Data Assimilation using Heteroscedastic Bayesian Neural Network Ensembles for Reduced-Order Flame Models

Maximilian L. Croci[⋆†], Ushnish Sengupta[⋆], and Matthew P. Juniper

Department of Engineering
University of Cambridge
Cambridge, United Kingdom

Abstract. The parameters of a level-set flame model are inferred using an ensemble of heteroscedastic Bayesian neural networks (BayNNEs). The neural networks are trained on a library of 1.7 million observations of 8500 simulations of the flame edge, obtained using the model with known parameters. The ensemble produces samples from the posterior probability distribution of the parameters, conditioned on the observations, as well as estimates of the uncertainties in the parameters. The predicted parameters and uncertainties are compared to those inferred using an ensemble Kalman filter. The expected parameter values inferred with the BayNNE method, once trained, match those inferred with the Kalman filter but require less than one millionth of the time and computational cost of the Kalman filter. This method enables a physics-based model to be tuned from experimental images in real time.

Keywords: Bayesian inference · Deep learning · Thermoacoustics · Data assimilation.

1 Introduction

1.1 Thermoacoustics

The prediction and control of thermoacoustic instability is a persistent challenge in jet and rocket engine design [1]. In gas turbines, the drive towards lower NO_x emissions has led to the use of lean premixed combustion, which is particularly susceptible to thermoacoustic instabilities [2]. Thermoacoustic instability is caused by the heat release rate and the pressure being in phase during combustion [3]. Heat release rate fluctuations are caused by flame surface area fluctuations, which in turn are caused by velocity perturbations and flame dynamics [4–7]. Any physics-based model must therefore contain the flame's response to velocity perturbations. This response can be calculated using detailed CFD simulations of the flame. However, these CFD simulations are expensive.

⋆ Equal contribution.
† Corresponding author: mlc70@cam.ac.uk.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

In this paper we use data to tune the parameters of physics-based reduced-order models, in order to reduce the cost while retaining as much accuracy as possible.

The simplest physics-based model sets the heat release rate fluctuation to be a linear multiple of the velocity perturbation at the base of the flame some time earlier. This delay models the time taken for perturbations to travel down the flame. This is known as the $n - \tau$ model [8]. It is too simple for our purposes because it cannot simulate the flame dynamics. In this paper we model the flame as the zero contour of a continuous function that advects with the flow. This is known as the G-equation model [9]. This allows the flame dynamics to be simulated cheaply but the parameters of this model need to be assimilated from experimental data in order to render the model quantitatively accurate. The ensemble Kalman filter [10] (EnKF) has been used previously to assimilate data into the G-equation model [11, 12]. The EnKF performs Bayesian inference to infer the variables (the state and parameters) of the G-equation model by statistically combining model forecasts with measurements of the variables. The EnKF in principle can be used online: Bayesian inference is performed whenever measurements become available. When used for data assimilation of our experiments of a conical Bunsen flame, however, the computational requirements of the EnKF render online Bayesian inference impossible: measurements are available every $O(10^{-3})$ seconds while forecasting between data assimilation steps takes $O(10^{1})$ seconds. This study proposes an alternative method for practical online assimilation of data into the G-equation model of the Bunsen flame.

1.2 Bayesian deep learning

Bayesian deep learning refers to the use of deep learning algorithms, such as deep neural networks (NNs) and deep Gaussian processes (GPs), for Bayesian inference [14, 15]. Bayesian NNs [16] replace the point estimates of each of the weights and biases with Gaussian probability distributions, with means and variances learned during training. The outputs can then be inferred from the inputs and the distribution of every weight and bias in the NN. Unfortunately, Bayesian NNs of practical size are too expensive to train [17]. More recently, ensembles of deep NNs have been used to perform approximate Bayesian inference [18–20], with the approximation improving with increasing width of the NN's hidden layers. These Bayesian NN ensembles (BayNNEs) learn the mean and variance of the posterior distribution of the outputs given the inputs. When multiple outputs are to be inferred, heteroscedastic BayNNEs learn the means and variances of each output, without assuming a common variance for all outputs. This study uses heteroscedastic BayNNEs to infer the parameters of the G-equation model given experimental observations.

2 Methods

2.1 Bunsen flame experiment

Figure 1 shows the Bunsen experiment setup: a Bunsen burner is placed inside a transparent duct and images of the flame are taken with a high-speed camera at

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

$f_s = 2500$ frames per second and a resolution of 1200×800 pixels. The flame is forced acoustically with a speaker from 250 Hz to 450 Hz. The gas composition (methane, ethene and air) and flow rate are varied using mass flow controllers. By varying the forcing frequency and amplitude and gas composition and flow rate, flames with different aspect ratios, propagation speeds and degrees of cusping of the flame edge are observed. In some cases, the flame edge cusping leads to pinch-off at the flame tip. For each of the 270 different flame operating conditions, 500 images are taken.

The flame images are processed and the flame edge is extracted as a radial location x, which is a singularly-valued function of the axial co-ordinate y: $x = f(y)$. First, the pixel intensities are thresholded and the flame location x for every vertical co-ordinate y is found by weighted interpolation of the thresholded pixels, where the weights are the pixel intensities. Next, splines with 28 knots are used to smooth $x(y)$. Each flame image is therefore converted into a 90×1 vector of flame edge x locations $\mathbf{x}$. The y co-ordinates are the same for all flames, so are discarded. Observation vectors $\mathbf{z}$ are created by stacking 10 consecutive $\mathbf{x}$ vectors. These observation vectors are the inputs to the neural networks. All 500 images of each Bunsen flame are processed in this way.

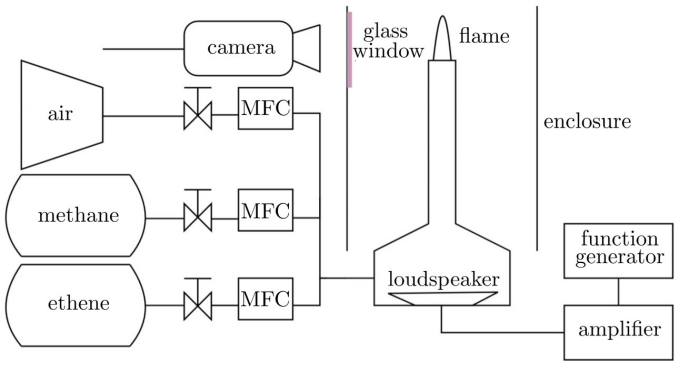

Fig. 1. Diagram of the Bunsen flame experiment setup: a Bunsen burner is placed inside a tube and a high-speed camera takes images of the flame through a glass window. The fuel composition (air, methane and ethene) and flow rate are controlled using mass flow controllers (MFCs). The flame is forced acoustically using a loudspeaker.

2.2 Flame edge model

In this paper the flame edge is defined to be the $G = 0$ contour (or level-set) of a scalar field $G(x, y, t)$. Regions of negative and positive G correspond to unburnt and burnt gases respectively (the magnitude of G has no physical significance). The evolution of G is governed by:

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

$$\frac{\partial G}{\partial t} + \mathbf{v} \cdot \nabla G = s_L |\nabla G|, \tag{1}$$

where $\mathbf{v}$ is a prescribed velocity field and s_L is the laminar flame speed: the speed at which the flame edge propagates normal to itself into the reactants. The flame speed s_L is a function of the unstretched (adiabatic) flame speed s_L^0, the flame curvature κ and the Markstein length $\mathcal{L}$, and is insensitive to pressure variations:

$$s_L = s_L^0 \left(1 - \mathcal{L}\kappa\right). \tag{2}$$

The unstretched flame speed s_L^0 depends only on the flame chemistry. The velocity field $\mathbf{v}$ comprises a parabolic base flow profile $V(x)$ and superimposed continuity-obeying velocity perturbations $u'(x, y, t)$ and $v'(x, y, t)$:

$$\mathbf{v} = (V(x) + v')\,\mathbf{j} + u'\mathbf{i}, \tag{3}$$

$$\frac{V(x)}{V} = 1 + \alpha\left(1 - 2\left(\frac{x}{R}\right)^2\right), \tag{4}$$

$$\frac{v'(x, y, t)}{V} = \epsilon \sin\left(\mathrm{St}\left(\frac{Ky}{R} - t\right)\right), \tag{5}$$

$$\frac{u'(x, y, t)}{V} = -\frac{\epsilon K \mathrm{St} x}{\beta R} \cos\left(\mathrm{St}\left(\frac{Ky}{R} - t\right)\right), \tag{6}$$

where α determines the shape of the base flow profile ($\alpha = 0$ is uniform flow, $\alpha = 1$ is Poiseuille flow), ϵ is the amplitude of the vertical velocity perturbation with phase speed V/K, $\mathrm{St} = 2\pi f R\beta/V$ is the Strouhal number with forcing frequency f and flame radius R, and β is the aspect ratio of the unperturbed flame. The parameters $K, \epsilon, \mathcal{L}, \alpha, \mathrm{St}$ and β are tuned to fit an observed flame shape. Figure 2 shows a diagram of the flame edge under the G-equation model. This model allows cusps to form at the flame edge and pockets of unburnt reactants to detach from the flame tip, as is observed in some experiments. It has proven to be a versatile flame edge model in several previous studies, despite having only a few parameters [21].

2.3 Forced cycle library

A library of flame edge locations is created with the flame edge model at known parameter values $K, \epsilon, \mathcal{L}, \alpha, \mathrm{St}, \beta$ and f/f_s in the same format as the observation vectors, $\mathbf{z}$. The parameter values are sampled using quasi-Monte Carlo sampling to ensure good coverage of the parameter space. The parameters are sampled from the following ranges: $0.0 < K \leq 1.5$, $0.0 < \epsilon \leq 1.0$, $0.02 \leq \mathcal{L} \leq 0.08$, $0.0 \leq \alpha \leq 1.0$, $2.0 \leq \beta \leq 10.0$ and $0.08 \leq f/f_s \leq 0.20$. The values of St are calculated by additionally sampling $0.002 \leq R \leq 0.004$ m and $1 \leq V \leq 5$ m/s and calculating $\mathrm{St} = 2\pi f R\beta/V$. The parameters are sampled 8500 times, normalised to between 0 and 1 and recorded in target vectors $\{\mathbf{t} = [K, \epsilon, \mathcal{L}, \alpha, \mathrm{St}, \beta]\}$.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

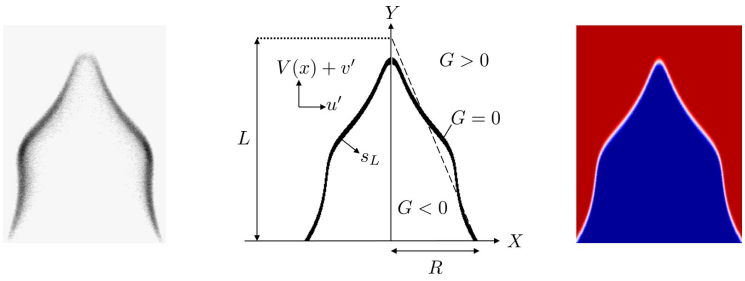

Fig. 2. *Left*: An image of a Bunsen flame. *Middle*: In the G-equation model, the flame edge is represented by the $G = 0$ contour (or level-set) of a continuous scalar field $G(x, y, t)$. Unburnt and burnt gases are regions where $G < 0$ and $G > 0$ respectively. The flame edge travels normal to itself into the unburnt gases with speed s_L. The flame edge advects under the prescribed velocity field, which comprises continuity-obeying velocity perturbations $u'(x, y, t)$ and $v'(x, y, t)$ superimposed onto a steady base flow profile $V(x)$. *Right*: A G field solution in LSGEN2D, a level-set solver. Blue and red regions are unburnt and burnt gases respectively, and the thin white band represents the flame edge.

For each of the 8500 parameter configurations, LSGEN2D [22] iterates the G field (1) until the solution is periodic and then stores 200 snapshots from one period of the forced cycle. For each snapshot the flame edge is found by interpolating G to find the contour $G = 0$. The same procedure as for the experimental images is then followed to find $x = f(y)$. Observation vectors $\mathbf{z}$ are created by stacking 10 consecutive $\mathbf{x}$ vectors. There are 200 observation vectors created from every cycle, resulting in a library of 1.7×10^6 observation-target parameter pairs $\{(\mathbf{z}, \mathbf{t})\}$. The neural networks are trained to recognise the parameter values from the observation vectors, $\mathbf{z}$.

2.4 Inference using heteroscedastic Bayesian Neural Network ensembles

We assume that the posterior probability distribution of the parameters, given the observations, can be modelled by a neural network, $p_\theta(\mathbf{t}|\mathbf{z})$, with its own parameters θ. We assume that this posterior distribution has the form:

$$p_\theta(\mathbf{t}|\mathbf{z}) = \mathcal{N}\left(\boldsymbol{\mu}(\mathbf{z}), \boldsymbol{\Sigma}(\mathbf{z})\right), \tag{7}$$

where $\boldsymbol{\Sigma}(\mathbf{z})$, the posterior covariance matrix of the parameters given the data, is diagonal with $\boldsymbol{\sigma}^2(\mathbf{z})$ on its diagonal. This enforces our assumption that the parameters are mutually independent, given the observations $\mathbf{z}$. We use an ensemble of $M = 20$ neural networks. The architecture of each neural network is shown in Figure 3. Each neural network comprises an input layer, four hidden layers with ReLU activations and two output layers: one for the mean vector $\boldsymbol{\mu}(\mathbf{z})$ and one for the variance vector $\boldsymbol{\sigma}^2(\mathbf{z})$. The output layer for the mean uses a sigmoid activation to restrict outputs to the range $(0, 1)$. The output layer

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

for the variance uses an exponential activation to ensure positivity. Each neural network in the ensemble is initialised with unique weights $\boldsymbol{\theta}_{j,anc}$ sampled from a Gaussian prior distribution $\mathcal{N}(0, \frac{1}{N_H})$ and biases $\mathbf{b}_{j,anc}$ sampled from a uniform prior distribution in the range $[-\frac{1}{\sqrt{N_H}}, \frac{1}{\sqrt{N_H}}]$.

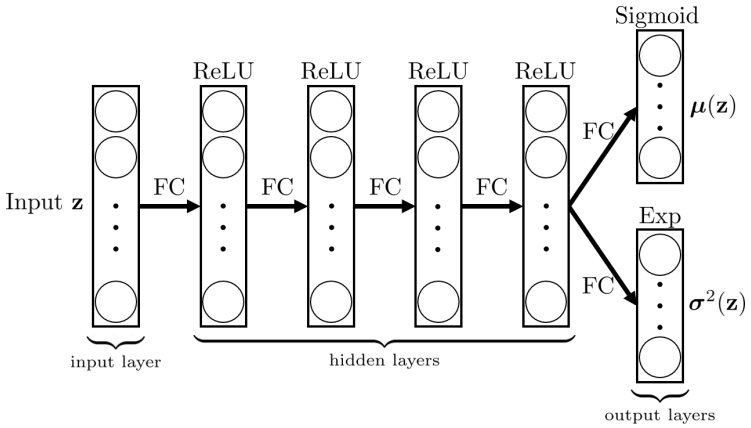

Fig. 3. Architecture of each neural network in the ensemble of 20. The input and hidden layers have 900 nodes each, while each output layer has 6 nodes each. All layers are fully connected (FC). Rectified Linear Unit (ReLU) activation functions are used for the hidden layers and sigmoid and exponential (Exp) activation functions are used for the mean and variance output layers respectively.

For a single observation $\mathbf{z}$, the j-th neural network in the ensemble produces a mean and variance estimate of the G-equation parameters:

$$\boldsymbol{\mu}_j(\mathbf{z}), \boldsymbol{\sigma}_j^2(\mathbf{z}). \tag{8}$$

This is achieved by minimising the loss function $\mathfrak{L}_j$:

$$\begin{aligned}\mathfrak{L}_j = (\boldsymbol{\mu}_j(\mathbf{z}) - \mathbf{t})^T \boldsymbol{\Sigma}_j(\mathbf{z})^{-1} (\boldsymbol{\mu}_j(\mathbf{z}) - \mathbf{t}) + \log(|\boldsymbol{\Sigma}_j(\mathbf{z})|) \\ + (\boldsymbol{\theta}_j - \boldsymbol{\theta}_{anc,j})^T \boldsymbol{\Sigma}_{prior}^{-1} (\boldsymbol{\theta}_j - \boldsymbol{\theta}_{anc,j}).\end{aligned} \tag{9}$$

The first two terms of the loss function are the negative logarithm of the normalised Gaussian likelihood function up to an additive constant. The third term is a regularising term that penalises deviation from prior anchor values $\boldsymbol{\theta}_{anc,j}$. The NNs produce samples from the posterior distribution. This is called randomised maximum a-posteriori (MAP) sampling [18].

For a single observation vector $\mathbf{z}$, the prediction from the ensemble of neural networks is therefore a distribution of M Gaussians, each centred at their respective means $\boldsymbol{\mu}_j(\mathbf{z})$. Following similar treatment in [23], this distribution is then approximated by a single multivariate Gaussian posterior distribution

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

$p(\mathbf{t}|\mathbf{z}) \approx \mathcal{N}(\boldsymbol{\mu}(\mathbf{z}), \boldsymbol{\Sigma}(\mathbf{z}))$ with mean and variance:

$$\boldsymbol{\mu}(\mathbf{z}) = \frac{\Sigma_j \boldsymbol{\mu}_j(\mathbf{z})}{M}, \quad \boldsymbol{\Sigma}(\mathbf{z}) = \operatorname{diag}\left(\boldsymbol{\sigma}^2(\mathbf{z})\right), \tag{10}$$

$$\boldsymbol{\sigma}^2(\mathbf{z}) = \frac{\Sigma_j \boldsymbol{\sigma}_j^2(\mathbf{z})}{M} + \frac{\Sigma_j \boldsymbol{\mu}_j^2(\mathbf{z})}{M} - \left(\frac{\Sigma_j \boldsymbol{\mu}_j(\mathbf{z})}{M}\right)^2. \tag{11}$$

This is repeated for every observation vector $\mathbf{z}$. The posterior distribution $p(\mathbf{t}|\mathbf{z}_i)$ with the smallest total variance $\sigma^2_{i,\text{tot}} = ||\boldsymbol{\sigma}^2(\mathbf{z}_i)||_1$ is chosen as the best guess to the true posterior. The M parameter samples from the chosen posterior are used for re-simulation, which allows us to check the predicted flame shapes and to calculate the normalised area variation over one cycle.

2.5 Inference using the ensemble Kalman filter

The Kalman filter iteratively performs Bayesian inference to find the probability distribution of the state of a system given noisy observations of the system and an imperfect model of the system dynamics. In this study, the state comprises the location of the flame edge and the parameters K and ϵ. These parameters are assumed to be independent given the observations of the flame edge and constant for each of the Bunsen flame experiments. The flame edge is modelled using the G-equation (1). The ensemble Kalman filter [10] (EnKF) evolves an ensemble of simulations forward in time. The covariance matrix of these ensembles is assumed to approximate the covariance matrix of the state evolved over the same period of time. The EnKF is more practical when the state contains many variables and the evolution is nonlinear. In this study, the state contains $O(10^2)$ variables and the governing equation (1) is nonlinear.

The parameters $\mathcal{L}, \alpha, \beta$ and St are calculated by solving the G-equation (1) when steady and do not need to be inferred with the EnKF. This reduces the cost of the EnKF but increases the number of steps compared with the BayNNE method. The forcing frequency f is manually set when running the Bunsen flame experiments. The unperturbed laminar flame speed s_L^0 is calculated using Cantera[1] and knowledge of the methane and ethene flow rates. V is calculated from s_L^0 and β: $V = s_L^0\sqrt{\beta^2 - 1}$. The Strouhal number can then be calculated: $\text{St} = 2\pi f \beta L/V$. Ref. [12] contains details about the implementation of the EnKF.

An ensemble size of 32 is used in this study. A multiplicative inflation factor of 1% is chosen to mitigate the underestimation of the error covariances due to the finite ensemble size [24]. Once the EnKF has converged, the parameters K and ϵ calculated by each ensemble member are recorded. These are samples from the posterior distribution of the parameters given all the flame x-location vectors: $p(K, \epsilon|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$. This differs from the BayNNE, which was given $\mathbf{x}$-location vectors in groups of 10.

[1] Cantera is a suite of tools for problems involving chemical kinetics, thermodynamics, and transport processes [13].

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

3 Results and Discussion

The ensemble of 20 Bayesian neural networks is trained on the forced cycle library for 5000 epochs, with a 80 : 20 train-test split and an Adam optimiser with learning rate 10^{-3}. Training takes approximately 12 hours per neural network on an NVIDIA P100 GPU. The ensemble is then evaluated on the observations of every Bunsen flame and the estimate of the parameters with the lowest total uncertainty is selected for re-simulation. The evaluation takes $O(10^{-3})$ seconds on an Intel Core i7 processor on a laptop.

For each Bunsen flame test case, the ensemble Kalman filter technique requires an hour to calculate estimates of $\mathcal{L}, \alpha, St$ and β on an Intel Core i7 processor on a laptop, followed by 2 hours of data assimilation on a pair of Intel Xeon Skylake 6142 processors[2] to produce estimates of K and ϵ.

Figures 4 and 5 show the results of inference on two different Bunsen flames. Both techniques produce good parameter estimates, in that the predicted flame shapes are in good agreement with the experiments. Furthermore, the BayNNE predicts normalised area variation curves at least as accurate as those predicted by the EnKF. However, the EnKF's predictions of K and ϵ are more confident than the BayNNE's predictions of all 6 parameters. The difference between the predicted uncertainties of the EnKF and BayNNE can be explained by the difference between the posterior distributions calculated by both techniques. The EnKF calculates $p(K, \epsilon|\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{500})$: the probability distribution of the parameters K and ϵ given the location vectors $\mathbf{x}$ from all 500 images. For the BayNNE technique, the posterior $p(\mathbf{t}|\mathbf{z}_i) = p(\mathbf{t}|\mathbf{x}_i, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_{i+9})$ with the smallest total variance is chosen. This is the probability distribution of the parameters $\mathbf{t}$ given only 10 image location vectors, as opposed to the 500 used by the EnKF. Therefore, it is expected that the BayNNE technique produces more uncertain parameter estimates.

This raises the question as to whether it is possible to increase the certainty of the parameters by providing the BayNNE with more data. It is not possible to infer a posterior $p(\mathbf{t}|\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N)$ from the individual posteriors $p(\mathbf{t}|\mathbf{z}_i)$ without knowledge of the dependence between any two observations, $p(\mathbf{z}_i|\mathbf{z}_j)$. Two observation vectors are not independent, because the information gained from a first observation restricts the expected subsequent observations to a likely set of forced cycle states. One solution is to increase the number of location vectors $\mathbf{x}$ in each observation vector, which increases the computational cost of training the neural networks. Another solution is a recurrent neural network, which can have variable length sequences of data as inputs. Future work will focus on the development of a Bayesian recurrent neural network solution to this problem.

To summarise, once the BayNNE has been trained on the forced cycle library it can be used to reliably infer all 6 parameters of the flame edge model based on 10 consecutive snapshots of the Bunsen flame experiments. If more snapshots are provided, this method finds the sequence of 10 snapshots that minimises

[2] The EnKF is fully parallelised: the processors have 32 cores in total, one for each member in the ensemble.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

the uncertainty in the parameters. This differs from the EnKF which requires hours to infer the parameters of each Bunsen flame. The BayNNE technique therefore provides similarly accurate parameter estimates at a fraction of the computational cost.

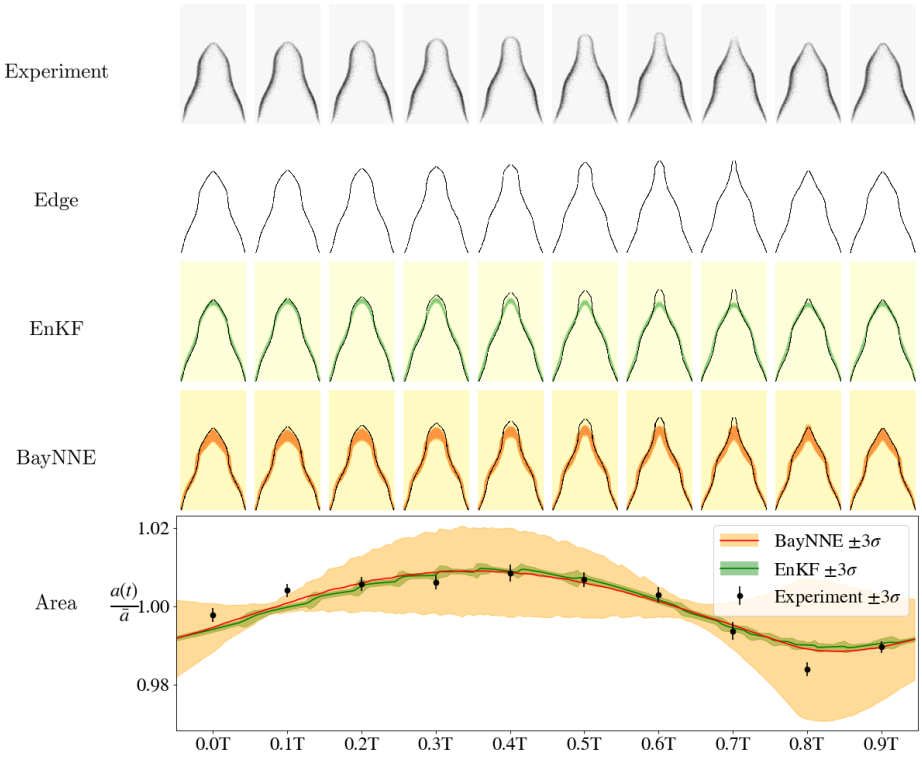

Fig. 4. Results of inference on a flame with flow rates of ethene 0.20 NL/min, methane 0.40 NL/min, air 4.50 NL/min and forcing frequency 450 Hz. Top row: Bunsen flame whose parameters are to be estimated. Second row: the detected flame edge. Third row: re-simulated flames using EnKF estimated parameters. Fourth row: re-simulated flames using BayNNE estimated parameters. Bottom: normalised surface area variations over one period.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

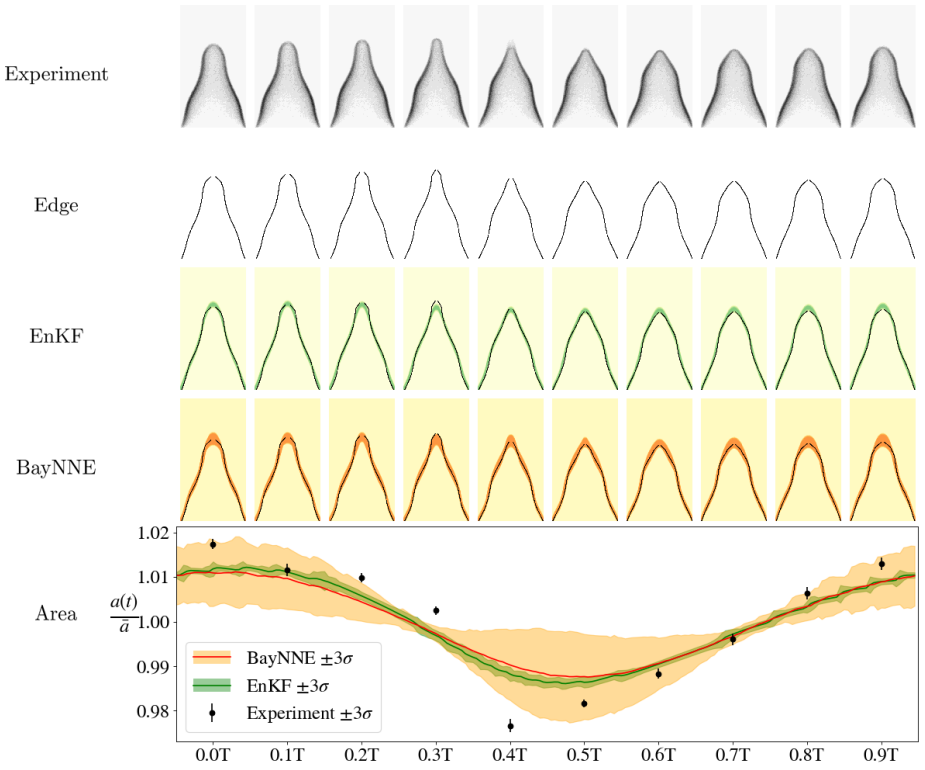

Fig. 5. Results of inference on a flame with flow rates of ethene 0.40 NL/min, methane 0.20 NL/min, air 6.00 NL/min and forcing frequency 425 Hz. Top row: Bunsen flame whose parameters are to be estimated. Second row: the detected flame edge. Third row: re-simulated flames using EnKF estimated parameters. Fourth row: re-simulated flames using BayNNE estimated parameters. Bottom: normalised surface area variations over one period.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

4 Conclusions

This study proposes a method to assimilate data from a Bunsen flame experiment into a kinematic model of a flame edge. The model parameters and their uncertainties are inferred using heteroscedastic Bayesian neural network ensembles. The neural networks are trained on a library of synthetic flame edge observations created using a level-set solver, LSGEN2D. Once trained, the Bayesian neural network ensemble accurately predicts the parameters and uncertainties from 10 consecutive images of the Bunsen flame. If more images are provided, this method selects the 10 consecutive images that give the smallest uncertainty in the parameters. This method is more than 6 orders of magnitude faster than the ensemble Kalman filter, which produces the same expected values of the parameters but lower variances. The Bayesian neural network ensemble also infers parameters of the model which cannot be inferred with the ensemble Kalman filter. Future work will focus on improving the parameter and uncertainty estimates by using more flame image data without necessarily increasing the size of the neural networks.

Disclosure of Funding

This project has received funding from the UK Engineering and Physical Sciences Research Council (EPSRC) award EP/N509620/1 and from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement number 766264.

References

1. Juniper, M. P., Sujith, R.: "Sensitivity and Nonlinearity of Thermoacoustic Oscillations", Annual Review of Fluid Mechanics (2018).
2. Keller, J. J.: "Thermoacoustic Oscillations in Combustion Chambers of Gas Turbines", AIAA 33-12 (1995).
3. Strutt, J. W.: "The Theory of Sound (Vol II)", Macmillan and Co., London, UK (1878).
4. Smart, A. E., Jones, B., Jewel, N. T.: "Measurements of Unsteady Parameters in a Rig Designed to Study Reheat Combustion Instabilities", AIAA 26-88 (1976).
5. Pitz, R. W., Daily, J. W.: "Experimental study of combustion in a turbulent free shear layer formed at a rearward facing step", AIAA 81-106 (1981).
6. Smith, D. A., Zukoski, E. E.: "Combustion instability sustained by unsteady vortex combustion", AIAA 85-1248 (1985).
7. Poinsot, T., Trouve, A., Veynante, D., Candel, S., Esposito, E. : "Vortex-driven acoustically coupled combustion instabilities", Journal of Fluid Mechanics 177-220 (1987), pp 265-292.
8. Crocco, L.: "Research on Combustion Instability in Liquid Propellant Rockets", Symp. (Int.) Combust., 12(1) (1969), pp. 85–99.
9. Williams, F. A.: "Turbulent Combustion", The Mathematics of Combustion (1985), pp. 97-131.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

10. Evensen, G.: "Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics", Journal of Geophysical Research 99 (1994) pp. 10143-10162.
11. Yu, H., Juniper, M. P., Magri, L.: "Combined State and Parameter Estimation in Level-Set Methods", J. Comp. Phys. 399 (2019).
12. Yu, H., Juniper, M. P., Magri, L.: "A data-driven kinematic model of a ducted premixed flame", Proceedings of the Combustion Institute, 399 (2020).
13. Goodwin, D. G., Speth, R. L., Moffat, H. K., Weber, B. W.: "Cantera: An object-oriented software toolkit for chemical kinetics, thermodynamics, and transport processes". https://www.cantera.org, (2018). Version 2.4.0.
14. Gal, Y.: "Uncertainty in Deep Learning", PhD Thesis (2016).
15. Damianou, A. C., Lawrence, N. D.: "Deep Gaussian Processes", Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS) (2013).
16. MacKay, D. J. C.: "Information theory, inference and learning algorithms", Cambridge university press (2003).
17. Gal, Y., Ghahramani, Z. : "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning", Proceedings of the 33 rd International Conference on Machine Learning, New York, NY, USA (2016).
18. Pearce, T., Zaki, M., Brintrup, A., Anastassacos, N., Neely, A.: "Uncertainty in neural networks: Bayesian ensembling", International Conference on Artificial Intelligence and Statistics (AISTATS) (2020).
19. Sengupta, U., Croci, M. L., Juniper, M. P.: "Real-time parameter inference in reduced-order flame models with heteroscedastic Bayesian neural network ensembles", Machine Learning and the Physical Sciences workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS) (2020).
20. Sengupta, U., Amos, M., Hosking, J. S., Rasmussen, C. E., Juniper, M. P, Young, P. J.: "Ensembling Geophysical Models with Bayesian Neural Networks", Advances in Neural Information Processing Systems (NeurIPS) (2020).
21. Kashinath, K. and Li, L. K. B. and Juniper, M. P.: "Forced synchronization of periodic and aperiodic thermoacoustic oscillations: lock-in, bifurcations and open-loop control", Journal of Fluid Mechanics, 838 (2018) pp. 690-714.
22. Hemchandra, S.: "Dynamics of Turbulent Premixed Flames in Acoustic Fields", PhD Thesis (2009).
23. Lakshminarayanan, B., Pritzel, A., Blundell, C.: "Simple and scalable predictive uncertainty estimation using deep ensembles", Advances in neural information processing systems (2017) pp. 6402–6413.
24. Luo, X. and Hoteit, I.: "Robust ensemble filtering and its relation to covariance inflation in the ensemble Kalman filter", Monthly Weather Review, 139(12) (2011) pp. 3938-3953.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33

A Supplementary material: Hyperparameter settings

Table 1. Hyperparameter settings.

Hyperparameter	Value
Training	
Train-test split	80:20
Batch size	2048
Epochs	5000
Optimiser	Adam
Learning rate	10^{-3}
Architecture	
Input units	900
Hidden layers	4
Units per hidden layer	900
Output layers	2
Units per output layer	6
Ensemble size	20

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77977-1_33