODELS

All latent forecasters read the window of Eq. 5, i.e. the latest $N_x{+}1$ latents and $N_i{+}1$ input values, and predict the increment $\hat{\mathbf{z}}_{t+1} = \mathbf{z}_t + g_\theta(\cdot)$, so that $g_\theta = 0$ corresponds to persistence.

Linear and control-affine models. ARX fits the increment as a linear function of the window,

$$\hat{\mathbf{z}}_{t+1} - \mathbf{z}_t = \mathbf{A}\,[\mathbf{z}_{t-N_x}; \ldots; \mathbf{z}_t] + \mathbf{B}\,[u_{t+1-N_i}; \ldots; u_{t+1}] + \mathbf{c}, \tag{7}$$

by ridge regression in closed form. NARX adds to Eq. 7 bilinear state–input terms $\mathbf{z}_{t-k} \cdot u_{t+1-l}$, and the parameters are again computed through ridge regression. Both are one-step fits ($h = 1$) and, for given latents, deterministic.

Sequence networks. GRU, LSTM, CNN and Transformer take as input a tensor $[\{\mathbf{z}_s\}_{s=t-b+1}^{t}, \{u_{s+1}\}_{s=t-b+1}^{t}] \in \mathbb{R}^{b\times(r+1)}$, with $b = \max\{N_x, N_i\}{+}1$ [2], and output the increment $\hat{\mathbf{z}}_{t+1} - \mathbf{z}_t$. The GRU and LSTM stack one or two recurrent layers and map the last hidden state to the latent dimension with a linear layer. The CNN applies one or two 1-D convolutions along time and a linear layer on the flattened output. The Transformer embeds each row as a token with a learned positional embedding, applies one or two transformer layers, and maps the last output token the latent dimension though a linear projection. All are trained with the rollout loss of Eq. 6; the architecture, optimizer, rollout weight, horizon h and input lags (N_x, N_i) are tuned as explained in Sec. 5.3.

Neural operator. Neural operators are designed to learn mappings between function spaces and can, in principle, be evaluated across different spatial discretizations (Kovachki et al., 2023). This advantage is not exploited here, since the same mesh resolution is used for both training and testing. Nevertheless, given their widespread use for PDE-based surrogate modeling, we include a two-dimensional FNO (FNO-2D) (Li et al., 2021) as a full-field baseline operating directly on the spatial fields without a separate compressor.

DMD. We test two variants of DMD with control (Proctor et al., 2016). On POD, DMDc is ARX with $N_x = N_i = 0$: $\hat{\mathbf{z}}_{t+1} - \mathbf{z}_t = \mathbf{A}\mathbf{z}_t + \mathbf{B}u_{t+1} + \mathbf{c}$, with the affine term $\mathbf{c}$ kept because the latent is centered on the training mean, not on a steady state. *Exact DMDc* follows the original algorithm on the full field (identity compressor). Both are one-step, deterministic fits.

OpInf. OpInf (Peherstorfer & Willcox, 2016) extends the same Markovian, one-step fit of DMD with a quadratic term: $\hat{\mathbf{z}}_{t+1} - \mathbf{z}_t = \mathbf{A}\mathbf{z}_t + \mathbf{H}(\mathbf{z}_t \otimes \mathbf{z}_t) + \mathbf{B}u_{t+1} + \mathbf{c}$ where $\mathbf{A}$, $\mathbf{H}$, $\mathbf{B}$, and $\mathbf{c}$ are fitted jointly by regularized least squares, with a separate ridge weight on $\mathbf{H}$, making it a nonlinear counterpart to DMDc.

5.3 HYPERPARAMETER OPTIMIZATION

For a fair comparison, all learnable components are tuned under the same search-trial budget, through Optuna Bayesian optimization. An overview on the hyperparameter search ranges is provided in Tab. 3. For each compressor–forecaster pair, hyperparameters are optimized in three consecutive stages, with 25 trials per stage and validation performance as the objective. First, four compressor configurations are optimized independently, one for each latent dimension $r \in \{8, 16, 32, 64\}$. POD is excluded from this optimization: it is fitted once at the maximum rank, and lower-dimensional representations are obtained by truncation. Second, the four compressors are frozen while the latent dimension r, the lags N_x and N_i, and the forecaster hyperparameters are optimized jointly. Third, when both the compressor and forecaster are neural networks, the hyperparameters governing end-to-end fine-tuning are optimized.

6 EVALUATION METRICS

Forecasting performance is evaluated by comparing the predicted states with the corresponding CFD reference solutions over the closed-loop rollout. All metrics are computed in physical units, after

[2] The shorter history has $|N_x - N_i|$ zero-padded rows.

inverting the training normalization, and evaluated only on the computational cells and on the predicted snapshots.

Field error. For each physical feature f, the normalized root-mean-square error is

$$\mathrm{nRMSE}_f = \frac{1}{\sigma_f}\sqrt{\frac{1}{TN}\sum_{t=1}^{T}\sum_{i=1}^{N}\left(\hat{x}_{f,i,t} - x_{f,i,t}\right)^2}, \tag{8}$$

where N is the number of cells, T is the number of predicted snapshots, and σ_f is the standard deviation of the reference field over all cells and snapshots. The overall nRMSE is obtained by averaging nRMSE_f over the retained physical features and then over the test trajectories. Features with negligible reference mean and standard deviation are excluded from the average to avoid ill-conditioned normalization. Since the error is normalized by σ_f, a prediction equal to the reference mean has $\mathrm{nRMSE}_f = 1$.

Heat-release error. The relative trajectory heat-release error between the predicted $\hat{\dot{Q}}$ and the true $\dot{Q}$ is computed as

$$E_{\dot{Q}} = \frac{\left\|\hat{\dot{Q}} - \dot{Q}\right\|_2}{\left\|\dot{Q}\right\|_2}, \qquad \text{with} \quad Q(t) = \sum_{i=1}^{N} \dot{q}_i(t) V_i, \tag{9}$$

where $\dot{q}_i$ and V_i are the heat-release rate and physical volume of cell i, respectively.

Flame response. For the fixed-frequency sinusoidal cases, the dynamical response of the flame is characterized through the transfer function between the prescribed inlet forcing $u(t) = \phi(t) - 1$ and the normalized heat-release response $(\dot{Q}((t) - \dot{Q}_0)/\dot{Q}_0$. The first harmonic is obtained by least-squares fitting after the initial transient, and the gain and phase errors are computed from the corresponding complex response at the forcing frequency. The relative gain error measures the mismatch in response amplitude, while the phase error quantifies the difference in temporal response and is reported in degrees.

Computational cost and stability. Inference cost is reported as the wall-clock time required per rollout step, including the latent-state transition and decoding.

7 Results

Table 2 reports, for each compressor, the three pairs with the lowest nRMSE. All pairs and the per-input errors are in Appendix C.

Closed-form latent forecasters are consistently more accurate than neural ones: ARX outperforms all neural forecasters on every latent space, reaching its lowest errors on CAE and CVAE (0.166 and 0.169), while POD-based closed-form models best reproduce heat release ($\varepsilon_{\dot{Q}} = 0.026$–$0.029$ versus 0.070 for the best neural forecaster). Neural forecasters are comparatively fragile: several underperform the constant forecast (nRMSE 0.326) and others show strong seed sensitivity or diverge. Field accuracy also does not guarantee accurate heat-release dynamics: FNO achieves the lowest nRMSE (0.107) but a larger heat-release error than the constant forecast ($\varepsilon_{\dot{Q}} = 0.611$ versus 0.206), with order-of-magnitude gain errors at 40 Hz. Field error alone is therefore insufficient to select an accurate surrogate.

8 Discussion and limitations

The observed advantage of linear models may have several causes. First, with only two training trajectories, a model with a single regularization parameter and a short input window is less prone to overfitting than a neural network. Second, the laminar configuration may result in less complex flow dynamics than turbulent systems, while the absence of experimental noise may further favor lower-complexity models. Finally, neural forecasters are trained over finite horizons ($h \leq 75$) and

Table 2: Test-set forecasting error, as mean $\pm$ standard deviation over three seeds: for each compressor, the three pairs with the lowest nRMSE, and all full-field models (identity compressor). nRMSE is averaged over the test trajectories, and ε_Q is the relative L_2 error of the volume-integrated heat-release rate (Section 6). POD with a closed-form forecaster, exact DMDc and the constant forecast are deterministic and run with a single seed, so they have no standard deviation. Best value per column in bold, second best underlined.

Compressor	Forecaster	nRMSE	ε_Q	Tuned configuration
POD	ARX	0.202	**0.026**	rank 40, $N_x = 5$, $N_i = 9$
POD	OpInf	0.213	<u>0.029</u>	rank 62
POD	Transformer	0.225 ± 0.015	0.070	rank 32, $N_x = 9$, $N_i = 17$, $h = 50$
CAE	ARX	<u>0.166 ± 0.006</u>	0.042	rank 16, $N_x = 11$, $N_i = 15$
CAE	Transformer	0.384 ± 0.012	0.156	rank 64, $N_x = 14$, $N_i = 10$, $h = 40$
CAE	GRU	0.495 ± 0.263	0.168	rank 32, $N_x = 16$, $N_i = 15$, $h = 30$
CVAE	ARX	0.169 ± 0.003	0.038	rank 16, $N_x = 8$, $N_i = 18$
CVAE	NARX	0.339 ± 0.148	0.047	rank 32, $N_x = 8$, $N_i = 18$
CVAE	Transformer	0.434 ± 0.055	0.201	rank 64, $N_x = 9$, $N_i = 13$, $h = 30$
ViTAE	ARX	0.250 ± 0.015	0.154	rank 8, $N_x = 16$, $N_i = 20$
ViTAE	NARX	0.257 ± 0.001	0.143	rank 16, $N_x = 11$, $N_i = 7$
ViTAE	GRU	0.317 ± 0.007	0.251	rank 8, $N_x = 16$, $N_i = 17$, $h = 30$
Identity	FNO	$\mathbf{0.107 \pm 0.003}$	0.611	$N_x = 2$, $N_i = 14$, $h = 10$
Identity	Exact DMDc	0.306	0.627	$p = 63$, $r = 56$
Identity	Constant	0.326	0.206	–

may accumulate errors during the $\approx$ 2000-step test rollout. The restarted protocol of Section 4, comparing one-step and closed-loop errors, can help distinguish this effect.

FireMark is limited to a single laminar axisymmetric configuration and one forced input, while turbulent and multi-input cases remain outside its scope. The benchmark also relies on only two training trajectories and a limited range of forcing conditions, which may disadvantage data-intensive models. The fixed hyperparameter-search budget of 25 trials per stage may favor models with fewer tunable parameters. Finally, because $\varepsilon_{\dot{Q}}$ is normalized by the total heat release, including its steady component, it can mask differences in the fluctuating response; gain and phase metrics therefore provide complementary measures of flame dynamics.

9 Conclusions

FireMark provides a standardized benchmark for data-driven forecasters of combustion systems, combining high-fidelity, spatially, and temporally resolved flame fields. Its closed-loop evaluation protocol is designed to assess not only the accuracy of long-horizon forecasting but also the generalization to forcing conditions and excitation types not observed during training. Using this framework, we find that a linear ARX model on a reduced POD basis provides a strong and robust baseline. Several neural forecasters fail to outperform a constant prediction, and that field-wise accuracy alone may not reflect the quality of the predicted integrated heat-release dynamics. These results demonstrate that FireMark poses a non-trivial forecasting challenge and provides a reproducible reference point for assessing future data-driven models for reacting flows.

AI use statement

In this work, we used generative AI tools for AI-assisted coding and to improve readability. We have not used generative AI tools to propose or refine hypotheses. We have reviewed all AI-assisted work. Every reported number is regenerated by scripts in the repository and traced to a result file; AI-assisted code was tested by at least one author. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

Anonymous Anonymous. Cfd dataset of an axisymmetric non-premixed methane coflow flame under fuel-velocity excitations, 2026. URL https://doi.org/10.5281/zenodo.22771298.

Robert S Barlow and Jonathan H Frank. Effects of turbulence on species mass fractions in methane/air jet flames. In *Symposium (international) on combustion*, volume 27, pp. 1087–1095. Elsevier, 1998.

Wai Tong Chung, Ki Sung Jung, Jacqueline H Chen, and Matthias Ihme. Blastnet: A call for community-involved big data in combustion machine learning. *Applications in Energy and Combustion Science*, 12:100087, 2022.

Wai Tong Chung, Bassem Akoush, Pushan Sharma, Alex Tamkin, Ki Sung Jung, Jacqueline Chen, Jack Guo, Davy Brouzet, Mohsen Talei, Bruno Savard, et al. Turbulence in focus: Benchmarking scaling behavior of 3d volumetric super-resolution with blastnet 2.0 data. *Advances in Neural Information Processing Systems*, 36:77430–77484, 2023.

A. Cuoci, A. Frassoldati, T. Faravelli, and E. Ranzi. Formation of soot and nitrogen oxides in unsteady counterflow diffusion flames. *Combustion and Flame*, 156:2010–2022, 10 2009. ISSN 0010-2180. doi: 10.1016/J.COMBUSTFLAME.2009.06.023.

A. Cuoci, A. Frassoldati, T. Faravelli, and E. Ranzi. Opensmoke++: An object-oriented framework for the numerical modeling of reactive systems with detailed kinetic mechanisms. *Computer Physics Communications*, 192:237–264, 7 2015. ISSN 0010-4655. doi: 10.1016/J.CPC.2015.02.014.

Alberto Cuoci, Alessio Frassoldati, Tiziano Faravelli, and Eliseo Ranzi. Numerical modeling of laminar flames with detailed kinetics based on the operator-splitting method. *Energy & Fuels*, 27:7730–7753, 2013. doi: 10.1021/ef4016334. URL https://doi.org/10.1021/ef4016334.

Giuseppe D'Alessio, Alessandro Parente, Alessandro Stagni, and Alberto Cuoci. Adaptive chemistry via pre-partitioning of composition space and mechanism reduction. *Combustion and Flame*, 211:68–82, 2020. ISSN 0010-2180. doi: https://doi.org/10.1016/j.combustflame.2019.09.010. URL https://www.sciencedirect.com/science/article/pii/S0010218019304249.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

AmirPouya Hemmasian and Amir Barati Farimani. Reduced-order modeling of fluid flows with transformers. *Physics of Fluids*, 35(5), 2023.

Matthias Ihme, Wai Tong Chung, and Aashwin Ananda Mishra. Combustion machine learning: Principles, progress and prospects. *Progress in Energy and Combustion Science*, 91:101010, 2022.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations, ICLR*, 2014.

Felix Koehler, Simon Niedermayr, Rüdiger Westermann, and Nils Thuerey. Apebench: A benchmark for autoregressive neural emulators of pdes. *Advances in Neural Information Processing Systems*, 37:120252–120310, 2024.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 2023.

Yi Li, Eric Perlman, Minping Wan, Yunke Yang, Charles Meneveau, Randal Burns, Shiyi Chen, Alexander Szalay, and Gregory Eyink. A public turbulence database cluster and applications to study lagrangian evolution of velocity increments in turbulence. *Journal of Turbulence*, (9):N31, 2008.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

Yining Luo, Yingfa Chen, and Zhen Zhang. Cfdbench: A large-scale benchmark for machine learning methods in fluid dynamics. *arXiv preprint arXiv:2310.05963*, 2023.

Shane A McQuarrie, Cheng Huang, and Karen E Willcox. Data-driven reduced-order models via regularised operator inference for a single-injector combustion process. *Journal of the Royal Society of New Zealand*, 51(2):194–211, 2021.

Rahima K. Mohammed, Michael A. Tanoff, Mitchell D. Smooke, Andrew M. Schaffer, and Marshall B. Long. Computational and experimental study of a forced, timevarying, axisymmetric, laminar diffusion flame. *Symposium (International) on Combustion*, 27:693–702, 1 1998. ISSN 0082-0784. doi: 10.1016/S0082-0784(98)80462-1.

Arvind T Mohan and Datta V Gaitonde. A deep learning based approach to reduced order modeling for turbulent flow control using lstm neural networks. *arXiv preprint arXiv:1804.09269*, 2018.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina J Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Stuart B Dalziel, Drummond B Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

Benjamin Peherstorfer and Karen Willcox. Data-driven operator inference for nonintrusive projection-based model reduction. *Computer Methods in Applied Mechanics and Engineering*, 306:196–215, 2016.

Eric Perlman, Randal Burns, Yi Li, and Charles Meneveau. Data exploration of turbulence simulations using a database cluster. In *Proceedings of the 2007 ACM/IEEE Conference on Supercomputing*, pp. 1–11, 2007.

Joshua L Proctor, Steven L Brunton, and J Nathan Kutz. Dynamic mode decomposition with control. *SIAM Journal on Applied Dynamical Systems*, 15(1):142–161, 2016.

E. Ranzi, A. Frassoldati, R. Grana, A. Cuoci, T. Faravelli, A. P. Kelley, and C. K. Law. Hierarchical and comparative kinetic modeling of laminar flame speeds of hydrocarbon and oxygenated fuels. *Progress in Energy and Combustion Science*, 38:468–501, 8 2012. ISSN 0360-1285. doi: 10.1016/J.PECS.2012.03.004.

Renee Swischuk, Boris Kramer, Cheng Huang, and Karen Willcox. Learning physics-based reduced-order models for a single-injector combustion process. *AIAA Journal*, 58(6):2658–2672, 2020.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in neural information processing systems*, 35:1596–1611, 2022.

Aaron Towne, Scott TM Dawson, Guillaume A Brès, Adrián Lozano-Durán, Theresa Saxton-Fox, Aadhy Parthasarathy, Anya R Jones, Hulya Biler, Chi-An Yeh, Het D Patel, et al. A database for reduced-complexity modeling of fluid flows. *AIAA journal*, 61(7):2867–2892, 2023.

Xiaopeng Wang, Chen'an Zhang, Wen Liu, Famin Wang, and Zhengyin Ye. Unsteady aerodynamic modeling based on pod-arx. *International Journal of Aerospace Engineering*, 2018(1):5902794, 2018.

A Input Excitations

Figure 3 summarizes the prescribed inlet excitations and the train–test split. Training uses two sine sweeps spanning 1–80–1 Hz at amplitudes $A = 0.2$ and $A = 0.4$. Testing uses fixed-frequency sinusoids at 10 and 40 Hz and step excitations, each at $A = 0.3$ and $A = 0.5$. The test set therefore probes fixed frequencies within the frequency range covered during training, amplitudes within and beyond the training range, and an unseen excitation type.

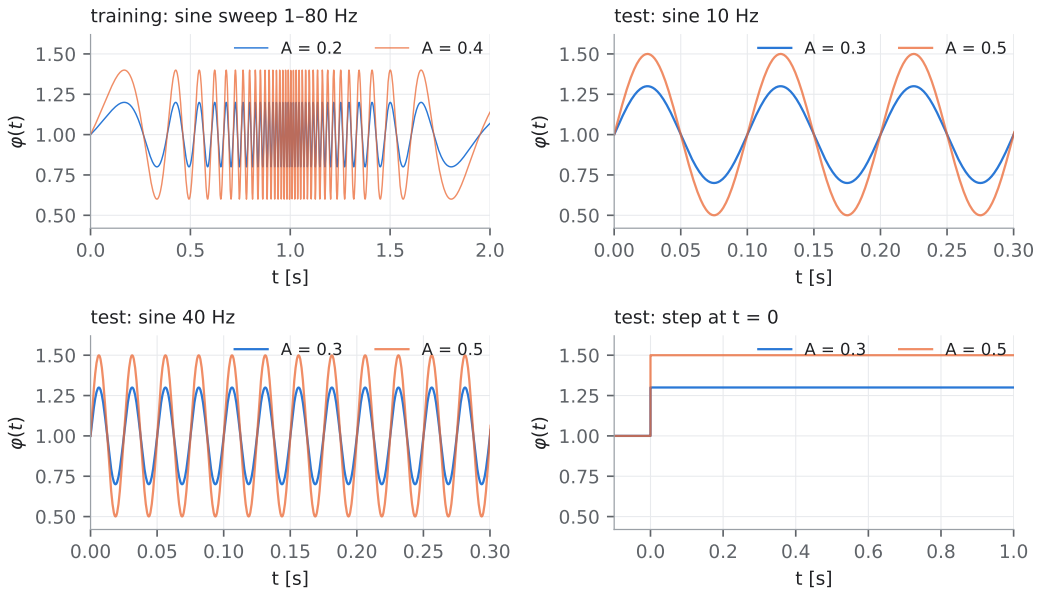

Figure 3: Excitation signals $\phi(t)$ used for training and testing. Top left: training sine sweeps with $A = 0.2$ and 0.4, spanning 1–80–1 Hz over 2 s. Top right and bottom left: test sinusoids at 10 and 40 Hz, respectively (first 0.3 s shown). Bottom right: test step excitations applied at $t = 0$ from the unforced state $\phi = 1$. For the test cases, $A = 0.3$ lies within the training amplitude range, whereas $A = 0.5$ lies beyond it.

B Hyperparameter Search

Hyperparameters are optimized using the tree-structured Parzen estimator (TPE) implemented in Optuna. Depending on the model, the optimization consists of up to three consecutive stages. First, learned compressors are optimized for reconstruction performance, separately for each candidate latent dimension. Second, the compressor is frozen while the latent dimension, input and state histories, and forecaster hyperparameters are selected using the validation error of 150-step recursive forecasts. Finally, when both the compressor and forecaster are neural networks, the hyperparameters for end-to-end fine-tuning are optimized. Each optimization stage uses 25 trials, and trials producing divergent predictions are treated as unsuccessful. Table 3 reports the complete search spaces.

C Additional Results

The following results complement the aggregate test-set performance reported in the main text and provide a more detailed comparison of the tuned baselines.

Table 3: Hyperparameter search space explored with the TPE sampler. Intervals are denoted by $[a, b]$; integer-valued ranges are used for ranks, window lengths, levels, and layers; parameters marked "log" are sampled on a logarithmic scale; and $\{\cdot\}$ denotes categorical choices. If a model is not listed for a given hyperparameter, that hyperparameter is not tuned for that model.

Model	Hyperparameter	Range
POD	rank r	$[8, 128]$ (log)
CAE, CVAE	rank r	$\{8, 16, 32, 64\}$
	learning rate	$[10^{-4}, 3 \cdot 10^{-3}]$ (log)
	batch size	$\{8, 16, 32\}$
	levels	$[1, 4]$
	base channels	$\{8, 16\}$
	kernel size	$\{2, 3\}$
ViTAE	as CAE, but levels	$[3, 4]$
	token width	$\{32, 64\}$
	heads	$\{2, 4\}$
	attention layers	$[1, 2]$
ARX, NARX, neural, FNO	N_x, N_i	$[0, 20]$
ARX, DMDc	α	$[10^{-8}, 1]$ (log)
NARX	α	$[10^{-8}, 10^{3}]$ (log)
	bilinear α	$[10^{-8}, 10^{6}]$ (log)
OpInf	α	$[10^{-8}, 10^{4}]$ (log)
	quadratic α	$[10^{3}, 10^{12}]$ (log)
Neural (latent)	horizon h	$\{30, 40, 50, 60, 75\}$
	learning rate	$[10^{-5}, 3 \cdot 10^{-3}]$ (log)
	optimizer	{Adam, AdamW}
	weight decay	$[10^{-6}, 10^{-2}]$ (log)
	dropout	$[0, 0.3]$
	rollout weight	$[0, 1]$
	joint learning rate	$[10^{-6}, 10^{-3}]$ (log)
	joint epochs	$\{5, 10, 20\}$
GRU, LSTM	hidden sizes	$\{[16], [32], [64], [32, 32]\}$
	layer normalization	{none, layer}
CNN	channels	$\{[16], [32], [16, 16], [32, 32]\}$
	kernel size	$\{3, 5\}$
	activation	{ReLU, GELU, SiLU}
	layer normalization	{none, layer, batch}
Transformer	token width	$\{16, 32, 64\}$
	layers	$[1, 2]$
	heads	$\{2, 4\}$
	activation	{ReLU, GELU}
Exact DMDc	ranks p, r	$[8, 256]$, $[8, 128]$ (log), $r \leq p$
FNO	Fourier layers	$[2, 4]$
	width	$\{20, 32, 64\}$
	mode pair (m_z, m_x)	$\{(8, 8), (12, 12), (16, 16)\}$
	horizon h	$\{1, 5, 10\}$
	optimizer, weight decay	as neural (latent)

Table 4: Test nRMSE per forcing input. Mean over three seeds (single seed for the deterministic pairs). Best value per column in bold, second best underlined; div.: the rollout grows without bound.

Compressor	Forecaster	Sine 10 Hz		Sine 40 Hz		Step	
		A=0.3	A=0.5	A=0.3	A=0.5	A=0.3	A=0.5
POD	ARX	0.166	0.310	0.145	0.199	0.123	0.272
POD	NARX	0.177	0.379	0.134	0.208	0.149	0.377
POD	DMDc	0.188	0.359	0.139	0.210	0.259	0.432
POD	OpInf	0.164	0.316	0.131	0.201	0.179	0.287
POD	Transformer	0.226	0.329	0.159	0.205	0.141	0.289
POD	LSTM	0.198	0.366	0.188	0.291	0.296	0.494
POD	CNN	0.279	0.402	0.238	0.268	0.240	0.437
POD	GRU	0.290	0.457	0.212	0.325	1.902	4.409
CAE	ARX	<u>0.127</u>	<u>0.255</u>	0.118	<u>0.179</u>	0.091	<u>0.225</u>
CAE	NARX	0.160	0.316	<u>0.097</u>	0.224	0.550	9.504
CAE	Transformer	0.388	0.641	0.276	0.327	0.310	0.362
CAE	GRU	0.294	0.487	0.249	0.281	0.713	0.944
CAE	LSTM	0.687	0.688	0.498	0.913	0.561	0.846
CAE	CNN	div.	div.	div.	div.	div.	div.
CVAE	ARX	0.132	0.263	0.119	0.183	<u>0.087</u>	0.227
CVAE	NARX	0.129	0.299	0.149	1.067	0.089	0.301
CVAE	Transformer	0.466	0.513	0.410	0.404	0.357	0.455
CVAE	GRU	0.988	0.960	0.982	0.931	0.858	0.946
CVAE	LSTM	2.054	3.024	1.374	1.288	2.240	1.924
ViTAE	ARX	0.199	0.333	0.178	0.259	0.188	0.341
ViTAE	NARX	0.202	0.335	0.190	0.286	0.189	0.341
ViTAE	GRU	0.317	0.463	0.218	0.286	0.245	0.373
ViTAE	LSTM	0.282	0.476	0.211	0.289	0.297	0.414
ViTAE	Transformer	0.338	0.477	0.241	0.297	0.264	0.382
Identity	FNO	**0.080**	**0.221**	**0.069**	**0.106**	**0.047**	**0.118**
Identity	Exact DMDc	0.293	0.437	0.197	0.274	0.249	0.386
Identity	Constant	0.316	0.468	0.195	0.291	0.262	0.424

Table 5: Relative L_2 error ε_Q of the volume-integrated heat-release rate per forcing input. Mean over three seeds (single seed for the deterministic pairs). Best value per column in bold, second best underlined; div.: the rollout grows without bound.

Compressor	Forecaster	Sine 10 Hz		Sine 40 Hz		Step	
		A=0.3	A=0.5	A=0.3	A=0.5	A=0.3	A=0.5
POD	ARX	0.032	0.078	<u>0.011</u>	<u>0.020</u>	<u>0.007</u>	<u>0.010</u>
POD	NARX	<u>0.030</u>	<u>0.070</u>	**0.005**	0.026	0.008	0.011
POD	DMDc	0.035	0.078	0.012	**0.014**	**0.005**	**0.009**
POD	OpInf	0.036	0.077	0.014	0.028	<u>0.007</u>	0.013
POD	Transformer	0.086	0.129	0.047	0.058	0.039	0.062
POD	LSTM	0.087	0.188	0.035	0.052	0.171	0.191
POD	CNN	0.078	0.200	0.063	0.069	0.048	0.081
POD	GRU	0.133	0.262	0.048	0.107	0.770	1.231
CAE	ARX	0.031	0.072	0.034	0.054	0.014	0.047
CAE	NARX	0.032	0.090	0.023	0.053	0.061	0.922
CAE	Transformer	0.191	0.361	0.079	0.094	0.110	0.099
CAE	GRU	0.110	0.288	0.059	0.067	0.147	0.336
CAE	LSTM	0.481	0.439	0.164	0.275	0.241	0.212
CAE	CNN	div.	div.	div.	div.	div.	div.
CVAE	ARX	0.039	0.076	0.019	0.029	0.019	0.044
CVAE	NARX	**0.027**	**0.065**	0.025	0.113	0.019	0.035
CVAE	Transformer	0.251	0.298	0.167	0.136	0.139	0.213
CVAE	GRU	0.398	0.484	0.285	0.279	0.238	0.398
CVAE	LSTM	0.464	0.921	0.430	0.244	0.544	0.592
ViTAE	ARX	0.131	0.241	0.037	0.050	0.160	0.304
ViTAE	NARX	0.063	0.167	0.090	0.165	0.168	0.206
ViTAE	GRU	0.283	0.376	0.176	0.185	0.188	0.299
ViTAE	LSTM	0.231	0.479	0.085	0.118	0.312	0.463
ViTAE	Transformer	0.294	0.392	0.160	0.164	0.209	0.318
Identity	FNO	0.604	1.113	0.512	0.680	0.284	0.475
Identity	Exact DMDc	0.600	0.637	0.573	0.574	0.670	0.710
Identity	Constant	0.224	0.349	0.034	0.047	0.238	0.343

Table 6: Relative gain error and absolute phase error (degrees) of the heat-release response at the forcing frequency, for the sinusoidal inputs, from the harmonic of the normalized heat-release rate over $t \cdot \Delta t \geq 0.5$ s (Section 6). Mean over three seeds (single seed for the deterministic pairs). Best value per column in bold, second best underlined; div.: the rollout grows without bound.

Compressor	Forecaster	Sine 10 Hz				Sine 40 Hz			
		A=0.3		A=0.5		A=0.3		A=0.5	
		gain	phase	gain	phase	gain	phase	gain	phase
POD	ARX	**0.006**	<u>1.5</u>	0.037	1.7	<u>0.064</u>	15.6	0.331	10.0
POD	NARX	<u>0.011</u>	2.4	0.047	1.3	**0.026**	**0.5**	0.274	24.3
POD	DMDc	0.065	2.1	<u>0.024</u>	<u>1.1</u>	0.157	17.4	**0.055**	<u>8.2</u>
POD	OpInf	0.036	3.9	**0.006**	**0.8**	0.353	13.7	<u>0.189</u>	39.3
POD	Transformer	0.119	13.4	0.116	4.3	0.454	27.6	0.485	30.9
POD	LSTM	0.261	8.8	0.414	6.2	0.821	28.3	0.886	49.2
POD	CNN	0.169	7.5	0.415	28.8	0.579	59.8	0.543	38.0
POD	GRU	0.310	25.0	0.528	26.9	0.310	69.3	0.292	103.1
CAE	ARX	0.084	<u>1.5</u>	0.160	4.7	0.521	28.7	0.475	**7.2**
CAE	NARX	0.097	**1.4**	0.194	5.1	0.251	14.7	0.320	21.4
CAE	Transformer	0.363	38.3	0.503	79.4	0.437	43.5	0.528	68.0
CAE	GRU	0.259	19.9	0.505	61.7	0.701	27.4	0.685	48.5
CAE	LSTM	0.817	61.5	0.844	53.0	0.598	114.7	0.789	75.5
CAE	CNN	div.	31.2	div.	29.5	div.	84.6	div.	63.6
CVAE	ARX	0.055	2.5	0.113	1.6	0.093	21.0	0.199	18.0
CVAE	NARX	0.051	1.8	0.119	2.9	0.417	<u>12.0</u>	0.340	40.1
CVAE	Transformer	0.902	63.3	0.759	39.7	0.911	77.4	0.801	127.5
CVAE	GRU	0.925	118.8	0.941	119.9	0.809	133.4	0.802	118.0
CVAE	LSTM	0.732	91.4	0.742	98.4	0.556	58.3	0.668	46.2
ViTAE	ARX	0.365	20.5	0.435	20.4	0.239	38.1	0.421	29.4
ViTAE	NARX	0.182	5.1	0.461	6.3	0.534	107.5	0.608	157.4
ViTAE	GRU	0.937	106.7	0.904	126.2	0.639	100.2	0.572	130.7
ViTAE	LSTM	0.603	55.0	0.766	75.1	0.698	55.6	0.560	56.9
ViTAE	Transformer	0.993	100.0	0.998	99.3	0.982	80.8	0.990	103.3
Identity	FNO	0.845	44.2	1.502	37.9	9.352	59.6	11.281	76.2
Identity	Exact DMDc	0.980	54.3	0.988	35.6	0.976	43.1	0.966	40.9
Identity	Constant	1.000	–	1.000	–	1.000	–	1.000	–

Table 7: Test-set forecasting error of the tuned compressor–forecaster pairs, as mean $\pm$ standard deviation over three seeds. nRMSE is averaged over the test trajectories, and ε_Q is the relative L_2 error of the volume-integrated heat-release rate (Section 6). POD with a closed-form forecaster (ARX, NARX, DMDc, OpInf), exact DMDc and the constant forecast are deterministic and run with a single seed, so they have no standard deviation. Best value per column in bold, second best underlined. No rollout became non-finite, but CAE with CNN (nRMSE $3.1 \times 10^4 \pm 4.4 \times 10^4$, $\varepsilon_Q = 1.3 \times 10^4$), and one seed of CAE with NARX, grow without bound.

Compressor	Forecaster	nRMSE	ε_Q	Tuned configuration
POD	ARX	0.202	<u>0.026</u>	rank 40, $N_x = 5$, $N_i = 9$
POD	NARX	0.237	**0.025**	rank 32, $N_x = 10$, $N_i = 14$
POD	DMDc	0.265	**0.025**	rank 59
POD	OpInf	0.213	0.029	rank 62
POD	Transformer	0.225 ± 0.015	0.070	rank 32, $N_x = 9$, $N_i = 17$, $h = 50$
POD	LSTM	0.306 ± 0.056	0.121	rank 17, $N_x = 16$, $N_i = 2$, $h = 30$
POD	CNN	0.311 ± 0.023	0.090	rank 25, $N_x = 9$, $N_i = 15$, $h = 40$
POD	GRU	1.266 ± 0.176	0.425	rank 56, $N_x = 8$, $N_i = 12$, $h = 30$
CAE	ARX	<u>0.166 ± 0.006</u>	0.042	rank 16, $N_x = 11$, $N_i = 15$
CAE	NARX	1.808 ± 2.295	0.197	rank 32, $N_x = 15$, $N_i = 13$
CAE	Transformer	0.384 ± 0.012	0.156	rank 64, $N_x = 14$, $N_i = 10$, $h = 40$
CAE	GRU	0.495 ± 0.263	0.168	rank 32, $N_x = 16$, $N_i = 15$, $h = 30$
CAE	LSTM	0.699 ± 0.151	0.302	rank 32, $N_x = 13$, $N_i = 15$, $h = 75$
CAE	CNN	diverged	diverged	rank 32, $N_x = 16$, $N_i = 18$, $h = 75$
CVAE	ARX	0.169 ± 0.003	0.038	rank 16, $N_x = 8$, $N_i = 18$
CVAE	NARX	0.339 ± 0.148	0.047	rank 32, $N_x = 8$, $N_i = 18$
CVAE	Transformer	0.434 ± 0.055	0.201	rank 64, $N_x = 9$, $N_i = 13$, $h = 30$
CVAE	GRU	0.944 ± 0.063	0.347	rank 64, $N_x = 12$, $N_i = 14$, $h = 50$
CVAE	LSTM	1.984 ± 0.876	0.532	rank 64, $N_x = 16$, $N_i = 13$, $h = 30$
ViTAE	ARX	0.250 ± 0.015	0.154	rank 8, $N_x = 16$, $N_i = 20$
ViTAE	NARX	0.257 ± 0.001	0.143	rank 16, $N_x = 11$, $N_i = 7$
ViTAE	GRU	0.317 ± 0.007	0.251	rank 8, $N_x = 16$, $N_i = 17$, $h = 30$
ViTAE	LSTM	0.328 ± 0.038	0.282	rank 8, $N_x = 19$, $N_i = 12$, $h = 40$
ViTAE	Transformer	0.333 ± 0.037	0.256	rank 8, $N_x = 20$, $N_i = 13$, $h = 40$
Identity	FNO	**0.107 ± 0.003**	0.611	$N_x = 2$, $N_i = 14$, $h = 10$
Identity	Exact DMDc	0.306	0.627	$p = 63$, $r = 56$
Identity	Constant	0.326	0.206	–