Machine Learning: Science and Technology
PAPER

Data Driven Surrogates for Uncertainty Quantification incorporating Higher Moments in Flame Spread Modeling

Hamza Jamil[1] and Fabian Brännström[1,*]

[1]Chair of Fire Dynamics, University of Wuppertal, Gaußstraße 20, Wuppertal, 42119, Germany
*Fabian Brännström

E-mail: braennstroem@uni-wuppertal.de

Keywords: Flame spread; Pyrolysis; Surrogates; Uncertainty quantification; Reduced order model; Proper orthogonal decomposition; Radial basis functions; Gaussian process regression; Autoencoder; Fourier neural operator;

Abstract

In this work we implement and compare data driven machine learning surrogates for uncertainty quantification in a flame spread model that captures an experimental setup. We start with a moment independent global sensitivity analysis to choose the most sensitive kinetic and thermophysical parameters affecting the model output. Stochastic uncertainty is then studied in the output over space and time for five most sensitive parameters by analyzing four statistical moments. Different surrogate approaches are compared, namely proper orthogonal decomposition with interpolation (PODI), Gaussian process regression (GPR), Fourier neural operator (FNO), fully connected autoencoder (MLP-AE) and convolutional autoencoder (CAE). Performance of surrogates for both 1D temporal and 2D spatial outputs are analyzed for accuracy and robustness using small to moderate training data sizes. For 1D, all surrogates with the exception of GPR give accurate moment predictions. For 2D, PODI outperforms in capturing expectation and standard deviation but is comparatively weak in approximating higher moments where the CAE shows more promise. FNO is effective for samples hundred and below while GPR exhibits largest errors for the studied case. Considering the empirical importance of the statistical moments, PODI presents the most reliable and robust approach for uncertainty quantification in skewed flame spread profiles.

1 Introduction

Building on De Ris's [1] analytical flame spread theory, Pello [2, 3] concluded that no single explicit expression could capture flame spread rate across varied materials and environmental conditions, motivating numerical [4] and eventually CFD-based [5–8] flame spread modeling. We delve further to explore the current transitional era of artificial intelligence where machine learning based models can be appended or even substituted with computationally expensive flame spread CFD models in order to gain considerable speedup, making repeated model evaluations for uncertainty quantification feasible. Flame spread is in itself a multi-physics phenomenon involving gas phase combustion, soot formation, oxidation, radiation, turbulence, solid phase heat transfer and pyrolysis [9]. Predicting flame spread accurately requires high-fidelity simulations as capturing tightly coupled, multi-physics processes where small changes in material properties or environmental conditions can produce dramatically different flame spread behaviors. Highly resolved CFD models used for these simulations need discretization of the whole volume and hence involve several numerical operations. Even on parallel interfaces, CFD simulations often need long computational hours making them computationally expensive [10].

The material properties used in the CFD models are largely derived from inverse modeling [11, 12] or even empirical approaches [13] making them uncertain and method specific [14]. Moreover, the robustness of these derived parameter values is often subject to the models used to obtain them making uncertainty quantification (UQ) vital for fire science. Lachaud [15] provided an elaborate review of differences and assumptions in pyrolysis models used in 54 simulation tools.

Another key aspect that UQ facilitates for flame spread modeling is robustness of the model itself meaning how well can the model deal with varying and extreme values of the parameters. Brännström et al. [16] highlighted PATO [6] pyrolysis model to break down for certain parameter values derived from inverse modeling using other pyrolysis CFD models.

The simplest and most common approach for UQ is the Monte Carlo method which uses design of experiments generated from the input probability distribution. But this requires high-fidelity (full order) simulations to be performed covering the whole input space and hence the computational resources can become a

bottleneck and for a large number of uncertain parameters, even computationally intractable [17].

The well known solution is use of data-driven surrogates which replace the expensive full order (CFD) model with an efficient approximated one that can predict output fields in matter of seconds rather than hours. However, there is no universal approach for surrogate modeling. Surrogates are based on the data used to train them [18] and are problem dependent [19]. A surrogate that might be able to approximate one case might not be the most efficient or accurate for another one. Dimensionality, physics to be captured, discretization, available data and resources can all have varying impacts on different surrogate methods. The highly non-linear and skewed nature of flame spread outputs [20–22] due to the exponential functions in Arrhenius based pyrolysis models makes the choice of surrogates for fire science a challenge.

Therefore, we present and compare surrogate choices for fire related problems with a specific bench scale setup in mind. We use a flame spread application and focus on propagating parametric input uncertainty through it. Using knowledge accumulated from our previous works [23, 24], we refine our surrogate choices and analysis. Moreover, we increase the model and parameter dimensionalities to enhance the intricacy of surrogate choice problem and make recommendations regarding the most suitable and viable approach for efficient and robust UQ in frame spread modeling. Since new predictions are generated directly from surrogates before extracting statistical information, the surrogates can very well be used simply for predictions. Nonetheless, for a robust predictive performance, a surrogate should also reliably predict all statistical moments.

Dealing with the impact of uncertainties in the input parameter to the output of a CFD model is termed uncertainty propagation (UP) [25] problem. Here aleatoric or stochastic uncertainties are propagated in a forward manner from inputs to the outputs. The aim of UP is to obtain output probability distributions based on input probability distributions. The latter is often multivariate giving rise to joint probability distributions. These distributions are defined via probability density functions (PDFs). Not only computing the output PDFs for each time step and mesh cell of a CFD model can be computationally demanding but also hard to visualize. Statistical moments are often used as an alternative to capture and visualize propagated uncertainties. Most modelers often focus on the first two moments, namely expectation and standard deviation. But this mapping of a distribution to just two moments leads to loss of information [26]. We also showcased in our previous work [24] that high order moments, namely skewness and kurtosis, are of significance where pyrolysis is involved due to the skewed nature of the outputs. Hence we analyze all four moments in this work.

For the purpose of UQ, specifically in flame spread CFD models, we explore four different kinds of surrogate approaches used in machine learning - projection based proper orthogonal decomposition with interpolation (PODI), probabilistic Gaussian process regression (GPR), deep learning autoencoders and operator learning based Fourier neural operator (FNO).

Autoencoders have been the most prominent surrogate approach in fire science with applications ranging from fire detection [27–29] to pyrolysis kinetics [30, 31]. GPR has also been applied for fire detection [32] and pyrolysis [33–35]. POD has only been used for wildfire predictions in fire science [36–38] outside our comparative study of PCE, GPR and PODI for uncertainty propagation in a pyrolysis model [24]. FNO, first introduced in 2020 [39] has gained significant attention in machine learning. It has been reported to yield good performance for wave propagation applications [40, 41], aerodynamics [42] and image classification [43]. Being the most recent surrogate method, it has not been explored in fire science in any capacity so far.

Continuing from our previous works [23, 24], we perform a comparative analysis of machine learning based surrogates that hasn't been conducted before in the fire community or for these surrogates in general. Hence the goal of this work is to compare the different types of machine learning surrogate methods known for their predictive performance in order to provide a comprehensive assessment of uncertainty propagation in a flame spread scenario. We additionally motivate and emphasize with our work the importance of considering higher moments for UQ in fire science due to the intrinsically skewed nature of pyrolysis and flame spread dynamics.

Before we dive into the overview of in this paper, we would like to distinguish between two often related but different concepts used in this paper, namely sensitivity analysis (SA) and uncertainty analysis (UA). Saltelli et al. [44] clarifies the misconception regarding the terminologies as follows: UA is the characterization of the uncertainty in model prediction while SA is the study of how uncertainty in the model output can be apportioned to different sources of uncertainty in the model input. In our study, sensitivity analysis is performed prior to the uncertainty analysis in order to limit the material model parameters to those that are most sensitive. This allows us to reduce the dimensionality of the UP problem from 13 to 5 uncertain parameters and thus avoid the need for simulations in order of thousands to train the surrogates.

The methodology is illustrated as a schematic flowchart in figure 1. Here we use the full order CFD model described in section 2 to generate the training data. The sensitivity analysis is performed for all 13 material kinetic and thermophysical parameters with help of PODI surrogate in section 3. The 5 most sensitive parameters are then used for uncertainty propagation using surrogates explained in section 4. The setup for UQ and comparison measures are detailed in section 5 for both 1D time curve and 2D spatial data. Results are provided in section 6 followed by their discussion and open questions in section 7. Lastly, we

close off with summary and conclusions in section 8.

2 Full Order Simulation Setup

To demonstrate the applicability and analyze the comparison of UQ surrogates, we choose previously developed flame spread model. We do not investigate or argue the validity of assumptions considered in the model itself, instead it is chosen as a flame spread application to showcase and analyze machine learning based uncertainty quantification. The analysis however can also be extended to other flame spread models since the core physical phenomenon starting with highly non-linear pyrolysis remains intact. The exponential nature of Arrhenius pyrolysis model make flame spread a uniquely non-linear case for surrogate modeling due to high level of skewness in the outputs as elaborated in our previous works [23, 24] as well. In this work, we also include mass transport, radiative and convective heat transfer along with turbulence and combustion in gas phase.

Single Burning Item (SBI) is a European standard test for flame spread of building materials and was chosen by MaCFP Workshop [45] as one of the flame growth cases. It is used for analyzing the flame spread in building walls without the influence of ceiling and flooring. A propane burner is used as flame source at the corner. The pyrolyzing material is mounted on a trolley with an exhaust hood above. For the MaCFP setup, the flame behaviour is monitored instrumentally and visually using heat gauges and cameras respectively.

Polymethyl methacrylate (PMMA) composed of monomer methyl methacrylate (MMA) is used as fuel for flame spread in this specific experiment. Two 146 cm×50 cm PMMA slabs are vertically held using a insulation cardboard orthogonal to each other generating a corner, shown in Fig 2. The experiment is performed in normal atmospheric conditions in a 200 cm×200 cm exhaust hood placed 220 cm above the floor. For further details regarding the experiment, we refer to Chaudhari et al. [46].

The basic phenomenology for the setup is as follows. The ignition from the propane burner transfers heat via convection and radiation on the nearby PMMA surface. The transferred heat causes increase in PMMA temperature resulting in pyrolysis, that is thermal decomposition of the solid material into gas. The pyrolzed gas, MMA here, is released which combusts in the air and results in heat feedback loop that sustains and spreads the flame.

The material properties used in our study is based on work by Fiola et al. [47] where they derived these via inverse analysis with experimental data from thermogravimetric analysis, differential scanning calorimetry and controlled atmosphere pyrolysis apparatus II.

CFD Simulation for the validation of data was performed on Fire Dynamics Simulator (FDS) [5] and constituted a part of FDS Validation Guide [48]. This coupled CFD model consists of the following modeling choices. In the solid phase, the pyrolysis behaviour of the PMMA material is modeled using Arrhenius law along with 1D heat conduction model for heat transfer within the material. To model the gas phase, Large Eddy Simulation (LES) is used by default in FDS. Mixing-controlled combustion is assumed while for for radiation modeling, radiative transfer equation (RTE) is applied. We use the existing model since the focus of this study is uncertainty quantification rather than flame spread modeling itself.

We use coarse mesh of 4 cm with 28×28×70 cells representing the whole domain for our uncertainty analysis because we need to run these simulations many times in order to draw comparisons between machine learning surrogates and need a *True* measure of statistics via 10000 converged Monte Carlo simulations. Hence, the model is not finely resolved, which would require mesh cell size on the order of millimeters rather than centimeters. As we consider only the propagation of stochastic uncertainties through the model, we assume this model to be the ground truth for the surrogates. However, as the surrogates are learning the same physics, a more resolved model with higher mesh refinement will provide a similar comparison for UQ via the surrogates. Hence, we believe the findings of this UQ study are also applicable to other detailed and highly resolved flame spread models.

3 Sensitivity Analysis

The flame spread model described above consists of many material parameters that impact the output, though not equally. To rank and quantify how much uncertainty each of these parameters contributes to the model output over the entire input sample space, we apply global sensitivity analysis [49]. Sobol sensitivity indices [50] are a common choice for this which use standard deviation as the sensitivity measure to quantify the spread of output caused by the input uncertainty.

Though standard deviation is often used as a standard to measure uncertainty and thereby sensitivity, it is just one of the moments to quantify the whole uncertainty response given by probability density function. Neglecting the other moments like skewness and kurtosis will result in loss of information for a robust and extensive sensitivity analysis. Borgonovo [51] proposed a moment-independent sensitivity measure called delta to use the entire PDF for sensitivity evaluation. The idea is use conditional probability densities instead of conditional variances in Sobol sensitivity. We measure the expected shift in area under the PDF from unconditional output density to conditional density of the output given that one of the parameters assumes a fixed value. Delta is then defined as half of this expected shift to normalize it between 0 and 1.

Another advantage of delta sensitivities over Sobol sensitivities is that the inputs need not be independent. But, the downside of this method is an increased computational expense to achieve converged results as an additional integral for evaluating the area under the PDF is needed for each run. And therefore, its appli-

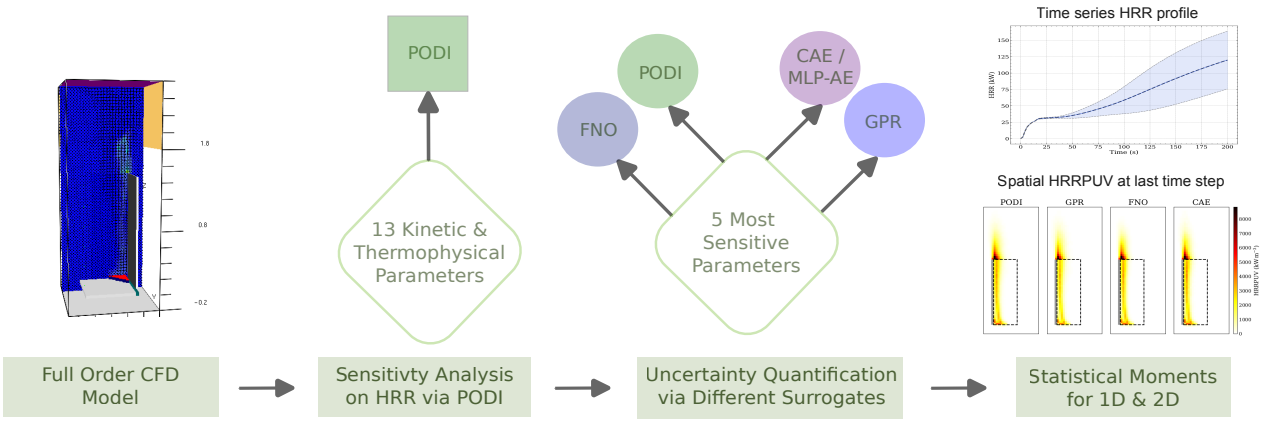


Figure 1: A simplified workflow showing the route to uncertainty quantification using machine learning surrogates. Full order CFD model is used for generating training data for surrogates for both sensitivity and uncertainty analysis. To speed up the sensitivity analysis, PODI is applied as a surrogate, motivated from our prior work, on the l^2-norm of HRR. From 13 material parameter, 5 most sensitive ones are used for uncertainty quantification to generate moments as explained in sections 4 and 5. Of note, the PODI in sensitivity analysis employs different data to that in uncertainty analysis, although the underlying approach is the same.

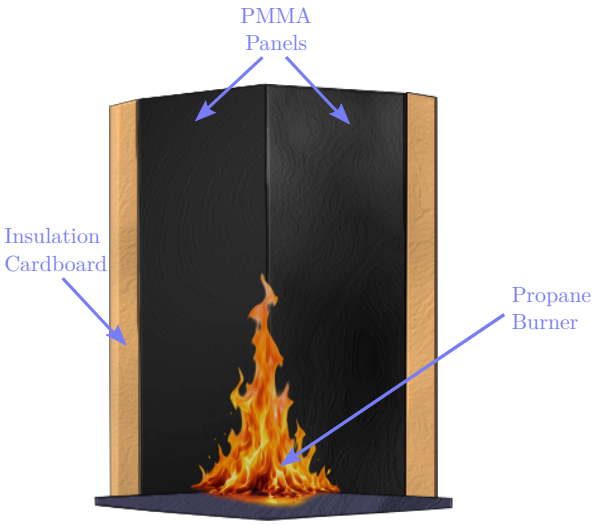


Figure 2: Insulated PMMA panels with a propane sandbox burner in SBI test.

cation is likely to be confined to models where thousands of evaluations can be performed in a reasonable amount of computing time [52].

To mitigate this, we first apply quasi-random sampling using Sobol sequence [50] as recommended by Borgonovo [51]. Sobol sampling was also found to be an effective sampling method in our previous work [23]. Secondly, motivated from our work on comparison of surrogates for pyrolysis modeling [24], we use PODI (describe below) to generate surrogate models for evaluating delta sensitivity indices.

For input uncertainty, we use uniform distribution with ±20% standard deviation around the nominal values from Fiola et al. [47] for 13 input parameters shown alongside converged delta sensitivities in figure 3. l^2-norm of HRR over time is used as the output measure. Varying 13 inputs by ±20% does generate a substantial spread of HRR that might help us capture the interaction effects at the extremities and hence generate a robust sensitivity analysis. We also plot converged main effect Sobol indices for comparison. These only consider the average effect on the output variance of varying a parameter alone without interaction effects between parameters. Convergence of sensitivities is performed with different number of samples. To obtain converged sensitivities, PODI is trained on 10000 FDS training data pairs and subsequently 100000 new sample predictions are generated for evaluation of sensitivities.

For this setup, activation energy E is by far the most sensitive parameter similar to what was observed in our previous sensitivity studies in [23, 24]. Following the ranking, radiative fraction χ_r and heat of combustion ΔH_{mma} of MMA show second and third highest sensitivities. Emissivity ϵ and specific heat capacity of PMMA c_p are seen to be marginally ahead others for fourth and fifth positions. Tasia et al. [53] also observed c_p to be important thermophysical parameter for HRR in Cone Calorimeter simulation and ϵ along with c_p for rate of spread in horizontal flame spread simulation. Thus we include these five for our UQ study below.

4 Surrogate Approaches

For a comprehensive comparison of surrogates for flame spread, we explore four prominent and distinct classes of data-driven machine learning methods used to create surrogates in this section. PODI is a linear model order reduction method employed profoundly in engineering applications. Autoencoder, on the other hand, constitutes a non-linear model order reduction method

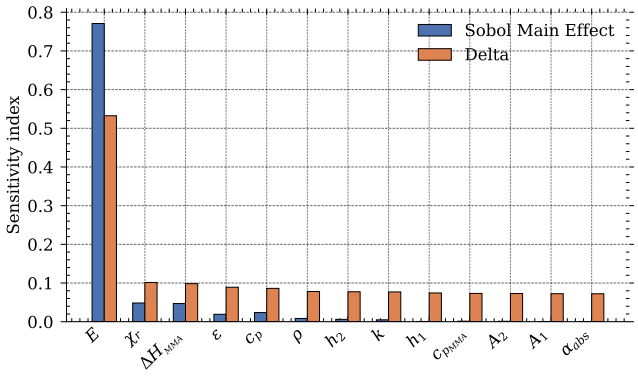


Figure 3: Converged delta and main effect Sobol sensitivity indices for the input parameters.

via deep neural networks. FNO learns solution operator by parameterizing neural networks in the Fourier domain and GPR is a well founded Bayesian approach for regression. We introduce targeted modifications below to the surrogates such that they are more tailored to the flame spread model.

4.1 Proper Orthogonal Decomposition with Interpolation

In our work we apply proper orthogonal decomposition with interpolation (PODI) approach using method of snapshots (MOS) and radial basis functions (RBF) described below.

4.1.1 Proper Orthogonal Decomposition aims at reducing the dimensionality of the data (system in general) by projecting the data onto a lower dimensional orthogonal subspace. We organize our output data that can be any field like heat release rate, temperature or burning rate (flattened in case of 2D) of dimension m in a snapshot matrix $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n] \in \mathbb{R}^{m\times n}$ for input samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ where d is the number of random variables (uncertain model parameters for us) and n is the number of samples.

Applying single value decomposition (SVD) on $\mathbf{Z}$:

$$\mathbf{Z} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \tag{1}$$

$\mathbf{U} \in \mathbb{R}^{m\times m}$ is an orthogonal matrix whose columns are the left singular vectors or POD modes $[\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m]$.
$\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is a diagonal matrix containing the singular values $\{\sigma_1, \sigma_2, \ldots, \sigma_s\}$ with $s = \min(m, n)$, arranged in descending order.
$\mathbf{V}^T \in \mathbb{R}^{n\times n}$ is an orthogonal matrix whose rows are the right singular vectors $[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$.

For a reduced approximation, we retain only a few POD modes corresponding to the r largest singular values and truncate the rest. This truncation is done in our case using optimal singular value hard threshold [54] that adapts optimally to unknown rank and noise level. We therefore obtain a reduced-basis approximation known as a reduced order model (ROM):

$$\mathbf{Z} \approx \mathbf{Z}_{rom} = \mathbf{U}_{rom}\mathbf{A}_{rom} \tag{2}$$

where $\mathbf{U}_{rom} \in \mathbb{R}^{m\times r}$ is the truncated POD basis matrix and $\mathbf{A}_{rom} \in \mathbb{R}^{r\times n}$ is the truncated POD coefficient matrix defined as:

$$\mathbf{A}_{rom} = \mathbf{\Sigma}_{rom}\mathbf{V}_{rom}^T = \mathbf{U}_{rom}^T\mathbf{Z} \tag{3}$$

containing POD coefficient vectors $[\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n]$.

However, if the snapshot matrix is thin ($n \ll m$), MOS [55] provides an efficient alternative to SVD for evaluating POD. In MOS, we use the eigenvalue equation for snapshot covariance matrix:

$$\mathbf{Z}^T\mathbf{Z}\Phi = \Lambda\Phi \tag{4}$$

Now, just for comparison, let's look at the SVD of $\mathbf{Z}^T\mathbf{Z}$:

$$\mathbf{Z}^T\mathbf{Z} = \mathbf{V}\Sigma^2\mathbf{V}^T \Leftrightarrow \mathbf{Z}^T\mathbf{Z}\mathbf{V} = \mathbf{V}\Sigma^2 \tag{5}$$

Note from equations (4) and (5), the right singular vectors of $\mathbf{Z}$ are the same as eigenvectors of $\mathbf{Z}^T\mathbf{Z}$ and the eigenvalues of $\mathbf{Z}^T\mathbf{Z}$ are squares of singular values of $\mathbf{Z}$. We can already do a truncation here for the eigenvalues Λ and the eigenvectors $\Phi = \mathbf{V}$. The truncated POD basis can be evaluated as:

$$\mathbf{U}_{rom} = \frac{1}{\sqrt{\Lambda_{rom}}}\mathbf{Z}\cdot\mathbf{V}_{rom} \tag{6}$$

The truncated singular values are given as $\Sigma_{rom} = \sqrt{\Lambda_{rom}}$ and the truncated POD coefficient matrix follows as before in equation (3).

4.1.2 Radial Basis Functions are applied in order to make predictions using ROM for new untrained input samples. The POD coefficients $\mathbf{a}_j, j = 1, \ldots, n$ are related to the input samples $\mathbf{x}$ as:

$$\mathbf{a}_j(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{b}_i g(\|\mathbf{x} - \mathbf{x}_i\|) \tag{7}$$

where $\mathbf{b}_i \in \mathbb{R}^r$ are the vectors of RBF coefficients, and $g(\|\mathbf{x} - \mathbf{x}_i\|)$ are the RBF functions with center points $\mathbf{x}_i$ taken as the training sample points. We use cubic interpolation $g(r) = r^3$ for our case as these performed much better than the others like thin plate spline $g(r) = r^2 log r$ and quintic $g(r) = r^5$ RBF functions.

For the coefficients in the reduced POD coefficient matrix $\mathbf{A}_{rom}$, the RBF interpolation can be expressed in the matrix form:

$$\mathbf{A}_{rom} = \mathbf{B}\mathbf{G} \tag{8}$$

where $\mathbf{B} = \{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\} \in \mathbb{R}^{r\times n}$ and

$$\mathbf{G} = \begin{bmatrix} g(\|\mathbf{x}_1 - \mathbf{x}_1\|) & g(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \cdots & g(\|\mathbf{x}_1 - \mathbf{x}_n\|) \\ g(\|\mathbf{x}_2 - \mathbf{x}_1\|) & g(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \cdots & g(\|\mathbf{x}_2 - \mathbf{x}_n\|) \\ \vdots & \vdots & \ddots & \vdots \\ g(\|\mathbf{x}_n - \mathbf{x}_1\|) & g(\|\mathbf{x}_n - \mathbf{x}_2\|) & \cdots & g(\|\mathbf{x}_n - \mathbf{x}_n\|) \end{bmatrix}$$

For positive definite and invertible matrix $\mathbf{G}$ [56], it follows:

$$\mathbf{B} = \mathbf{A}_{rom}\mathbf{G}^{-1} \tag{9}$$

This leads to the approximated output for new samples $\mathbf{x}_*$ using PODI as follows:

$$\mathbf{z}_{rom}(\mathbf{x}_*) = \mathbf{U}_{rom}\mathbf{B}\mathbf{g}(\mathbf{x}_*) \tag{10}$$

4.2 Autoencoder

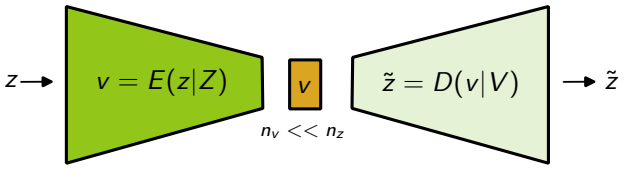


Figure 4: A simple autoencoder schematic where snapshot data z is approximated by compression in encoder and expansion in decoder.

Autoencoders [57] are neural network structures that try to study the hidden patterns present in the snapshot data Z. An autoencoder A discovers latent variables by squeezing the data through the bottleneck using an encoder E and reconstructs them back to original data using a decoder D.

figure 4 shows the basic idea behind an autoencoder where snapshot data $z \in Z \subset \mathbb{R}^{m\times n}$ is fed into the encoder that compresses the data to latent space variables $v \in V \subset \mathbb{R}^{v}$. This compressed information is then expanded back to the original space Z via a decoder $\tilde{z} = D(v|V)$. A reconstruction loss, typically mean square error, is computed and optimized between the original data and the reconstructed data to improve the neural network recursively.

In order to make new predictions for input samples $\mathbf{x}$, we apply RBF interpolation to the latent space as above in section 4.1.2. Here, instead of the POD coefficients in equation (7), the latent variables are approximated and fed into the decoder for expansion to the original space.

So far, we have not discussed the structure of the encoder and decoder. Since, in this paper we deal with both 1D (heat release rate) and 2D snapshot data (for heat release rate per unit volume, wall temperature and burning rate), we use two different structures of autoencoders. For 1D, we apply a fully connected multi layer perceptron (MLP) [58] architecture and for 2D, we apply convolutional neural network (CNN) [59]. Figure 5 shows the an example of the former and figure 6 the latter.

The issue with using MLP-AE for 2D spatial data is that neural networks treat all input mesh cell data equally and hence requiring large number of parameters to accurately capture the spatial patterns. Convolutional autoencoder (CAE) overcomes this by using multi-dimensional filters and thus requires training of fewer parameters [60]. These filters are applied within each convolutional layer to detect features generating the feature maps (output channels) [61]. Maxpooling

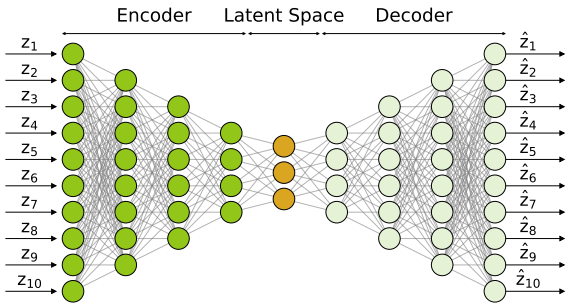


Figure 5: An simple MLP autoencoder with three hidden layers each for encoder and decoder. (This is only for illustration purpose. Our MLP autoencoder consists of much larger number of neurons in each layer. These are chosen based on hyper-parameter optimization (HPO))

layers are applied in the encoder to compress the 2D snapshots recursively while upsampling layers in the decoder expand the latent variables back to the original dimension.

4.3 Gaussian Process Regression

Gaussian process regression (GPR) uses bayesian inference for updating a prori distribution defined by a kernel to a posterior distribution that can be used to make new predictions. The positive definite kernel $k(\mathbf{x}, \mathbf{x}')$ maps a pair of inputs $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{x}' \in \mathbb{R}^d$ to a complex valued $\mathbb{C}$ [62]. For a finite set of inputs, the kernel can also be defined as a covariance matrix:

$$\mathbf{K} = k(\mathbf{x}, \mathbf{x}) \tag{11}$$

An independent identically distributed Gaussian noise ϵ with variance σ_ξ^2 is added along the diagonal to the covariance matrix as $\mathbf{K} + \sigma_\xi^2\mathbf{I}$. Other than observation noise, σ_ξ^2 also known as "nugget" [63], helps in numerical stability.

Predictions $f(\mathbf{x}_*)$ for the new samples $\mathbf{x}_*$ in GPR are evaluated as:

$$f(\mathbf{x}_*) = \mathbf{K}_*^T(\mathbf{K} + \sigma_\xi^2\mathbf{I})^{-1}\mathbf{y} \tag{12}$$

where $\mathbf{K}_* = k(\mathbf{x}, \mathbf{x}_*)$.

The choice of kernel function influences the GPR model's predictability significantly. Based on its empirical success previously [24] and in this study, we focus on an isotropic kernel below which is a function of Euclidean length $r = r(\mathbf{x}_i) = \|\mathbf{x} - \mathbf{x}_i\|$, same as for the RBF functions in section 4.1.2.

$$\sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{r}{\ell}\right)^{\nu} K_\nu \left(\sqrt{2\nu}\frac{r}{\ell}\right) \tag{13}$$

This Matern kernel consists of hyper-parameters: signal variance σ_f, length scale l, scale mixture α and smoothness factor ν. With functions modified Bassel K_ν and gamma $\Gamma(\nu)$ [62].

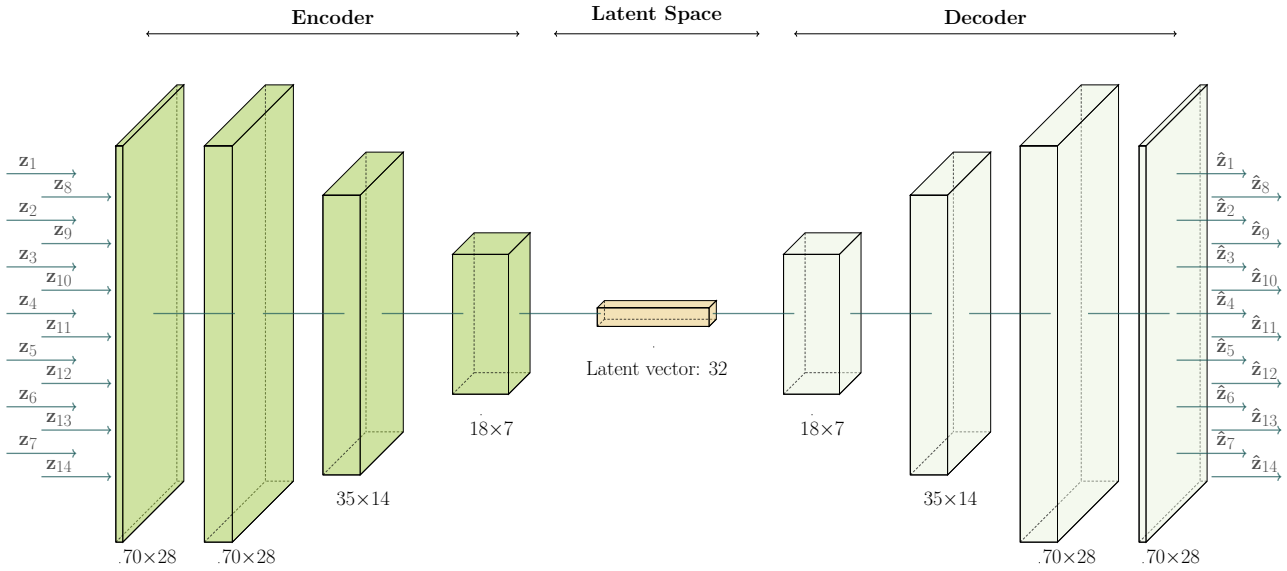


Figure 6: Our convolutional autoencoder with three maxpooling and upsampling layers by a factor of two. The number of channels and latent vector illustrated by thickness of each layer are optimized and thus vary for each output.

The hyper-parameter values θ above are optimized by maximizing the log marginal likelihood:

$$\log p(\mathbf{y} \mid \mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^T(\mathbf{K}+\sigma_\xi^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log\det(\mathbf{K}+\sigma_\xi^2\mathbf{I}) - \frac{n}{2}\log(2\pi) \tag{14}$$

4.4 Fourier Neural Operator

FNO is an operator learning method [64] with Fourier transform at its core. It has an additional advantage of being resolution invariant meaning it can be trained on low resolution data and evaluated at high resolution data [65].

The basic idea entails the following. For further details, kindly refer to [39].

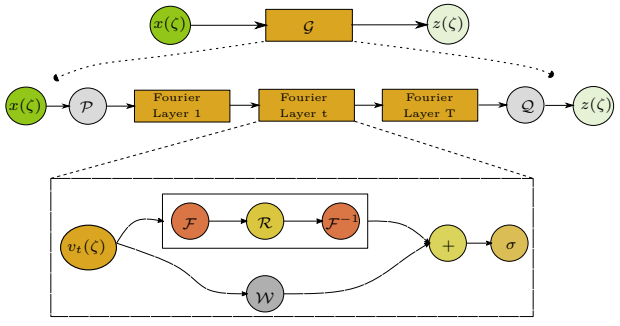


Figure 7: Sequence of operations in a FNO Model. In our case, $x(\zeta) \in \mathbb{R}^{n\times d}$ represents d parameters expanded to time steps (1D) or mesh cells (2D) of size n and $z(\zeta) \in \mathbb{R}^{n\times k}$ represents single output field with k=1 and $T = 4$ Fourier layers are used.

Let ζ be the n-point discretization of the domain Ω, we wish to learn an solution operator $\mathcal{G}$ from input $x(\zeta) \in \mathbb{R}^{n\times d}$ to output $z(\zeta) \in \mathbb{R}^{n\times k}$ such that:

$$z(\zeta) = \mathcal{G}x(\zeta) \tag{15}$$

First a neural operator $\mathcal{P}$ is applied to lift the input to a higher dimension channel space:

$$v_1(\zeta) = \mathcal{P}(x(\zeta)) \tag{16}$$

This is followed a series of T Fourier convolution operations $v_t(\zeta), t = 1, \ldots T$ with trainable parameters θ:

$$v_{t+1}(\zeta) = \sigma\left(\mathcal{F}^{-1}\left(R_\theta \cdot \mathcal{F}(v_t)\right)(\zeta) + W v_t(\zeta)\right) \tag{17}$$

Each operator performs a Fourier transform $\mathcal{F}$, truncation of higher modes, learnable linear transform R_θ and inverse Fourier transform $\mathcal{F}^{-1}$ in the frequency domain.

Subsequently, similar to a neural network, this is added to the bias W in real domain and followed by an activation function. Finally another neural operator $\mathcal{D}$ is applied to project to the real output space.

$$z(\zeta) = \mathcal{D}(v_T(\zeta)) \tag{18}$$

Although FNO is efficient on rectangular domains, it's effectiveness for arbitrary non-uniform meshes remains an open question [66]. Since, our case deals only with rectangular grid, we do not encounter or investigate this bottleneck with FNO.

5 Setup for Uncertainty Analysis

For the training of surrogates in section 4, the input parameters are sampled and the FDS model is subsequently ran to generate the input-output pairs. The computational expense of running these simulations increases linearly with the number of samples. Around 4

hours on a single (AMD EPYC 7452) core is needed for each simulation. We use DAKOTA [67] to distribute parallel jobs and process data for FDS training simulations on HPC cluster.

Since no tool encompasses all the aforementioned surrogate approaches, surrogate models are developed in Python language using different libraries available. For defining probability distributions and generating samples, we use Chaospy [68]. SAlib [69] is used for sensitivity analysis, EZyRB [70] for PODI, scikit-learn [71] for GPR, Keras [72] for CAE and NeuralOperator [73] for FNO. Statistical moments from the predictions are evaluated using SciPy [74]. For hyper-parameter optimization (HPO) in autoencoders and FNO, we use Raytune [75]. Train-test split of 80%-20% is used to obtain the optimized hyper-parameters with the help of asynchronous successive halving algorithm [76] scheduler. We generate 1000 configurations (samples) in total to be explored from the search spaces mentioned below.

In order to limit the number of parameters for HPO, we largely use default settings in the libraries listed above. Learning rate of 1e-3, epochs of 1000 for training and rectified linear unit (ReLU) as activation function is applied for all neural networks. The search spaces for HPO is chosen based on initial trials and are listed in table 1. Here, the growth value hyperparameter sets the neuron/channel count for each decoder layer. Starting from the base hidden/convolutional layer, each subsequent layer's count is calculated as the truncated (toward-zero) value of the product growth value $\times$ neurons/channels. In our study, the encoder for each type is simply the decoder mirrored. Furthermore, in 2D FNO, same number of modes are chosen for both directions and for 1D FNO, only a single FNO mode is used. It is to be additionally noted that the HPO is done again for every sample number plotted in Results section 6. This is to ensure that UQ results across different training data sizes are independent of each other. Hence, for each surrogate method, the best performing model is selected at each training data size enabling a fair comparison of how training data volume affects the surrogates accuracies.

For PODI and GPR, interpolation functions and isotropic kernels are compared and chosen as described in section 4.

Similar to our previous UQ comparison work [24], we use uniformly distributed Sobol samples for the input parameters listed in section 3. Since there are more uncertain parameters included in this study, we decrease input standard deviation around nominal values from 20% to 10%. We observed that for 1D HRR data, use of 20% input standard deviation resulted in the output HRR to vary excessively with a standard deviation of more than 80 kW (i.e. over 60% around expectation) at the last time step of the simulation.

The joint uniform input distribution is additionally transformed to a uniform cube ranging from 0 to 1 using Rosenblatt transformation [77]. The samples are thus taken from this distribution. This prevents presence of unnecessarily high gradients and exploding activation functions enabling better performance for the surrogates with no loss in generality.

Surrogate	Hyperparameter	Range
MLP-AE	Neurons growth value	1.5 - 4 (continuous)
	Base neurons	5 - 16 (discrete)
CAE	Channels growth value	[1.5, 2, 4]
	Base channels	[8, 16, 32, 64]
	Latent dimension	[5, 8, 16, 32]
	Batch size	[4, 8, 16, 32, 64]
FNO 1D	Channels	[8, 16, 32, 64]
	Batch size	[4, 8, 16, 32, 64]
FNO 2D	Fourier modes	8 - 16 (discrete)
	Channels	[16, 32, 64]
	Batch size	[4, 8, 16, 32, 64]

Table 1: Hyperparameters and their ranges or choice of discrete values used in hyperparameter optimization. The hyperparameters for the surrogates not listed in table 1 are specified values based on prior optimization trials.

To obtain statistical moments, each surrogate is used to predict outputs for 10000 new samples from the same joint probability distribution as used for training data. From these surrogate outputs, statistical moments are evaluated and consequently used for comparison. Figure 8 entails the basic idea for training and prediction to generate moments via surrogates.

Use of more than 10000 samples from our 5-dimensional input parameter space for predictions does not improve the output moments, hence we choose this converged number. For skewness and kurtosis in 2D cases, we zero out mesh cells where expectation is less than 1% of the global maximum, since in these regions floating point noise caused by division of near zero standard deviation can be numerically explosive and physically meaningless.

To analyze and compare the accuracy of surrogates, input-output pairs from FDS simulations ranging from 20 to 500 are used for training. Since, less than 20 samples are observed to be highly inaccurate for the surrogates especially considering a five dimensional input space, we take this as the lower threshold. On the other hand, 500 training pairs (in offline phase) are already computationally expensive for most high-fidelity models and the aim is to choose the most robust approach with the least amount of simulations. Hence, we choose 500 data samples as our upper threshold for training data. Furthermore, this range is consistent with our surrogate comparison is earlier study [24].

The data from FDS model is extracted for 100 time steps. Each surrogate is trained for a single time step. Since we are dealing with two different dimensional datasets, the surrogates are assessed based on different comparison measures. For one dimensional HRR data for 100 time steps, training data composes of spatially summed HRR over the computational domain. For surrogate comparisons concerning heat release rate

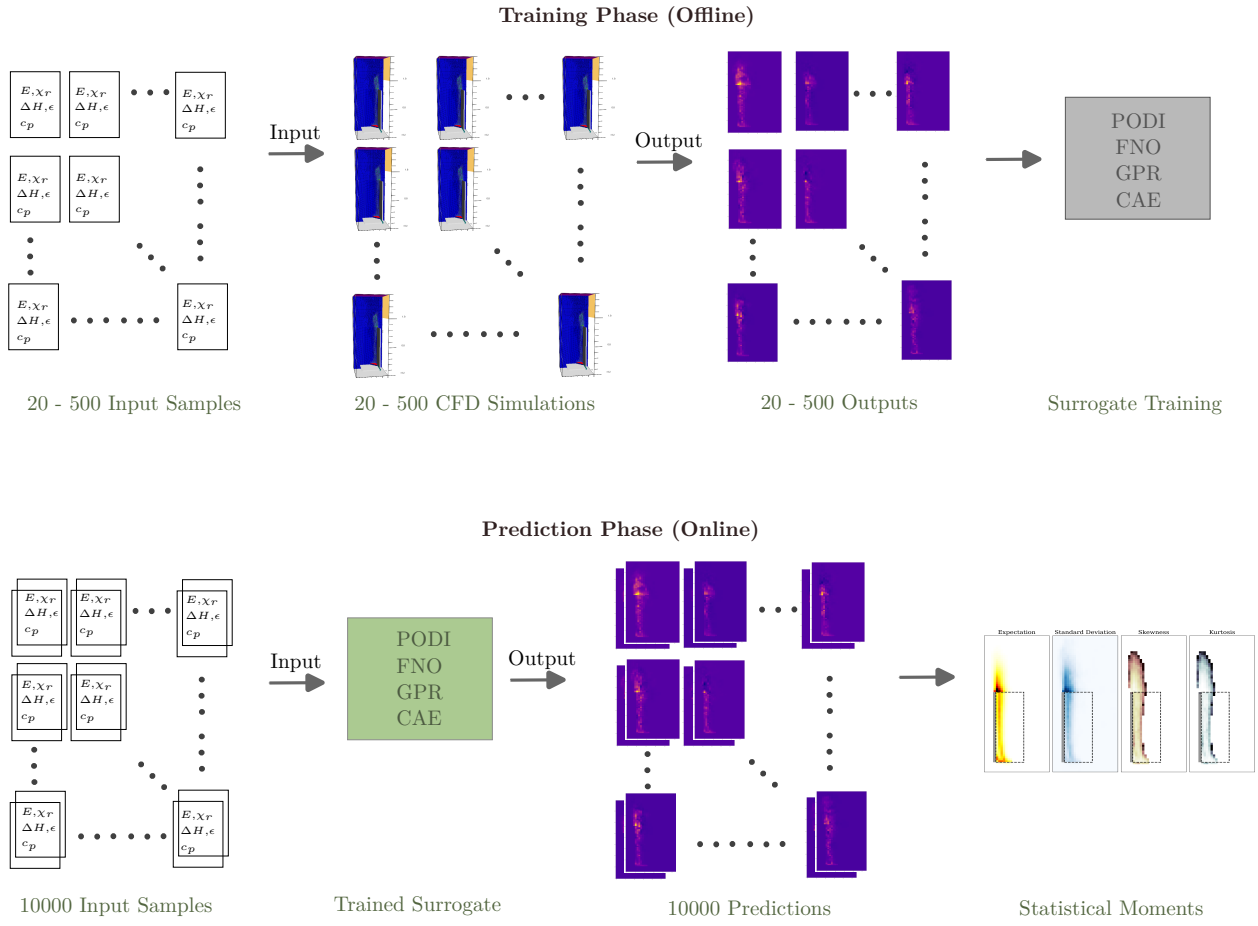


Figure 8: A schematic showing the computationally expensive offline training phase on the top where CFD simulations are performed to generate input-output pairs for training of surrogate. Below is the online prediction phase that uses trained surrogate to generate new data. In our study, 10000 new predictions are generated from new 10000 input samples to obtain final moments. This example shows 2D HRRPUV data, however, the same procedure is applied for 1D HRR and 2D wall temperature and burning rate data.

per unit volume (HRRPUV) near the PMMA surface, burning rate (BR) and wall temperature (WT) at the PMMA surface, we use 2D spatial data at the last time step.

To compare each moment i, generated via surrogate S, a relative error $\mathcal{R}_i$ is defined:

$$\mathcal{R}_i = 100 \cdot \frac{\left\| M_i^{(0)} - M_i^{(S)} \right\|}{\left\| M_i^{(0)} \right\|} \tag{19}$$

$M_i^{(S)}$ are the moments from surrogate predictions and $M_i^{(0)}$ are the *True* moments obtained from converged Monte Carlo results using 10000 full order FDS simulations. To clarify, this is the same number of samples used for predictions from surrogates in order to generate moments. For 2D case, the difference $M_i^{(0)} - M_i^{(S)}$ is evaluated for each mesh cell before applying the l^2-norm. Whereas for 1D case, the difference is evaluated for each time step. This is done to ensure that the surrogates are compared for each point in time and space against the *True* result. Moreover, each time step and mesh cell is weighted equally for the moment comparison.

It is to be noted that since the predictions are done for all time steps in 1D case and mesh cells in 2D case, one is free to choose focus on a specific time range in the simulation or a specific region of the PMMA surface respectively, via an appropriate norm. For simplicity of the overall comparison, we choose to assign equal weighting in both cases.

Furthermore, for a collective comparison for all moments, we define a weighted error $\mathcal{W}$ incorporating weighted importance and the relative error (19) above for each moment:

$$\mathcal{W} = 100 \cdot \sum_{i=1}^{4} w_i \frac{\left\| M_i^{(0)} - M_i^{(S)} \right\|}{\left\| M_i^{(0)} \right\|} \tag{20}$$

The weights w_i are chosen empirically according to the importance of the moment for UQ. These are listed in table 2.

Moment M_i	Weight w_i
Expectation	0.45
Standard Deviation	0.35
Skewness	0.15
Kurtosis	0.05

Table 2: Statistical moments and their respective weights to define the weighted percentage error $\mathcal{W}$ in equation (20).

6 Results

We start with comparing moment predictions for HRR. Figure 9 shows the weighted error percentage based on equation (20).

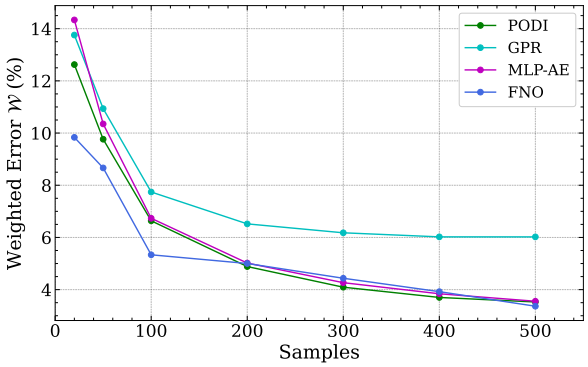


Figure 9: Weighted percentage error defined in equation (20) for the four approaches over different number of training data samples.

FNO is observed to be the most accurate in general based on the weighted error $\mathcal{W}$ followed by PODI and MLP-AE while GPR under-performs compared to the rest. The similarity in weighted error profiles of PODI and MLP-AE amount to the same RBF interpolation used for both. Though we tried other interpolation functions as well, these performed significantly poor relative to cubic RBF demonstrating the impact of interpolation methods for these two surrogates. Interestingly, FNO showed near-linear convergence for samples ranging form 100 to 500. We would like to point out that HPO was performed for each number of samples separately to generate the best result using each surrogate. At 500 samples, all surrogates except GPR exhibit errors below 4% with FNO showing the lowest error of 3.36%.

To access the error contributions of each moment to the overall weighted error $\mathcal{W}$, we list the relative errors in table 3. Expectation and standard deviation is predicted almost precisely by all surrogates. Hence for 1D data, all the four surrogates are adequate for quantifying the first two moments. The differences becomes more appreciable for skewness and kurtosis which are usually the weak element of machine learning surrogates. This is due to the fact that the surrogates are trained based on loss functions that prioritize accuracy near the bulk of the data, rare events that lead to higher moments are under represented and at times neglected. Hence skewness and kurtosis requires more number of samples especially when there are a larger number of uncertain variables - curse of dimensionality.

Relative Error (%)	Expectation	Standard deviation	Skewness	Kurtosis
PODI	0.11	0.40	10.9	34.0
GPR	0.11	0.39	21.6	52.1
FNO	0.096	0.35	13.0	25.0
MLP-AE	0.12	0.44	11.0	34.0

Table 3: Relative errors for heat release rate moments evaluated using surrogates trained on 500 samples.

Moreover, we do not aim complete convergence for the surrogates here as the goal is to compare the accuracy and performance of these surrogates for least number of training data possible. To access the errors qualitatively, we first plot FNO's first two moments for 500 training samples against converged *True* result in figure 10. We do not see any visible difference for the expectation as well as for standard deviation. The standard deviation is plotted above and below expectation which would be reflective of a normal distribution.

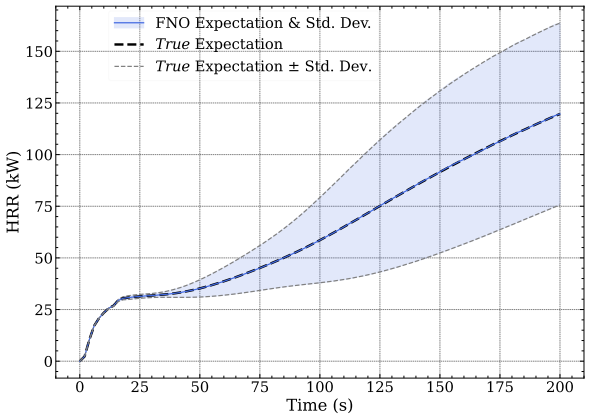


Figure 10: Comparison of standard deviation around expectation for FNO using 500 training data samples and *True* result derived from 10000 Monte Carlo simulations.

Another alternative to view our output uncertainty is using quartiles, shown in figure 11 using FNO. Here, the gap between expectation and median emphasizes appreciable skewness in the output.

The most informative approach to observe output uncertainty is by plotting the PDF itself which we do so in figures 12 and 13 corresponding to two specific times of 150 s and 200 s. These are chosen as they represent times with significant standard deviation where expectation and median deviate substantially at 150 s and coincide closely at 200 s, as observed in figure 11.

Comparing the performance of surrogates at 150 s using 50 training samples in figure 12(a), FNO is observed to match *True* curve the best. PODI and MLP-AE show very similar profiles and GPR is visibly the

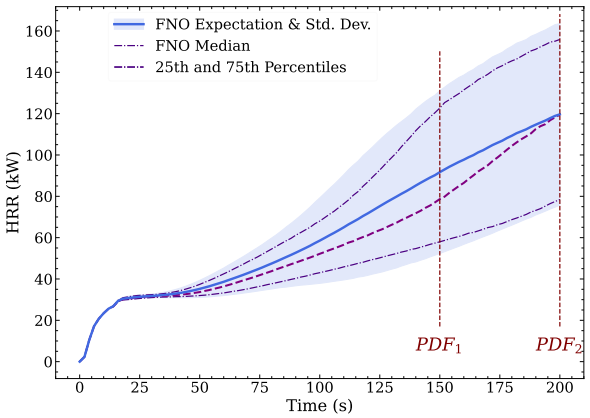


Figure 11: Quartiles along with first two moments evaluated using FNO with 500 training data samples.

weakest in capturing the *True* PDF. These results are reflective of the weighted percentage error plotted in figure 9.

For 500 training samples in figure 12(b), FNO outshines again in capturing the overall PDF profile and is almost indistinguishable to the *True* curve. There is an improvement in PDF profiles generated from the rest but still the second peak is smoothed out. Though these small deviations are inconsiderable for UQ compared to the choice and input uncertainties of the uncertain parameters, they provide insights about the robustness and precision of the surrogates.

The difference between surrogate predictions is observed to be more pronounced for PDFs at 200 s plotted in figures 13. All surrogates with exception of FNO fail to represent the existence of two peaks using 50 training samples plotted in figure 13(a). The PDF prediction improves considerably with increase in training samples to 500 in figure 13(b) for all surrogates with FNO again outperforming other surrogates that underpredict the first peak as in figure 12(a).

It is important to note here that even when median and expectation are relatively similar at 200 s in figure 11, the PDFs in figures 13 are non-gaussian and skewed, thereby pointing out that quartiles are also not a sufficient measure for a thorough and robust uncertainty quantification. Therefore, a reliable model comparison for flame spread scenarios of similar scale requires a proper assessment of higher moments.

Another aspect to note in figures 13 is the standard deviation around the expectation (shown in dotted vertical lines) not capturing the peak in the PDF which again proves the importance of capturing higher moments for flame spread problems.

Furthermore, the consistency of results depicted in the PDFs, at different times and training samples, compared to weighted percentage error $\mathcal{W}$ in figure 9 demonstrates that our error definition in equation (20) along with weighting in table 2 are adequate and reasonable in this case.

We further dive down to analyze the 2D statistical moments at 200 s. We commence with heat release rate per unit volume (HRRPUV). Figure 14 shows the moments for surrogates using 500 training samples compared to *True* in the first column. Qualitatively all surrogates yield accurate predictions for expectation and standard deviation.

We would like to point out that the sharp horizontal edge at the top of PMMA in the expectation and standard deviation plots is mainly to the coarseness of the mesh and this effect reduces with mesh refinement. However, quality of the deterministic CFD simulations is not the focus of the current study as this would most likely not impact the surrogate comparison. Furthermore, smoothened data via mesh refinement will rather further facilitate the surrogates due to diminution of sharp gradients in the mesh data.

The differences in the surrogates performance become more apparent for skewness and kurtosis. To investigate this quantitatively, table 4 shows the relative errors $\mathcal{R}_i$ for each moment i and figure 15 shows the weighted error $\mathcal{W}$ for each surrogate. PODI provides

Relative Error (%)	Expectation	Standard deviation	Skewness	Kurtosis
PODI	4.4	10.1	71.9	84.6
GPR	12.3	26.6	56.8	79.2
FNO	9.3	18.2	68.0	83.7
CAE	7.8	18.6	49.4	75.3

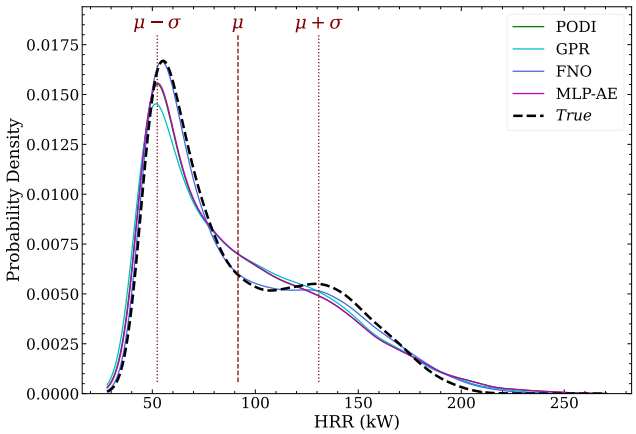
Table 4: Relative errors for heat release rate per unit volume (HRRPUV) moments at 200 s evaluated using surrogates trained on 500 samples.

overall best weighted error due to the most accurate prediction of first two moments. Since for expectation and standard deviation the relative error using PODI in table 4 is almost half of that using other surrogates, it's weaker performance for skewness and kurtosis is subdued. Similarly for GPR, highest weighted error is observed relative to other surrogates because of the weakest prediction of first two moments. Thus, here the weights in table (2) play a significant role. Another choice of these weights might render another surrogate to be best overall.

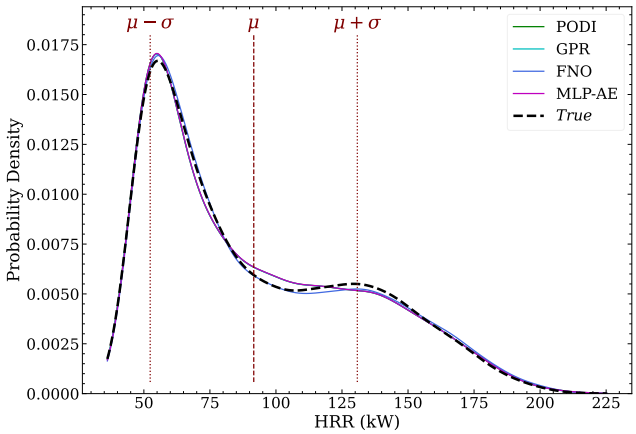
CAE provides a compromise between predictability of all moments. It's not as accurate as PODI for expectation and standard deviation but is better than GPR. Whereas for skewness and kurtosis, it generates the most accurate result.

It is up to the modeler to choose the best surrogate based on the efficacy required. However, considering the aforementioned skewed PDFs, a reasonably good accuracy for higher moment evaluation especially skewness is vital to categorize UQ in flame spread problems.

Next, to compare the uncertainty profiles for PMMA fuel consumption, we investigate surrogate moment predictions for burning rate (BR) at the wall. In figure 16, qualitatively all moment predictions via surrogates look indistinguishable. Hence we compare further quantitatively the relative errors $\mathcal{R}_i$ in table 5 and weighted error $\mathcal{W}$ in figure 17. GPR again gives the

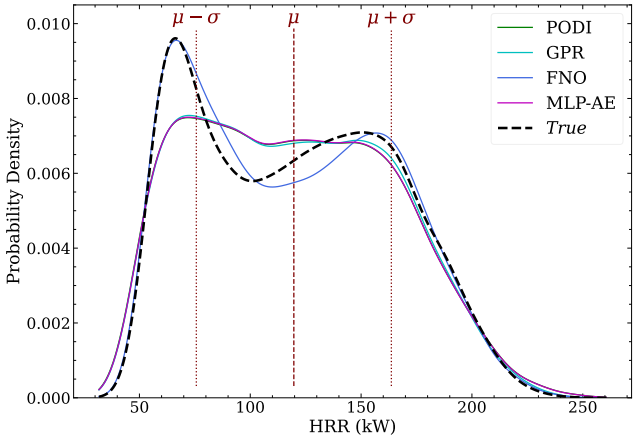


(a) PDF at 150 s generated by surrogates using 50 training data samples.

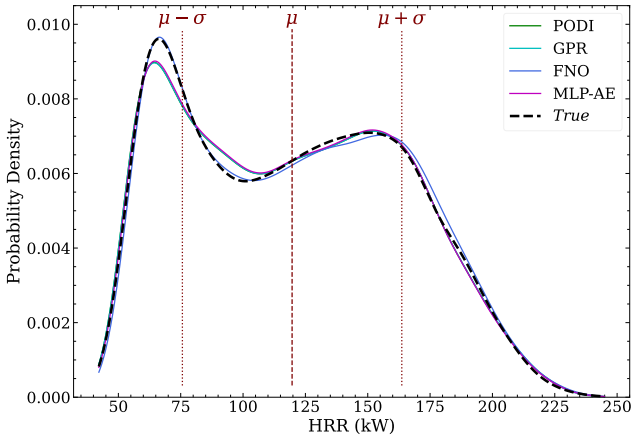


(b) PDF at 150 s generated by surrogates using 500 training data samples.

Figure 12: PDFs for HRR at 150 s generated using surrogates generated via 50 and 500 training FDS samples. μ and σ refer to the *True* expectation and standard deviation, respectively.



(a) PDF at 200 s generated by surrogates using 50 training data samples.



(b) PDF at 200 s generated by surrogates using 500 training data samples.

Figure 13: PDFs for HRR at 200 s generated using surrogates generated via 50 and 500 training data samples. μ and σ refer to the *True* expectation and standard deviation, respectively.

Relative Error (%)	Expectation	Standard deviation	Skewness	Kurtosis
PODI	0.29	0.67	9.10	24.1
GPR	0.89	0.97	13.0	28.0
FNO	0.26	0.64	14.0	24.0
CAE	0.99	1.70	10.7	25.2

Table 5: Relative errors for burning rate moments at 200 s evaluated using surrogates trained on 500 samples.

weakest performance while PODI again leads with lowest error. As before in the case of HRRPUV, CAE is better at predicting the higher moments while weaker for first two moments. FNO shows improvement with better approximation for all moments this time. Additionally as evident from figure 17, FNO is even better than PODI for samples below 200.

In order to be also able to effectively quantify the uncertainties at the preheating zone ahead of the flame front, we analyze the moment predictions for the PMMA wall temperature (WT) in figure 18 and table 6. Similar to BR, qualitatively the results look largely similar. Weighted error $\mathcal{W}$ in the figure 19 shows again FNO to be effective for smaller training data samples and PODI for higher number of samples. GPR shows improvement possibly due to more smoothened data here in comparison to BR and HRRPUV. Again, CAE gives the lowest relative errors for higher moments in table 6.

7 Discussion

Due to the presence of highly non-linear exponential term in the Arrhenius equation, predicting pyrolysis and hence flame spread behaviour is a challenge especially using a small amount of training data. Hence, it is vital to choose a robust machine learning approach when the high-fidelity simulations are expensive.

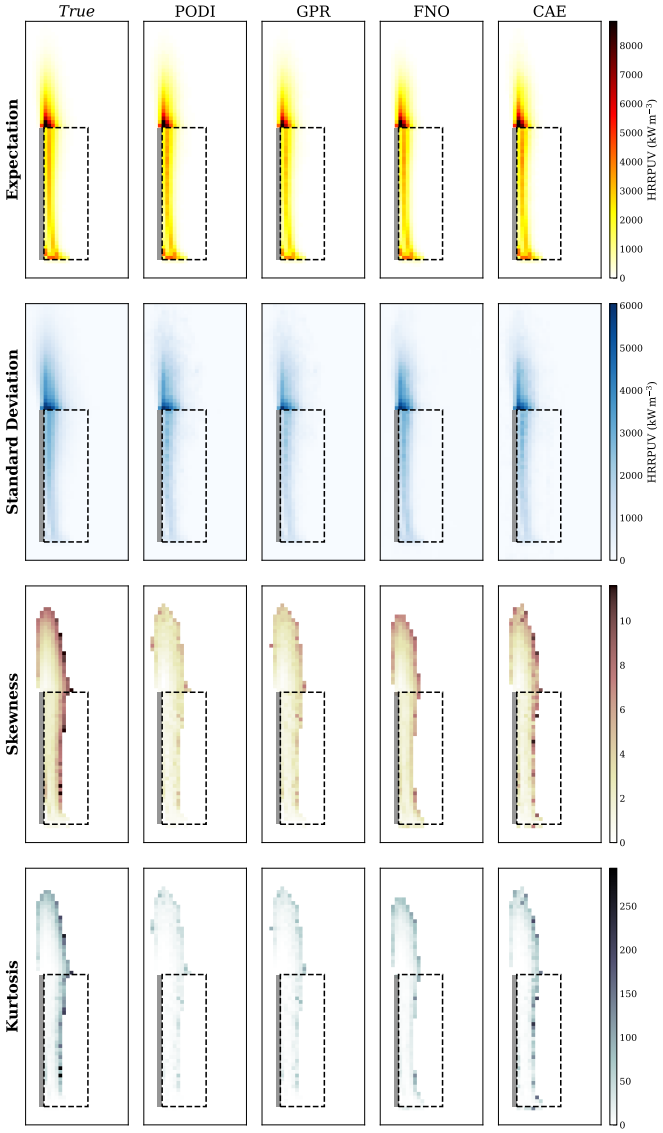


Figure 14: Moment comparison for heat release rate per unit volume (HRRPUV) using surrogates near the PMMA surface.

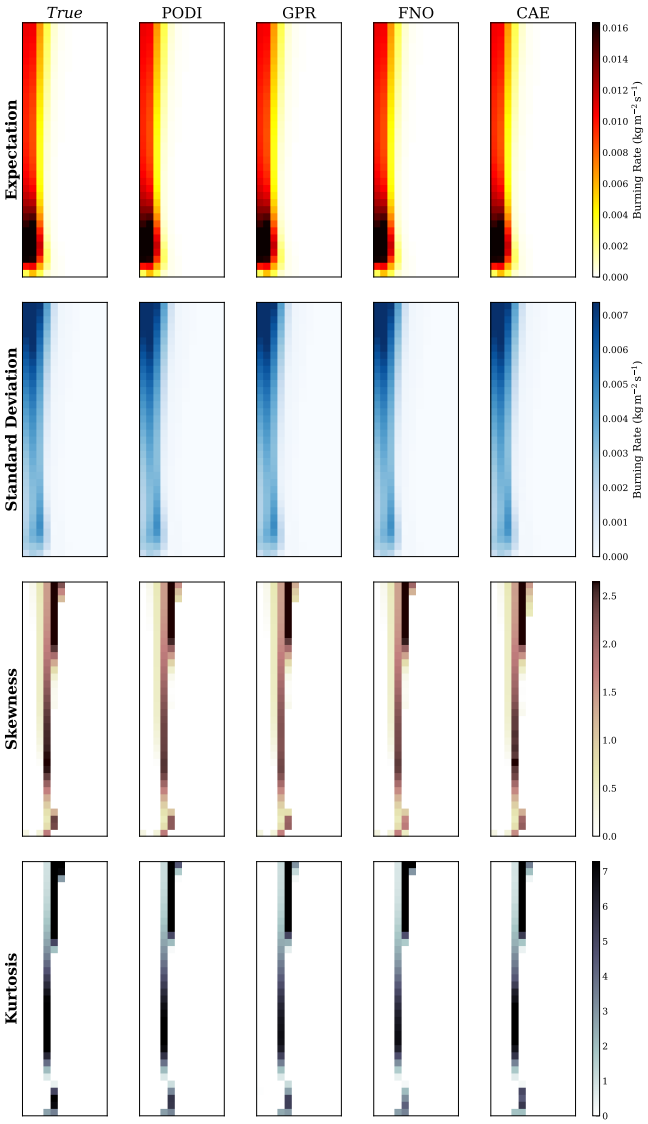


Figure 16: Moment comparison for burning rate using surrogates at the PMMA surface.

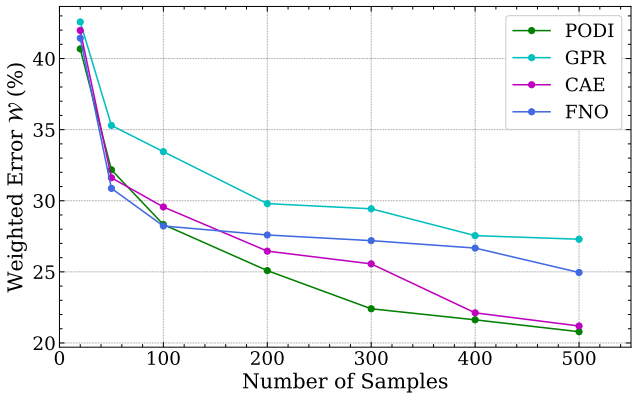


Figure 15: Weighted percentage error defined in equation (20) for heat release rate per unit volume (HRRPUV) against number of training data samples.

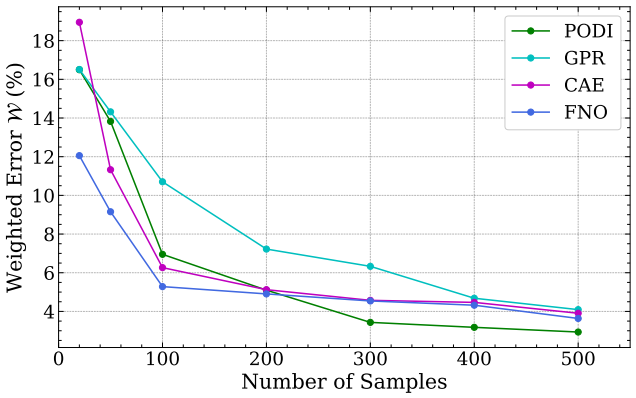


Figure 17: Weighted percentage error for burning rate moments estimation against number of training data samples.

Relative Error (%)	Expectation	Standard deviation	Skewness	Kurtosis
PODI	0.086	0.70	8.61	9.84
GPR	0.093	0.57	13.5	16.1
FNO	0.111	0.48	14.3	5.21
CAE	0.285	1.66	8.32	4.70

Table 6: Relative errors for wall temperature moments at 200 s evaluated using surrogates trained on 500 samples.

From the results above, it is evident that the predictive capability of all surrogates deteriorates with higher moments. This is well known phenomenon as observed by other researchers [78–82]. The goal nevertheless is to choose a surrogate approach that also able to capture skewness and kurtosis reasonably well. This attributes to the additional robustness of the approach compared to just simple prediction or evaluation of expectation and standard deviation.

We choose four different types of machine learning surrogates to diversify our comparison and make it broad spectrum for modelers who wish to apply data driven surrogates for flame spread related problems. Though all surrogates discussed above use the same training data, they generate different results even after HPO. This demonstrates the varying capabilities of each approach.

PODI is a reliable approach but an appropriate choice of interpolation functions is essential especially for higher input parameter dimensionality. Moreover, higher moments prediction is a weaker aspect of PODI as observed for 2D data in this work. Though this is not optimal, approaches like polynomial chaos expansion even exhibit divergence for higher moments. [24, 83]. All in all, PODI exhibits a good compromise for overall uncertainty quantification. Though it is observed to be not the most accurate surrogate for skewness and kurtosis, it is able to capture these adequately for a small amount of training data. It is evident from the results above that skewness and kurtosis are not inconsequential for UQ, nevertheless, expectation and standard deviation are still the key ingredient for comprehensive UQ. Therefore, we choose our empirical weights in descending order with moments. Modelers can alter these weights depending on the focus, but the significance of third and fourth moments shouldn't transcend that of the first two moments [84].

Though GPR generated accurate results in our previous work for a pyrolysis model [24], we see much weaker performance in the current flame spread case. This can be attributed to the higher dimensionality of the input space as pointed out by Hou and Behdinan [85].

Autoencoders appear to be more suited for skewness and kurtosis evaluation in high dimensional, nonlinear and non-smooth problems compared to the PODI and GPR. Hence we observe for 2D HRRPUV, BR and WT data that CAE is more suitable for higher

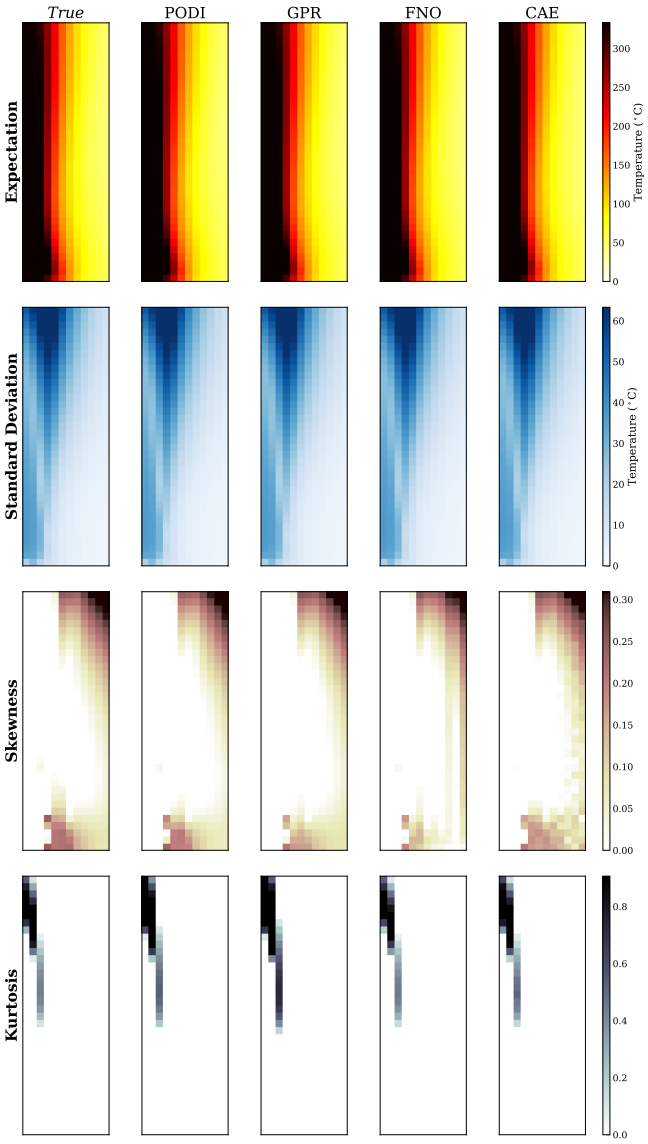


Figure 18: Moment comparison for wall temperature using surrogates at the PMMA surface.

moments than PODI.

FNO shows varying results for different outputs and training data sizes. It gives better overall performance for 100 samples and below for all outputs. For 1D HRR, it is most accurate at 500 samples while for 2D, it falls behind PODI and CAE.

The reason for this contrasting behaviour of the surrogates for moment estimation can be attributed to distinction between local and global operations intrinsic to these methods. PODI truncates global basis that represent the whole spatial or temporal domain where localized extremes require many more modes to capture. FNO also applies global truncation but in Fourier space where a localized extreme event is averaged across all Fourier modes [86]. CAE, on the other hand, employs a local to global hierarchy where early convolutional layers operate with small receptive fields [87], encoding spatially localized features before progressively integrating global spatial patterns with depth. Hence, CAE is more suited for rare events to be

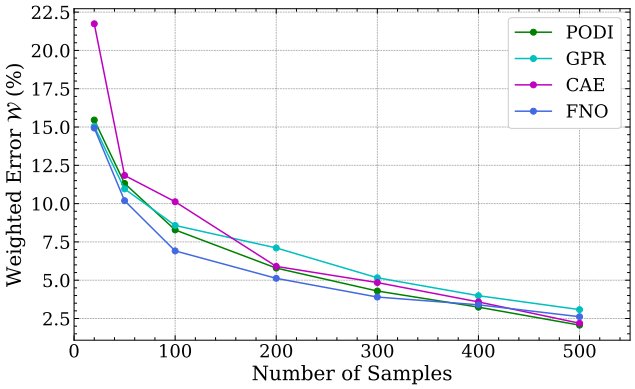


Figure 19: Weighted percentage error for wall temperature moments estimation against number of training data samples.

captured by skewness and kurtosis but comparatively less effective at estimating expectation and standard deviation.

A key downside of both FNO and autoencoder is the need for HPO, which is critical for effectiveness of deep learning methods but adds substantial computational expense. Future work could comprise of diving into better HPO strategies. For example, rather than HPO for frequency modes in FNO, there is potential for a dynamic choice of these frequency modes via incremental FNO [88]. There are also other operator learning methods available like DeepONet [89] which can also be appended with FNO as a decoder in Fourier-DeepONet [90].

Furthermore, since FNO is conceptualized in Fourier domain, all finite dimensional approximations originate from the same continuous function and share a common set of parameters. Therefore, FNO is invariant to discretizations and resolutions. Li et al. [39] showcased this for Navier-Stokes equation for a viscous, incompressible fluid where FNO trained on coarse mesh was used for fine mesh prediction. However, this can be problematic when the FNO is trained on insufficiently fine grids as systematic bias caused by aliasing errors results in poor predictions [91]. We investigated this by using 4 cm coarse mesh HRRPUV data to predict 2 cm fine mesh HRRPUV profiles. The results didn't show much promise and lead to large errors in predictions. Nevertheless, as this is not the scope of this study, we didn't investigate it further and thus is an open area for further research.

Variational autoencoder (VAE) [92] is an enhancement to CAE by regularization of the latent space. However, in our initial attempts, this didn't generate better performance than standard CAE mainly as the constraining of latent space by regularization smoothens the impact of local convolutions that are beneficial for capturing higher moments. Another alternative would be generative adversarial networks (GANs) [93].

8 Summary and Conclusions

This work aims at capturing propagated uncertainty in flame spread models using prevalent and contemporary machine learning surrogates. We compare and analyze surrogates' ability to quantify four moments for an experimental flame spread case. The comparison is done for both 1D temporal and 2D spatial uncertainty profiles. The high skewness in the output demands robust surrogate approach that is not only efficient but also is able to capture higher moments accurately.

We commence with applying input uncertainties to the five most sensitive PMMA material parameters. These are chosen based on a novel delta sensitivity analysis approach that estimates whole output probability density shift rather than change in a specific output moment apportioned to input uncertainties. The uncertainties propagated through the model are evaluated for different output parameters that include heat release rate (HRR), heat release rate per unit volume (HRRPUV), burning rate (BR) and wall temperature (WT).

We apply four types of surrogate approaches namely, projection based proper orthogonal decomposition with interpolation (PODI), probabilistic Gaussian process regression (GPR), deep learning autoencoders (MLP-AE and CAE) and operator learning based Fourier neural operator (FNO). Input-output data pairs in the range of 20 to 500 are used for surrogate training and the trained surrogates are then subsequently used for new predictions. Using a large number of surrogate predictions, statistical moments are estimated and compared against the *True* result obtained from converged 10000 full order CFD simulations.

In case of 1D HRR, all methods with the exception of GPR give highly accurate moment predictions. For 2D HRRPUV, BR and WT moments estimation, FNO is the most effective for small training data sets, whereas in general, PODI shows better performance over the whole data range analyzed. Though all methods are reasonably good is evaluation expectation and standard deviation, the difference in predictive capability becomes prominent for skewness and kurtosis. CAE is better able to predict the latter. GPR underperforms all and FNO shows varying results. In conclusion, base on empirical importance of the four moments, PODI is observed to be the most consistent and robust data-driven surrogate for prediction and uncertainty in skewness influenced full scale flame spread model as well as for milligram scale pyrolysis model analyzed in our previous work [24].

9 Data availability

The code and dataset used in this publication are publicly available at https://doi.org/10.5281/zenodo.21314513. (To be made publicly available on acceptance.)

10 Acknowledgments

The computations were carried out on the Pleiades cluster at the University of Wuppertal, which was

supported by the Deutsche Forschungsgemeinschaft (DFG) and the Bundesministerium für Bildung und Forschung (BMBF).

Author contributions

Hamza Jamil: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Fabian Brännström:** Resources, Supervision, Writing – review & editing.

Acronyms

BR	burning rate
CAE	Convolutional autoencoder
FNO	Fourier neural operator
GPR	Gaussian process regression
HPO	hyper-parameter optimization
HRR	heat release rate
HRRPUV	heat release rate per unit volume
MLP	multi layer perceptron
MMA	methyl methacrylate
MOS	method of snapshots
PDFs	probability density functions
PMMA	Polymethyl methacrylate
PODI	proper orthogonal decomposition with interpolation
RBF	radial basis functions
ROM	reduced order model
SA	sensitivity analysis
SVD	single value decomposition
UA	uncertainty analysis
UQ	uncertainty quantification
WT	wall temperature

References

[1] J. De Ris, Spread of a laminar diffusion flame, Symposium (International) on Combustion 12 (1969) 241–252.

[2] A. C. Fernandez—Pello, Flame spread modeling, Combustion Science and Technology 39 (1984) 119–134.

[3] A. C. FERNANDEZ-PELLO, T. HIRANO, Controlling mechanisms of flame spread, Combustion Science and Technology 32 (1983) 1–31.

[4] G. , I. S. W. , A. B. , Opposed-flow ignition and flame spread over melting polymers with navier-stokes gas flow, Combustion Theory and Modelling 6 (2002) 317.

[5] K. McGrattan, R. McDermott, C. Weinschenk, G. Forney, Fire dynamics simulator users guide, sixth edition, 2013. doi:10.6028/NIST.sp.1019.

[6] J. Lachaud, J. Scoggins, T. Magin, M. Meyer, N. Mansour, A generic local thermal equilibrium model for porous reactive materials submitted to high temperatures, International Journal of Heat and Mass Transfer 108 (2017) 1406–1417.

[7] C. Lautenberger, C. Fernandez-Pello, Generalized pyrolysis model for combustible solids, Fire Safety Journal 44 (2009) 819–839.

[8] S. Stoliarov, R. Lyon, Thermo-kinetic model of burning for pyrolyzing materials, Fire Safety Science 9 (2009) 1141–1152.

[9] M. M. Khan, A. Tewarson, M. Chaos, Combustion Characteristics of Materials and Generation of Fire Products, Springer New York, New York, NY, 2016, pp. 1143–1232. doi:10.1007/978-1-4939-2565-0_36.

[10] L. Valasek, J. Glasa, A note on parallel efficiency of fire simulation on cluster, Journal of Physics: Conference Series 738 (2016) 012129.

[11] C. Lautenberger, A. Fernandez-Pello, Optimization algorithms for material pyrolysis property estimation, Fire Safety Science 10 (2011) 751–764.

[12] P. Lauer, L. Arnold, F. Braennstroem, Inverse modelling of pyrolization kinetics with ensemble learning methods, Fire Safety Journal 136 (2023) 103744.

[13] Y. J. Rueda-Ordóñez, K. Tannous, E. Olivares-Gómez, An empirical model to obtain the kinetic parameters of lignocellulosic biomass pyrolysis in an independent parallel reactions scheme, Fuel Processing Technology 140 (2015) 222–230.

[14] T. Nyazika, M. Jimenez, F. Samyn, S. Bourbigot, Pyrolysis modeling, sensitivity analysis, and optimization techniques for combustible materials: A review, Journal of Fire Sciences 37 (2019) 073490411985274.

[15] J. Lachaud, Pyrolysis models and simulation tools: A cross-community comparative review highlighting open challenges, International Journal of Heat and Mass Transfer 259 (2026) 128347.

[16] F. Brännström, H. Jamil, M. Nouroozyan, J. Sailer, Challenges in Fire Spread Modeling—From Bench Scale up to Wildfire Scale, Springer Fachmedien Wiesbaden, Wiesbaden, 2025, pp. 299–319. doi:10.1007/978-3-658-47473-7_13.

[17] R. Moser, T. Oliver, Validation of Physical Models in the Presence of Uncertainty, 2017, pp. 129–156. doi:10.1007/978-3-319-12385-1_2.

[18] S. Koziel, A. Pietrenko-Dabrowska, Surrogate Modeling for High-Frequency Design, WORLD SCIENTIFIC (EUROPE), 2022. doi:doi.org/10.1142/9781800610750_0001.

[19] D. Samadian, I. B. Muhit, N. Dawood, Application of data-driven surrogate models in structural engineering: A literature review, Archives of Computational Methods in Engineering 32 (2025) 735–784.

[20] P. E. Lapenna, R. Lamioni, G. Troiani, F. Creta, Large scale effects in weakly turbulent premixed flames, Proceedings of the Combustion Institute 37 (2019) 1945–1952.

[21] D. Durox, T. Schuller, N. Noiray, S. Candel, Experimental analysis of nonlinear flame transfer functions for different flame geometries, Proceedings of the Combustion Institute 32 (2009) 1391–1398.

[22] A. González-Cencerrado, B. Peña, A. Gil, Coal flame characterization by means of digital image processing in a semi-industrial scale pf swirl burner, Applied Energy 94 (2012) 375–384.

[23] H. Jamil, F. Brännström, Intrusive and non-intrusive uncertainty quantification methodologies for pyrolysis modeling, Fire Safety Journal 143 (2024) 104060.

[24] H. Jamil, F. Brännström, Surrogate based uncertainty quantification in pyrolysis modeling: A comparative analysis, Fire Safety Journal 159 (2026) 104570.

[25] I. Bilionis, N. Zabaras, Bayesian Uncertainty Propagation Using Gaussian Processes, Springer International Publishing, Cham, 2016, pp. 1–45. doi:10.1007/978-3-319-11259-6_16-1.

[26] J. Helton, F. Davis, Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems, Reliability Engineering and System Safety 81 (2003) 23–69.

[27] Z. A. Khan, T. Hussain, F. U. M. Ullah, S. K. Gupta, M. Y. Lee, S. W. Baik, Randomly initialized cnn with densely connected stacked autoencoder for efficient fire detection, Engineering Applications of Artificial Intelligence 116 (2022) 105403.

[28] H. Wang, M. Fayaz, A. Ahmad, Y. Li, T. N. Nguyen, L. M. Dang, Masked autoencoder-based vision framework for robust fire detection in complex environments, Process Safety and Environmental Protection 203 (2025) 108019.

[29] H. Nugroho, M. Susanty, A. Irawan, M. Koyimatu, A. Yunita, Fully convolutional variational autoencoder for feature extraction of fire detection system, Jurnal Ilmu Komputer dan Informasi 13 (2020) 9–15.

[30] H. Kim, H. Jo, C. Ryu, Derivation of kinetic parameters and lignocellulosic composition from thermogram of biomass pyrolysis using convolutional neural network, International Journal of Energy Research 2024 (2024) 6184508.

[31] Y. Zhang, B. Lei, A. Mahdaviarab, X. Wang, Z. Liu, Robust biochar yield and composition prediction via uncertainty-aware resnet-based autoencoder, Biochar 7 (2025) 61.

[32] G. Li, B. Yang, S. Chang, H. Yi, Y. Zhang, Y. Wang, Gaussian process for the machine learning-based smart fire detection system, in: 2023 International Conference on Computers, Information Processing and Advanced Education (CIPAE), 2023, pp. 446–452. doi:10.1109/CIPAE60493.2023.00091.

[33] O. Dogu, A. Eschenbacher, R. John Varghese, M. Dobbelaere, D. R. D'hooge, P. H. Van Steenberge, K. M. Van Geem, Bayesian tuned kinetic monte carlo modeling of polystyrene pyrolysis: Unraveling the pathways to its monomer, dimers, and trimers formation, Chemical Engineering Journal 455 (2023) 140708.

[34] P. Viriya-amornkij, K. Kuwana, Determining kinetic parameters of cellulose and lignin pyrolysis by gaussian process regression (gpr) method, Progress in Scale Modeling, an International Journal 3 (2022) Article 5.

[35] M. E. G. R. Domingos, D. Flórez-Orrego, J. Granacher, M. Jones, M. T. dos Santos, F. Maréchal, Comparison of regression techniques for generating surrogate models to predict the thermodynamic behavior of biomass gasification systems, in: 33rd European Symposium on Computer Aided Process Engineering, volume 52 of *Computer Aided Chemical Engineering*, Elsevier, 2023, pp. 1235–1240. doi:10.1016/B978-0-443-15274-0.50197-9.

[36] C. Zhong, S. Cheng, M. Kasoar, R. Arcucci, Reduced-order digital twin and latent data assimilation for global wildfire prediction, Natural Hazards and Earth System Sciences 23 (2023) 1755–1768.

[37] F. Black, P. Schulze, B. Unger, Efficient wildland fire simulation via nonlinear model order reduction, Fluids 6 (2021).

[38] G. Elisa, A. Sciacovelli, V. Vittorio, D. Ascoli, Faster prediction of wildfire behaviour by physical models through application of proper orthogonal decomposition, International Journal of Wildland Fire 25 (2016).

[39] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, in: International Conference on Learning Representations (ICLR), 2021.

[40] F. Lehmann, F. Gatti, M. Bertin, D. Clouteau, 3d elastic wave propagation with a factorized fourier neural operator (f-fno), Computer Methods in Applied Mechanics and Engineering 420 (2024) 116718.

[41] M. Zhong, Z. Yan, Data-driven soliton mappings for integrable fractional nonlinear wave equations

via deep learning with fourier neural operator, Chaos, Solitons and Fractals 165 (2022) 112787.

[42] L. He, F. Chen, Y. Qin, A novel data-driven method for predicting heat flux of hypersonic aircraft based on fourier neural operator, Aerospace Science and Technology 169 (2026) 111397.

[43] W. Johnny, H. Brigido, M. Ladeira, J. C. F. Souza, Fourier neural operator for image classification, in: 2022 17th Iberian Conference on Information Systems and Technologies (CISTI), 2022, pp. 1–6. doi:10.23919/CISTI54924.2022.9820128.

[44] A. Saltelli, K. Aleksankina, W. Becker, P. Fennell, F. Ferretti, N. Holst, S. Li, Q. Wu, Why So Many Published Sensitivity Analyses Are False. A Systematic Review of Sensitivity Analysis Practices, arXiv e-prints (2017) arXiv:1711.11359.

[45] I. T. Leventon, B. Batiot, M. C. Bruns, S. Hostikka, Y. Nakamura, P. Reszka, T. Rogaume, S. I. Stoliarov, The MaCFP Condensed Phase Working Group: A Structured, Global Effort towards Pyrolysis Model Development, in: Obtaining Data for Fire Growth Models, ASTM International, 2023. doi:10.1520/STP164220210111.

[46] D. M. Chaudhari, G. J. Fiola, S. I. Stoliarov, Experimental analysis and modeling of buoyancy-driven flame spread on cast poly(methyl methacrylate) in corner configuration, Polymer Degradation and Stability 183 (2021) 109433.

[47] G. J. Fiola, D. M. Chaudhari, S. I. Stoliarov, Comparison of pyrolysis properties of extruded and cast poly(methyl methacrylate), Fire Safety Journal 120 (2021) 103083. Fire Safety Science: Proceedings of the 13th International Symposium.

[48] K. McGrattan, S. Hostikka, R. McDermott, J. Floyd, C. Weinschenk, K. Overholt, Fire Dynamics Simulator Technical Reference Guide, Volume 3: Validation, Technical Report NIST Special Publication 1018-3, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, 2023. doi:10.6028/NIST.SP.1018-3.

[49] A. Saltelli, S. Tarantola, F. Campolongo, M. Ratto, Sensitivity analysis in practice: A guide to assessing scientific models (2004).

[50] I. Sobol', On the distribution of points in a cube and the approximate evaluation of integrals, USSR Computational Mathematics and Mathematical Physics 7 (1967) 86–112.

[51] E. Borgonovo, A new uncertainty importance measure, Reliability Engineering and System Safety 92 (2007) 771–784.

[52] E. Borgonovo, W. Castaings, S. Tarantola, Model emulation and moment-independent sensitivity analysis: An application to environmental modelling, Environmental Modelling and Software 34 (2012) 105–115. Emulation techniques for the reduction and sensitivity analysis of complex environmental models.

[53] T. L. Quaresma, T. Hehnen, L. Arnold, Sensitivity analysis for an effective transfer of estimated material properties from cone calorimeter to horizontal flame spread simulations, Fire Safety Journal 144 (2024) 104116.

[54] M. Gavish, D. L. Donoho, The optimal hard threshold for singular values is $4/\sqrt{3}$, IEEE Transactions on Information Theory 60 (2014) 5040–5053.

[55] L. Sirovich, Turbulence and the dynamics of coherent structures. i - coherent structures. ii - symmetries and transformations. iii - dynamics and scaling, Quarterly of Applied Mathematics - QUART APPL MATH 45 (1987).

[56] M. D. Buhmann, C. A. Micchelli, On radial basis approximation on periodic grids, Mathematical Proceedings of the Cambridge Philosophical Society 112 (1992) 317–334.

[57] Y. Wang, H. Yao, S. Zhao, Auto-encoder based dimensionality reduction, Neurocomputing 184 (2016) 232–242. RoLoD: Robust Local Descriptors for Computer Vision 2014.

[58] F. Murtagh, Multilayer perceptrons for classification and regression, Neurocomputing 2 (1991) 183–197.

[59] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: A comprehensive review, Neural Computation 29 (2017) 2352–2449.

[60] S. Cheng, Y. Jin, S. P. Harrison, C. Quilodrán-Casas, I. C. Prentice, Y.-K. Guo, R. Arcucci, Parameter flexible wildfire prediction using machine learning techniques: Forward and inverse modelling, Remote Sensing 14 (2022).

[61] H. Sedighi, A. Abbaskhah, P. Akbarzadeh, H. Marvi, An approach to flow field prediction over the horizontal axis wind turbine by passive flow control device using CNN-Autoencoder model, Scientific Reports 15 (2025) 33305.

[62] C. E. Rasmussen, C. K. I. Williams, Gaussian Processes for Machine Learning, The MIT Press, 2005. doi:10.7551/mitpress/3206.001.0001.

[63] I. Bilionis, N. Zabaras, Bayesian Uncertainty Propagation Using Gaussian Processes, 2015, pp. 1–45. doi:10.1007/978-3-319-11259-6_16-1.

[64] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Multipole graph neural operator for parametric partial differential equations, 2020. arXiv:2006.09535.

[65] V. Duruisseaux, J. Kossaifi, A. Anandkumar, Fourier neural operators explained: A practical perspective, 2025. doi:10.48550/arXiv.2512.01421.

[66] N. Kovachki, S. Lanthaler, S. Mishra, On universal approximation and error bounds for fourier neural operators, Journal of Machine Learning Research 22 (2021) 1–76.

[67] B. M. Adams, W. J. Bohnhoff, K. R. Dalbey, et al., Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.15 user's manual (2021).

[68] J. Feinberg, H. P. Langtangen, Chaospy: An open source tool for designing methods of uncertainty quantification, Journal of Computational Science 11 (2015).

[69] J. Herman, W. Usher, SALib: An open-source python library for sensitivity analysis, The Journal of Open Source Software 2 (2017).

[70] N. Demo, M. Tezzele, G. Rozza, EZyRB: Easy Reduced Basis method, The Journal of Open Source Software 3 (2018) 661.

[71] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[72] F. Chollet, et al., Keras, https://keras.io, 2015.

[73] J. Kossaifi, N. Kovachki, Z. Li, D. Pitt, M. Liu-Schiaffini, R. J. George, B. Bonev, K. Azizzadenesheli, J. Berner, V. Duruisseaux, A. Anandkumar, A library for learning neural operators, 2026. URL: https://arxiv.org/abs/2412.10354. arXiv:2412.10354.

[74] P. Virtanen, R. Gommers, T. E. Oliphant, et al., SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nature Methods 17 (2020) 261–272.

[75] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J. E. Gonzalez, I. Stoica, Tune: A research platform for distributed model selection and training, arXiv preprint arXiv:1807.05118 (2018).

[76] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, A. Talwalkar, A system for massively parallel hyperparameter tuning, 2020. URL: https://arxiv.org/abs/1810.05934. arXiv:1810.05934.

[77] M. Rosenblatt, Remarks on a Multivariate Transformation, The Annals of Mathematical Statistics 23 (1952) 470 – 472.

[78] D. Tartakovsky, P. Gremaud, Method of Distributions for Uncertainty Quantification, 2017, pp. 763–783. doi:10.1007/978-3-319-12385-1_27.

[79] Y. Marzouk, T. Moselhy, M. Parno, A. Spantini, Sampling via Measure Transport: An Introduction, 2016, pp. 1–41. doi:10.1007/978-3-319-11259-6_23-1.

[80] Y. Xu, M. Netto, L. Mili, Propagating parameter uncertainty in power system nonlinear dynamic simulations using a koopman operator-based surrogate model, IEEE Transactions on Power Systems 37 (2022) 3157–3160.

[81] J. Bai, S. Ng, Tests for skewness, kurtosis, and normality for time series data, Journal of Business and Economic Statistics 23 (2005) 49–60.

[82] D. Zhang, J. Jia, Z. Han, H. Ouyang, J. Liu, X. Han, An efficient uncertainty quantification and propagation method through skewness and kurtosis fitting region, Structural and Multidisciplinary Optimization 66 (2023) 36.

[83] S. Rahman, Higher-order moments of spline chaos expansion, Probabilistic Engineering Mechanics 77 (2024) 103666.

[84] P. Schuster, Distributions, Moments, and Statistics, Springer International Publishing, Cham, 2016, pp. 83–197. doi:10.1007/978-3-319-39502-9_2.

[85] C. K. J. Hou, K. Behdinan, Dimensionality reduction in surrogate modeling: A review of combined methods, Data Science and Engineering 7 (2022) 402–427.

[86] A. I. Khrabry, E. A. Startsev, A. T. Powis, I. D. Kaganovich, Deep wave network for modeling multi-scale physical dynamics, 2026. URL: https://arxiv.org/abs/2605.04198. arXiv:2605.04198.

[87] J. Heaton, Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618, Genetic Programming and Evolvable Machines 19 (2017).

[88] R. J. George, J. Zhao, J. Kossaifi, Z. Li, A. Anandkumar, Incremental spatial and spectral learning of neural operators for solving large-scale pdes, 2024. URL: https://arxiv.org/abs/2211.15188. arXiv:2211.15188.

[89] L. Lu, P. Jin, G. E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nature Machine Intelligence 3 (2021) 218–229.

[90] M. Zhu, S. Feng, Y. Lin, L. Lu, Fourier-deeponet: Fourier-enhanced deep operator networks for full waveform inversion with improved accuracy, generalizability, and robustness, Computer Methods in Applied Mechanics and Engineering 416 (2023) 116300.

[91] V. S. Fanaskov, I. V. Oseledets, Spectral neural operators, Doklady Mathematics 108 (2023) S226–S232.

[92] D. P. Kingma, M. Welling, Auto-encoding variational bayes, 2022. URL: https://arxiv.org/abs/1312.6114. arXiv:1312.6114.

[93] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, Y. Zheng, Recent progress on generative adversarial networks (gans): A survey, IEEE Access 7 (2019) 36322–36333.