RESEARCH ARTICLE | AUGUST 08 2023

Probabilistic deep learning of turbulent premixed combustion

Junsu Shin ; Victor Xing ; Michael Pfitzner ; Corentin Lapeyre

AIP Advances 13, 085110 (2023)
https://doi.org/10.1063/5.0146268

Probabilistic deep learning of turbulent premixed combustion

Cite as: AIP Advances 13, 085110 (2023); doi: 10.1063/5.0146268
Submitted: 12 February 2023 • Accepted: 11 May 2023 •
Published Online: 8 August 2023

Junsu Shin,[1,a] Victor Xing,[2] Michael Pfitzner,[1] and Corentin Lapeyre[2]

AFFILIATIONS

[1] Bundeswehr University Munich, Werner-Heisenberg-Weg 39, 85577 Neubiberg, Germany
[2] CERFACS, 42 avenue Gaspard Coriolis, Toulouse 31057, France

[a] Author to whom correspondence should be addressed: junsu.shin@unibw.de

ABSTRACT

A probabilistic data-driven approach that models the filtered reaction rate in large-eddy simulation (LES) is investigated. We propose a novel framework that incorporates a conditional generative adversarial network and a Gaussian mixture model to take into account the statistical fluctuations that are present in LES of turbulent reacting flows due to non-resolved subgrid structures, which cannot be predicted by purely deterministic models and machine learning algorithms. The data from a direct numerical simulation of turbulent premixed combustion are spatially filtered using a wide range of filter widths and employed for the training. We extract physically relevant parameters from the database and reduce the input features to the network to the most influential ones based on the result of feature importance analysis. The trained model is then tested on unseen timesteps and untrained LES filter widths, where it is able to accurately predict the distribution of the filtered reaction rate.

I. INTRODUCTION

Turbulent premixed flames of fuel-oxidizer combinations with large activation energies are present in many technical combustion applications. Prominent examples are flames using hydrogen or hydrocarbons as fuel and pure oxygen or air as the oxidizer. Large activation energies and low diffusivities of gaseous components result in very thin reaction layers of premixed flames even at atmospheric pressure. With increasing pressure and density, chemical reaction rates rise while the diffusivities and heat conductivity drop, reducing the reaction layer thickness further.

These thin reaction layers usually cannot be resolved numerically in Reynolds averaged Navier–Stokes (RANS) simulations or in large-eddy simulations (LES) where the size of typical computational cells is more than an order of magnitude larger than necessary to numerically resolve the laminar flame structure embedded in the turbulent flow field. Combustion models that represent the effects of these subgrid structures are required to accurately simulate premixed combustion processes in most technical applications and academic configurations.

Reaction layers are folded and stretched by the turbulent flow field, but there is evidence from experiments and direct numerical simulations (DNS) that even at quite large Karlovitz numbers, their inner structure remains largely unaffected.[1–3] The filtered reaction term will then be roughly proportional to the laminar filtered one multiplied by a wrinkling factor representing the subgrid flame surface density. These results also suggest that such turbulent flames may be approximately characterized by a single reaction progress variable. The (density-weighted) Favre averaged transport equation for such a progress variable c can be written as

$$\frac{\partial \bar{\rho}\tilde{c}}{\partial t} + \frac{\partial \bar{\rho}\tilde{u}_k\tilde{c}}{\partial x_k} + \frac{\partial}{\partial x_k}\left(\bar{\rho}\widetilde{u_k c} - \bar{\rho}\tilde{u}_k\tilde{c}\right) = \frac{\partial}{\partial x_k}\left(\overline{\rho D_{diff}\frac{\partial c}{\partial x_k}}\right) + \bar{\dot{\omega}}, \quad (1)$$

where $\bar{\dot{\omega}}$ is the filtered chemical reaction source term and $\tilde{c}$ is the Favre average of the progress variable.

The quasi-1D structure of many turbulent premixed flames is used in many turbulent combustion models. In models of the filtered laminar flame (FLF) model[4] type, the chemical source term is filtered from a 1D laminar flame profile and tabulated as a function of the 1D filtered progress variable and of filter width. At runtime, the tabulated source term is multiplied by a suitable wrinkling factor from a separate model. Pfitzner *et al.*[5] proposed recently an FLF variant that considers the effect of oblique flame propagation through the filter volume on the flame surface area. During the runtime of an LES, Eq. (1) will be solved with $\bar{\dot{\omega}}$ from the model, which only

depends on the filter size and the transported LES variables. The filtered diffusion term may have to be modeled as well if ρD_{diff} is a strongly nonlinear function of the progress variable. Here, we follow a similar route, however, representing the filtered reaction rate source term through machine learning based on data from a DNS simulation.

For non-unity Lewis number fuels, such as lean hydrogen combustion, accurately modeling the combustion requires more than just a single quantity such as the progress variable. This is because the fuel's diffusivity is different from that of the rest of the mixture. Local variations in equivalence ratio can cause thermo-diffusive instabilities, leading to unstable combustion, flashback, and blow-off. To accurately model non-unity Lewis number flames, an additional transport equation for the mixture fraction is needed.[6–8]

The filtering operation in LES or RANS destroys information about high-resolution structures of the velocity and progress variable fields. After filtering, the full details of the fields present in a fully resolved (i.e., DNS) representation usually cannot be fully reconstructed from the filtered variable fields due to loss of information. A certain LES field can obviously result from a multitude of different DNS representations.[9] While the filtered value of a linear function of transported variables is equal to the function of the filtered variables, this is not so for very nonlinear functions like chemical source terms. In the LES transport equations, the effect of non-resolved structures on non-linear functions of transported variables needs, therefore, to be modeled. An example is the use of a wrinkling factor model capturing the effect of non-resolved subgrid flame wrinkling.

As an alternative, subgrid effects can be represented by probability density functions (PDFs, which in the LES context are often called filtered density functions, FDF).[10,11] Usually, single-point PDFs are used, which can either be presumed or derived from their transport equation. To determine the parameters of presumed PDFs, transport equations for moments of the relevant quantities (i.e., velocities and species) are solved. Alternatively, transport equations for the PDF itself can be solved.[12] The advantage is that the very nonlinear chemical source term appears in closed form in this transport equation but here the effect of turbulent fluctuations is unclosed. Due to the high dimensionality of the PDF (when using it with detailed chemistry mechanisms where each species generates a new dimension in pdf space), usually stochastic methods are used to approximate the solution of the PDF transport equation. The PDF is represented by a number of particles (or Eulerian fields), which are transported by stochastic transport equations. A stochastic term in these transport equations simulates the effect of non-resolved turbulent fluctuations of the velocity and species fields. Local mean values of variables are extracted from the ensemble of PDF particles (or fields) as simple averages.

Such stochastic methods will not simply yield the mean value (plus in some cases variances) of the transported quantities. Instead, the stochastic process will produce an approximation to the full PDF, from which theoretically the mean values and all higher moments can be derived, and often represented by linear superposition of multiple PDFs resulting in the joint distribution of mean and covariance. The models that estimate stochasticity underlying in data have been successfully applied to a wide variety of engineering applications, such as the prediction of atmospheric turbulence[13,14] and modeling in molecular dynamics simulation.[15,16]

In contrast to deterministic models, stochastic models will not yield the same solution when starting from a certain set of initial values. Thus, several simulations need to be run and averaged to compute mean values (and their variances) or long averaging times have to be run in the case of steady-state situations. A potential advantage of stochastic methods is that, in principle, it may be easier to model the effect of backscatter from the non-resolved (DNS) scales to the resolved (LES) ones. This is, however, seldomly done. Recently, Akram *et al.* investigated a reduced order LES model based on the inertial manifold applicable to non-reacting flows and it was capable of reconstructing unresolved dynamics (backscatter) without using the stochastic method.[17]

Another fact that supports the necessity of stochastic models can be introduced by the incomplete nature of under-resolved simulations such as RANS and LES. In these simulations, turbulence often has to resort to the coarse-grained models at the scales for which numerical resolutions are not sufficient, which causes *predictive variability*.[18] Deterministic models, e.g., the eddy-viscosity concept, rely on the assumption that the turbulence can be perfectly expressed by a finite set of local vectors, and it can generate only single unique evolution rather than predicting all the evolutions possible. However, the models based on uncertainty are capable of estimating a broad range of possible evolutions.[18]

When filtering LES quantities from DNS, for strongly nonlinear functions (e.g., the reaction source term) filtered with a large filter size, it is probable that the filtered functions cannot be fully described by functions of the filtered LES transported variables alone. Due to the *variability* arising from the coarse-grained description, plots of the filtered reaction source term conditioned on LES variables (or functions thereof) will exhibit some scatter with a certain variance. While a perfect model using the chosen set of LES variables will reproduce the conditioned filtered DNS data, the remaining scatter will be a sign of the inability of models with the chosen variables to represent the effect of subgrid structures (irreducible error[19] and optimal LES model[20]).

Deep learning has been applied to model a wide range of LES subgrid-scale (SGS) combustion quantities, such as scalar dissipation rates,[21,22] filtered density functions,[22–24] flame surface densities,[25–27] progress variable variances,[28,29] and filtered reaction rates directly.[30–32] SGS modeling is usually set up as a supervised regression task, with the model trained to minimize a least squared error with ground truth SGS data computed from a filtered DNS. The most straightforward network architecture is a multilayer perceptron composed of fully connected layers, which provides fully local predictions.[21–24,26] Shallow convolutional neural networks (CNNs) have often been used to incorporate short-range contextual information.[26–28,30,31] CNNs have pioneered exceptional breakthroughs in image and video computer vision[33–35] and are also well-suited to work on numerical combustion data discretized on large multidimensional grids. Deep CNNs with fully convolutional architectures, such as U-Nets[25,29] and ESRGANs,[36] benefit from efficiently working on the entire field for increased modeling precision and sampling efficiency at the cost of increased parameter count and model complexity.

Although many combustion models based on deep learning have been proposed, most of the studies[23,25,28–30] have applied deterministic machine learning modeling to the turbulent reacting flows,

which can be considered as a result of a random process driven by the turbulence. Recently, Ihme *et al.*[37] emphasized the importance of a probabilistic approach and statistical learning for the machine learning frameworks that analyze and model the combustion data.

While deterministic models will reproduce the mean values of variables, a stochastic method can attempt to also represent the scatter around the mean values with correct statistical properties. The generative adversarial neural network (GAN) is a type of deep learning algorithm that is capable of probabilistic prediction to represent the stochastic relationship that underlies any observed data. Goodfellow *et al.* argued that the GANs could actually learn the probability latent in the data under the condition that the training data are sufficiently detailed and the network architecture is fully capable of mimicking the data.[38] In the computational fluid dynamics (CFD) community, the GANs have been successfully used in the representation of unsteady hydrodynamic data[39] and high-fidelity combustion data.[36]

Probabilistic modeling has been used to super-resolve the low-resolution fields of scalar and vector fields using GANs.[36,40] Given low-resolution filtered fields, the generator of the GAN is adversarially trained to generate corresponding high-resolution fields. At inference time, it can be queried to generate samples from this distribution of high-resolution data. Probabilistic neural networks (PNNs) have also been investigated to quantify the uncertainty of neural network predictions in surrogate modeling of fluid flows.[41] PNNs are trained to output a set of mixing probabilities, means, and variances that defines a Gaussian mixture model, which can, therefore, provide uncertainty estimates over its predictions.

We propose a novel data-driven turbulent combustion model that provides a stochastic prediction of the filtered reaction rate source term using an uncertainty-aware deep learning method. In Sec. II, we present the DNS of a premixed turbulent flame in a slot burner configuration that is used as our training data. Section III describes the details of the GAN-based modeling, with an emphasis on its motivation and distinctive factors compared to the previous probabilistic modeling we have conducted. Results for the feature importance analysis to reduce the complexity of the input parameters are presented in Sec. IV A. The test results of the trained model for the unseen time snapshot and untrained LES filter widths are provided in Sec. IV B. Finally, the conclusion of this study is presented in Sec. V.

II. DIRECT NUMERICAL SIMULATION DATABASE

A. Characteristics of the DNS

The slot burner Bunsen configuration studied here was chosen as it contains a large surface area of wrinkled flame fronts covering

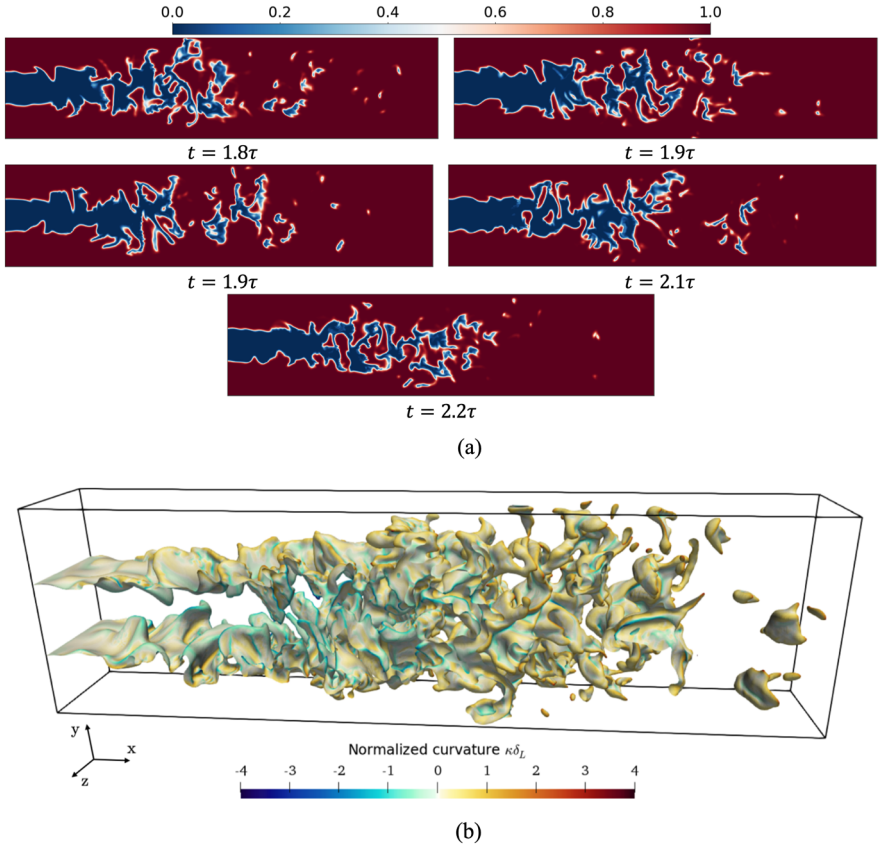

FIG. 1. Visualization of the DNS. (a) Progress variable c slice view in the xy-plane over time. (b) 3D view of the iso-surface $c = 0.75$ at $t = 2.2\tau$ corresponding to the progress variable value at the peak heat release in an unstrained laminar flame, colored by the normalized curvature $\kappa\delta_L$.

a wide range of wrinkling intensity from quasi-laminar flame fronts near the inlet to highly wrinkled regions downstream that feature flame front interactions and detached pockets of the unburnt mixture. It is, therefore, well-suited to be converted into a diversified database to train a machine learning model to predict the wrinkling factor, specifically for flames with a unity Lewis number. Visualizations of the flame in its steady state are shown in Fig. 1, and key simulation parameters are recalled in Table I.

Chemistry is described using a recently proposed reaction source term formulation.[42–44] It relies on an analytical solution to the 1D laminar transport equation of the progress variable, which leads to a reaction source term $\omega(c)$ expressed as a polynomial expression in the progress variable c. A reaction source term that provides an analytical solution of the 1D laminar progress variable conservation equation in a canonical coordinate space is

$$\omega_m(c) = (m+1)\left(1-c^m\right)c^{m+1}. \tag{2}$$

This non-dimensional source term describes a single-step irreversible chemical mechanism and depends on a scalar parameter m, which determines the flame thickness. It mimics closely a one-step Arrhenius source term. This simplified description of the chemistry was chosen to reduce the computational cost of the DNS since it is generally accepted that wrinkling statistics of premixed flames can be studied using simplified chemistry and transport assumptions[45–47] and that the models that are derived can be adapted to simulations with detailed chemistry and transport.[48]

A single-step irreversible mechanism is derived from the reaction $CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O$. A progress variable is defined from the methane mass fraction Y_F and its unburnt value Y_F^u,

$$c = 1 - \frac{Y_F}{Y_F^u}. \tag{3}$$

Under unity Lewis number conditions, the rate of progress Q of the reaction has been computed as in Ref. 29

$$Q = \frac{(\rho_u s_L)^2 Y_F^u}{\rho D W_F}\omega_m, \tag{4}$$

where ρ_u, s_L, ρ, D, and W_F are the unburnt density, laminar flame speed, density, diffusivity, and fuel molar mass, respectively. An unburnt gas temperature of $T_u = 300$ K and a constant pressure $P = 1$ bar are set. m and s_L are inputs to the source term formulation, which are chosen to match the laminar flame speed and thermal flame thickness δ_L of a GRI 3.0 detailed mechanism containing 53 species and 325 reactions,[49] leading to $s_L = 0.37$ m/s and $m = 1.9$ ($\delta_L = 450\ \mu$m), where δ_L is defined by $(T_b - T_u)/\max(|\nabla T|)$, where T_b and T_u are the temperature of the burnt mixture and unburnt mixture, respectively. Transport properties are set to a constant unity Lewis number and a constant Prandtl number $\mathrm{Pr} = \frac{\mu C_p}{\lambda} = 0.7$, where C_p is the heat capacity at constant pressure, λ is the thermal conductivity, and μ is the dynamic viscosity that is modeled by a power law

$$\mu = \mu_{ref}\left(\frac{T}{T_{ref}}\right)^{\alpha}. \tag{5}$$

The reference dynamic viscosity $\mu_{ref} = 1.8 \times 10^{-5}$ kg/m/s at $T_{ref} = 300$ K and the power-law exponent $\alpha = 0.683$ are fitted on a GRI 3.0 mechanism. Species diffusivities $D_k = \frac{\mu}{\rho Sc_k}$ are computed through constant Schmidt coefficients $Sc_k = \mathrm{Pr}$ to satisfy the unity Lewis number assumption.

This DNS was performed using the AVBP code,[50,51] which solves the three-dimensional compressible Navier–Stokes equations with a third-order finite element TTGC Taylor-Galerkin numerical scheme.[52] The domain is a rectangular box of size $L_x \times L_y \times L_z = 10.24 \times 2.56 \times 1.28$ cm^3 discretized on a regular Cartesian mesh composed of $1600 \times 400 \times 200$ hexahedral cells of size $dx = 64\ \mu$m. NSCBC inlet and outlet boundary conditions[53] are imposed in the x streamwise direction. A bulk flow of unburnt mixture is injected at a bulk velocity $U_{bulk} = 25$ m s^{-1} in the central third of the inlet $H = L_y/3 = 8.5$ mm. It is surrounded by coflows of burnt mixture with a velocity $U_{coflow} = 0.1$ m s^{-1}. In the unburnt–burnt transition region at the inlet boundary, species mass fraction and temperature profiles are imposed to follow the laminar flame profile

TABLE I. Numerical and physical parameters. Unless specified, the values of the physical parameters are taken from the inlet unburnt mixture conditions.

Jet slot width $H = L_y/3$	8.5 mm
Domain size $L_x \times L_y \times L_z$	$12\,H \times 3\,H \times 1.5\,H$
Bulk flow velocity U_{bulk}	25 m/s
Coflow velocity U_{coflow}	0.1 m/s
Flow-through time $\tau = L_x/U_{bulk}$	4.1 ms
Laminar flame speed s_L	0.37 m/s
Laminar thermal flame thickness δ_L	450 μm
Turbulent fluctuation intensity u'	2.5 m/s
Integral length scale l_t	2.1 mm
Preheat zone Kolmogorov length scale η	54 μm
Damköhler number $Da = (l_t/\delta_L)/(s_L/u')$	0.69
Karlovitz number $Ka = (u'/s_L)^{3/2}(l_t/\delta_L)^{-1/2}(s_L\delta_L/\nu)^{1/2}$	27
Jet Reynolds number $Re = U_{bulk}H/\nu$	14 200

corresponding to the DNS operating conditions, and a smooth transition from U_{bulk} to U_{coflow} is enforced through a hyperbolic tangent function. The remaining boundary conditions are no-slip adiabatic walls in the y-direction and periodic conditions in the z-direction.

Turbulent velocity fluctuations are added to the bulk flow using a synthetic generation method.[54] A turbulent velocity fluctuation field u' is built from a Fourier series decomposition

$$u'(r,t) = \sum_{n=1}^{N} \boldsymbol{v}^n(\boldsymbol{k}^n)\cos(\boldsymbol{k}^n \cdot r + \boldsymbol{\omega}^n t) + w^n(\boldsymbol{k}^n)\sin(\boldsymbol{k}^n \cdot r + \boldsymbol{\omega}^n t), \tag{6}$$

with $N = 200$ modes. The Fourier modes $\{\boldsymbol{v}^n, \boldsymbol{w}^n\}$, wave vectors $\boldsymbol{k}^n$, and pulsations $\boldsymbol{\omega}^n$ are random variables sampled to obtain a Passot–Pouquet[55] turbulence spectrum,

$$E(k) = 16\frac{u'^2}{k_e}\sqrt{\frac{2}{\pi}}\left(\frac{k}{k_e}\right)^4 \exp\left[-2\left(\frac{k}{k_e}\right)^2\right], \tag{7}$$

where $u' = 2.5$ m s^{-1} $= U_{bulk}/10$ is the turbulent fluctuation intensity and k_e is the wave number associated with the most energetic length scale. k_e is related to the integral length scale l_t of the spectrum through $l_t = \sqrt{2\pi}/k_e$, and its value is chosen to obtain $l_t = 2.1$ mm $= H/3$.

The impact of the mesh size on the resolution of the flame front and the turbulence is discussed in the following. The thickness of the laminar flame front δ_L is resolved on ~7 mesh points, namely, $\delta_L \approx 7 \cdot dx$, where dx is the length of the unit cell. This allows for an appropriate resolution of the flame structure with a simplified single-step chemical mechanism.

The Kolmogorov length scale is not measured from the theoretical Passot–Pouquet turbulence spectrum, as the synthetic generation does not produce eddies as small as the theoretical Kolmogorov length scale, and the shear layers at the inlet also contribute to the turbulence seen by the flame. As a consequence, the Kolmogorov length scale is computed from the density-weighted averages in the preheat zone (defined as the region where $0.1 < c < 0.5$) of the kinematic viscosity $\tilde{\nu}$ and dissipation rate $\tilde{\varepsilon}$. Both solenoidal and dilatational contributions to the dissipation rate are considered,[56]

$$\tilde{\varepsilon} = \frac{1}{\tilde{\rho}}\left[\overline{\mu\omega^2} + \frac{4}{3}\overline{\mu(\nabla\cdot\boldsymbol{u})^2} + 2\overline{\mu\frac{\partial^2}{\partial x_i \partial x_j}(u_i u_j)} - 4\overline{\mu\frac{\partial}{\partial x_i}(u_i \nabla\cdot\boldsymbol{u})}\right]. \tag{8}$$

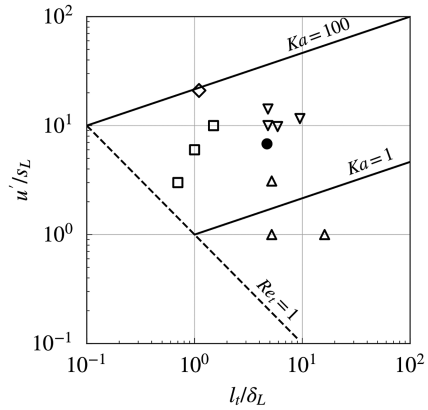

FIG. 2. Borghi–Peters turbulent combustion diagram. •: present DNS; ▽: Luca *et al.*, 2019;[3] □: Sankaran *et al.*, 2015;[59] ◇: Wang *et al.*, 2017;[60] △: Klein *et al.*, 2018.[61]

The value of the resulting Kolmogorov scale $\eta = (\tilde{\nu}^3/\tilde{\varepsilon})^{1/3}$ $= 54$ μm guarantees that the Pope criterion $\Delta x/\eta < 2$ is fulfilled for the turbulent eddies near the cold boundary of the flame front.

The operating point of the present DNS in the Borghi–Peters turbulent combustion diagram[57,58] is represented in Fig. 2, along with similar methane–air slot jet flame DNS configurations (or single-step virtual chemistry in the case of Klein *et al.* 2018.[61]) from the literature. For the comparison of configurations with varying temperature and pressure conditions, the $Re_t = 1$, $Ka = 1$, and $Ka = 100$ lines are plotted with the assumption that $s_L\delta_L/\nu = 1$ in the unburnt mixture, although for the present DNS $s_L\delta_L/\nu = 11$.

B. Data pre-processing

Spatially filtered variables are generated from exact DNS solutions through a convolution operation with a kernel G,

$$\overline{Q}(\boldsymbol{x}) = (G*Q)(\boldsymbol{x}) = \int G(\boldsymbol{x}-\boldsymbol{x}')Q(\boldsymbol{x}')d\boldsymbol{x}'. \tag{9}$$

Following modern conventions in *a priori* testing of LES models,[62,63] G is a 3D Gaussian kernel. For computational efficiency, it is factorized in three 1D Gaussian kernels G_{1D} successively applied along the x, y, z-directions,

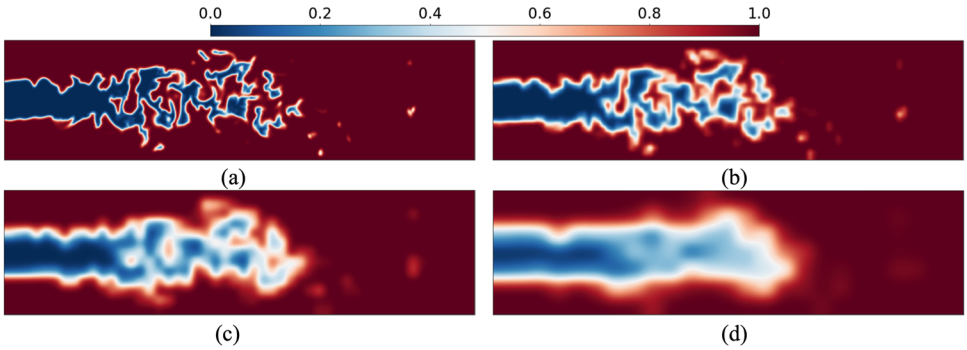

FIG. 3. Center slice view of Favre-filtered progress variable $\tilde{c}$ at $t = 2.2\tau$ for different filter widths. (a) $\Delta/\delta_{th} \approx 0.57$ ($n_\Delta = 4$) (b) $\Delta/\delta_{th} \approx 1.71$ ($n_\Delta = 12$). (c) $\Delta/\delta_{th} \approx 3.43$ ($n_\Delta = 24$). (d) Δ/δ_{th} ≈ 9.14 ($n_\Delta = 64$).

$$G_{1D}(r) = \left(\frac{6}{\pi\Delta^2}\right)^{1/2} \exp\left(\frac{-6r^2}{\Delta^2}\right), \tag{10}$$

where Δ is the size of the filter and denotes its cutoff length. Filtered variables are then downsampled by a factor $n_\Delta = \Delta/dx$ to reproduce the coarse resolution of LES meshes. The filtered reaction source term $\overline{\dot{\omega}}$ learned by the ML method is calculated by evaluating $\omega_m(c)$ from the DNS c field using Eq. (2) and filtering according to Eq. (9). Figure 3 shows the result of the filtering process of $\tilde{c}$ for different filter widths. We have filtered the DNS database with seven filter widths of $\Delta/\delta_{th} \approx 0.57, 1.14, 1.71, 2.29, 3.43, 4.57, 9.14$, which corresponds to $n_\Delta = 4, 8, 12, 16, 24, 32, 64$.

C. Data mining from DNS database

Data-driven modeling heavily relies on the quality of the input data as it finds a mapping function between input and output, which are often called *feature* and *label*, in the supervised learning concept. Hence, determining which *feature* will enter the model is critical.

In the context of machine learning, feature importance analysis refers to techniques that provide a quantitative measure of how strongly input *features* are correlated with the target variable *label*, giving us better interpretability of observed data.

A total of 11 features that are expected to represent the filtered reaction rate $\overline{\dot{\omega}}$ are selected as given in Table II. These features can be computed using the variables available in an LES. They are selected based on knowledge of turbulent combustion physics and on features used in existing models. The importance of the Favre averaged progress variable $\tilde{c}$, the magnitude of the gradient of Favre filtered progress variable $|\nabla\tilde{c}|$, and the magnitude of the Laplacian of Favre filtered progress variable $|\nabla^2\tilde{c}|$ are already studied in Ref. 26. Parameters that are related to the velocity fields are also considered: the magnitude of SGS velocity fluctuations u'_Δ, the magnitude of the filtered velocity $|\tilde{u}_i|$, the magnitude of the gradient of the filtered velocity $|\nabla\tilde{u}_i|$, the magnitude of the filtered strain rate tensor $|S_{ij}|$, the magnitude of the filtered vorticity rate tensor from the velocity fields $|\omega_{ij}|$, the resolved curvature $\tilde{\kappa}$, the resolved tangential strain rate $\tilde{a}_T$, and the LES filter width Δ. The relationship between the filtered reaction rate $\overline{\dot{\omega}}$ and the curvature κ, tangential strain rate a_T has been studied in Refs. 64 and 65. It should be noted that $\tilde{c}$, $|\nabla\tilde{c}|$, $|\nabla^2\tilde{c}|$, u'_Δ, $|S_{ij}|$, $|\omega_{ij}|$, $\tilde{\kappa}$, and $\tilde{a}_T$ are Galilean invariant. The parameters are normalized by δ_{th} and s_L as given in Table II and then used for the training. Figure 4 depicts slices of the selected parameters at $\Delta/\delta_{th} = 2.29$ as an example.

TABLE II. Full set of features extracted from the DNS to model the filtered reaction rate.

Features	Definition	Normalization
$\tilde{c}$	$\tilde{c} = \frac{\overline{\rho c}}{\bar{\rho}}$	...
$\lvert\nabla\tilde{c}\rvert$	...	$1/\delta_{th}$
$\lvert\nabla^2\tilde{c}\rvert$	...	$1/{\delta_{th}}^2$
u'_Δ	$u'_\Delta = \sqrt{\frac{1}{3}\frac{\tau_{ii}}{\bar{\rho}}}, \tau_{ii} = \overline{\rho u_i u_i} - \left(\frac{\overline{\rho u_i}\times\overline{\rho u_i}}{\bar{\rho}}\right)$	s_L
$\lvert\tilde{u}_i\rvert$	...	s_L
$\lvert\nabla\tilde{u}_i\rvert$	...	s_L/δ_{th}
$\lvert S_{ij}\rvert$	$\lvert S_{ij}\rvert = \left\lvert\frac{1}{2}\left(\frac{\partial\tilde{u}_i}{\partial x_j} + \frac{\partial\tilde{u}_j}{\partial x_i}\right)\right\rvert$	s_L/δ_{th}
$\lvert\omega_{ij}\rvert$	$\lvert\omega_{ij}\rvert = \lvert\Omega_x + \Omega_y + \Omega_z\rvert, \Omega_{ij} = \frac{1}{2}\left(\frac{\partial\tilde{u}_i}{\partial x_j} - \frac{\partial\tilde{u}_j}{\partial x_i}\right)$	s_L/δ_{th}
$\tilde{\kappa}$	$\tilde{\kappa} = \frac{\partial N_i}{\partial x_i}, N_i = -\frac{\partial\tilde{c}}{\partial x_i}/\lvert\nabla\tilde{c}\rvert$	$1/\delta_{th}$
$\tilde{a}_T$	$\tilde{a}_T = (\delta_{ij} - N_i N_j)\frac{\partial\tilde{u}_i}{\partial x_j}, \delta_{ij} \equiv \begin{cases} 0, & i \neq j \\ 1, & i = j \end{cases}$	s_L/δ_{th}
Δ	...	δ_{th}

To keep the most important input features and prune the unnecessary features, we employed the maximal information coefficient (MIC)[66] and Shapley value (SHAP)[67] methods. The MIC method provides a non-linear correlation measure between two continuous features x, y. We consider the set of all possible binnings of (x, y) containing at most n_x bins for x and n_y bins for y, where n_x, n_y are constrained by the user. For each binning $(\hat{x}, \hat{y})$, the mutual information $I(\hat{x}, \hat{y})$, defined as the Kullback–Leibler divergence between the joint distribution $p(\hat{x}, \hat{y})$ and the product of the marginal distributions $p(\hat{x})p(\hat{y})$, is computed. The maximal information coefficient is the largest mutual information over the set of all the binnings, with appropriate normalization between 0 and 1.

While the MIC method only operates on the features, the SHAP algorithm needs a pre-trained model because this method computes the average marginal contribution of each *feature* with respect to the prediction of the model. A previous study[26] has demonstrated how the input variables computed from the filtered progress variable $\bar{c}$ influence on the target variable, flame surface density $\overline{\Sigma}$, using the SHAP method. Recently, Wang *et al.*[68] have employed the SHAP method to explain the effect of geometrical design parameters for turbomachinery systems and the MIC has been also exploited to interpret the turbulence CFD database to analyze uncertainty estimates for turbulence models.[69]

Results of the feature importance analysis using SHAP and MIC algorithms are indicated in Fig. 5. Regarding the SHAP analysis, a deep learning model that will be introduced in Sec. III has been utilized to compute the correlations with respect to the filtered reaction rate $\overline{\dot{\omega}}$. It is then trained on a total of 11 features listed in Table II in order to compute the correlation indices for all the features considering in this study. Both algorithms indicate that the filtered progress variable $\tilde{c}$ has the highest impact, followed by $|\nabla\tilde{c}|$, $|\nabla^2\tilde{c}|$, u'_Δ, Δ, and the rest, in descending order. Usually, the parameters $|\nabla\tilde{c}|$, u'_Δ, Δ have been widely used as the model input features of conventional flamelet surface density (FSD) models.[70–72] The higher impact of $|\nabla^2\tilde{c}|$ relative to u'_Δ has not been expected and the effect of $|\nabla^2\tilde{c}|$ has not been considered anywhere (to the best of the author's knowledge).

Based on the presented result of the feature importance analysis, two groups of parameters are defined: full and compact. The full set includes: $\tilde{c}$, $|\nabla\tilde{c}|$, $|\nabla^2\tilde{c}|$, u'_Δ, $|\tilde{u}_i|$, $|\nabla\tilde{u}_i|$, $|S_{ij}|$, $|\omega_{ij}|$, $\tilde{\kappa}$, $\tilde{a}_T$, and Δ, a total of 11 parameters. The compact set includes: $\tilde{c}$, $|\nabla\tilde{c}|$, $|\nabla^2\tilde{c}|$, u'_Δ, and Δ, a total of five parameters. In Sec. III, we compare the performance of the proposed deep learning surrogate models trained on the two sets of parameters.

III. PROBABILISTIC DEEP LEARNING MODELING

The contribution of this paper can be considered as an extension of a previous study[26] that proposed a data-driven turbulent combustion model using a residual neural network in conjunction

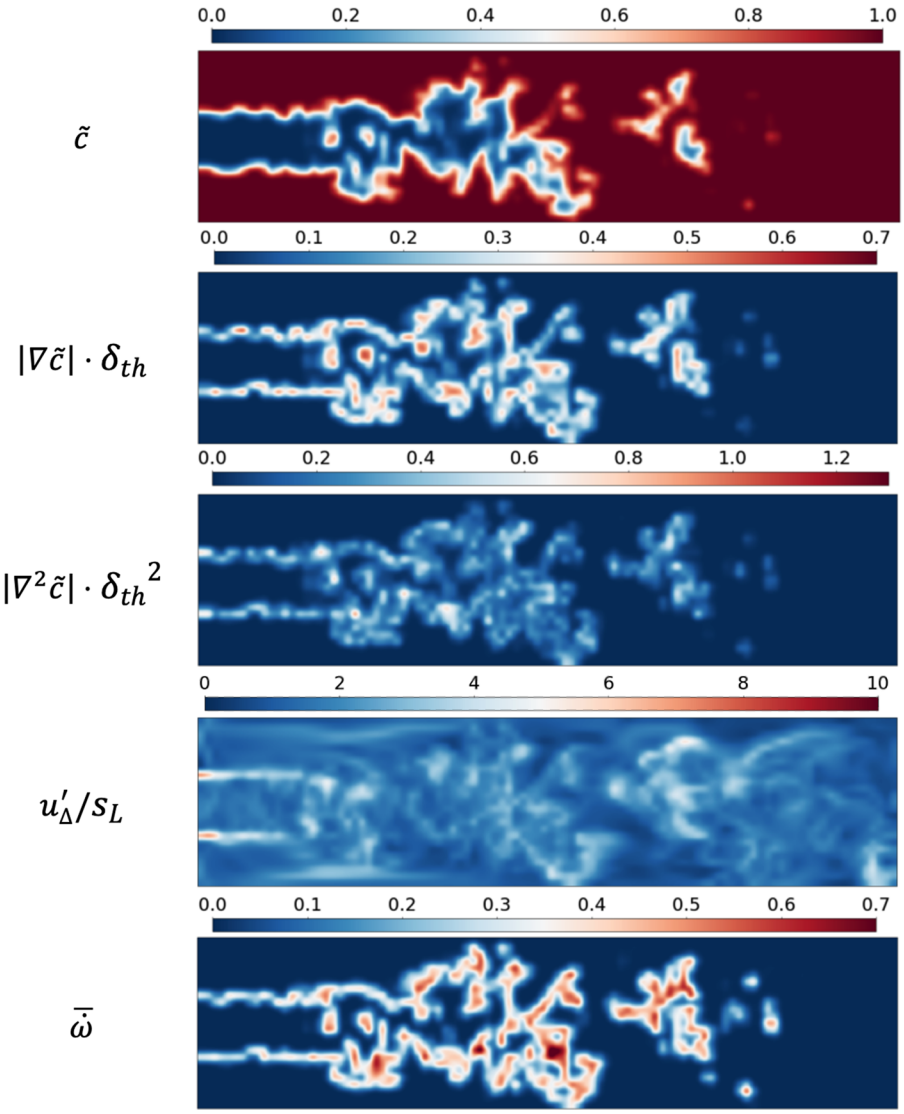

FIG. 4. Center slice view of the variables extracted for the filter width $\Delta/\delta_{th} = 2.29$ ($n_\Delta = 16$).

with a mixture density network (RES–MDN) to predict the statistical quantities such as the mean μ and standard deviation σ of the flame surface density $\overline{\Sigma}$. This work has shown that the RES–MDN has the ability to predict the stochastic behavior stored in the DNS database accurately in *a priori* as well as *a posteriori*.

This paper presents improvements along the following two perspectives.

(1) Introduction of a Gaussian mixture distribution: While a RES–MDN that forecasts μ and σ representing a single Gaussian distribution was employed in the previous study, this work employs a mixture of Gaussian distributions to approximate any arbitrary distribution. It assumes that the data distribution can be represented as a linear combination of kernel functions (e.g., Gaussian[73]). A Gaussian mixture has a probability density function that reads

$$p(y|x) = \sum_{i=1}^{K} \alpha_i(x)\phi_i(y|x),$$
$$\phi_i(y|x) = \frac{1}{(2\pi)^{1/2}\sigma_i(x)} \exp\left\{-\frac{(y-\mu_i(x))^2}{2\sigma_i(x)^2}\right\}, \tag{11}$$

where y is the target data, x is the input vector, K is the number of components in the mixture, and α is the mixing coefficient with the constraint $\sum_{i=1}^{K} \alpha_i = 1$. The parameters α, μ, and σ are unknown parameters to be predicted by the neural network. In this current contribution, we have used $K = 10$ distributions, which were found to adequately model the current filtered DNS database. Readers who are interested in the details of this approach can refer to chapter 5.2 in Ref. 74. It

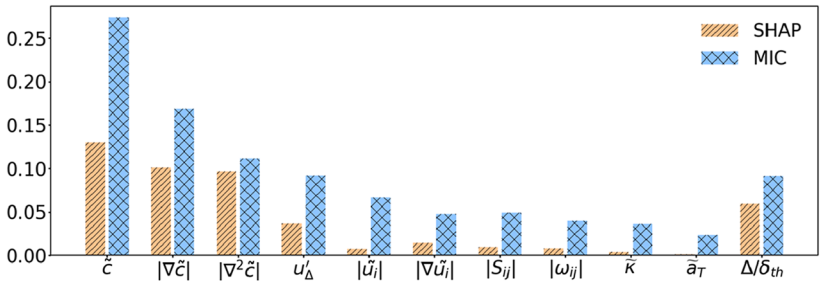

FIG. 5. Importance indices for each parameter computed by the feature importance analysis methods, SHAP and MIC.

has to be emphasized that the use of a Gaussian mixture distribution is motivated by the fact that the filtered DNS dataset for the current work covers several LES filter widths spanning a wide range from $\Delta/\delta_{th} \sim 0.57$ to ~9.14, whereas the training database in the previous work involved a single LES filter width $\Delta/\delta_{th} \sim 2.4$ only. This increases the complexity of the distribution to be modeled and justifies the use of a more sophisticated model.

(2) Conditional adversarial training: Adversarial training is used to train a GAN model that learns to generate outputs drawn from a distribution that is indistinguishable from the ground truth target distribution.[28] More specifically, adversarial training involves two neural networks, a generator G and a discriminator D. G is trained to generate outputs that closely match the target distribution of $\bar{\omega}$. D is alternatively fed "fake" samples generated by G and "true" samples from the ground truth distribution. D is trained to correctly classify the true and fake samples. G and D are jointly optimized by an adversarial loss. In the end, the trained network G is a generative model for the filtered reaction rate $\bar{\omega}$. While a traditional GAN learns to sample probability distribution $p(y)$, a conditional GAN (cGAN) can learn to sample conditional probability $p(y|x)$,[75] which means that cGAN actually can represent the conditional probability $p(\bar{\omega}|x)$ where $x = \{\tilde{c}, |\nabla\tilde{c}|, |\nabla^2\tilde{c}|, \ldots, \Delta\}$. The samples generated by the cGAN are, therefore, controlled by a set of informative features. Thus, the proposed model provides a mapping from $[x] \rightarrow \{\alpha^m, \mu^m, \sigma^m\} \rightarrow [\bar{\omega}]$, where m is the number of components in the mixture. Adversarial training is a good choice for probabilistic modeling as the generator is encouraged to learn an output distribution that is indistinguishable from the ground truth distribution, without exactly copying the ground truth values. Using traditional supervised training with a mean squared error loss function could collapse the model distribution to the specific sample distribution of the

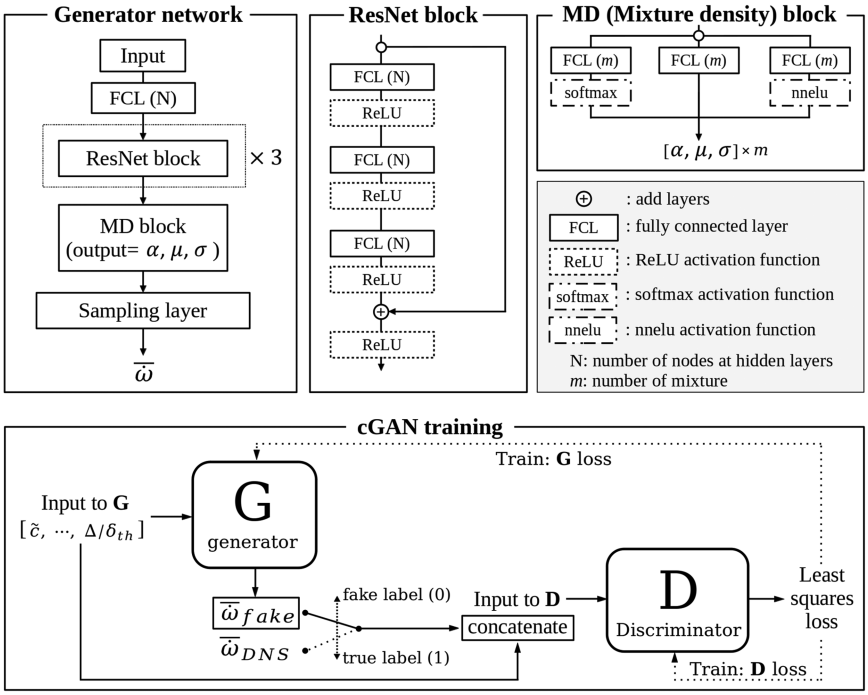

FIG. 6. An overview of the training of the conditional generative adversarial network (cGAN) and the deep learning network architecture employed in this study, where α, μ, and σ denote mixing coefficient, mean, and standard deviation, respectively.

training dataset, which is only a small sample of the true distribution to be learned. This learning task can be viewed as probabilistic regression, for which cGAN has been studied comprehensively in Refs. 76–79.

The schematic of the training process of cGAN is shown in Fig. 6. As described above, the input vector x enters the generator G in order to produce the fake $\bar{\dot{\omega}}$. The discriminator D alternatively takes the fake $\bar{\dot{\omega}}$ and the real $\bar{\dot{\omega}}$ from the filtered DNS as input. The information on whether $\bar{\dot{\omega}}$ is fake or not is also provided to D by concatenating the labels with binary tags 0 (fake) or 1 (real). The most challenging part of this study was to overcome mode collapse, which is a commonly experienced problem when dealing with GAN, in which the trained model fails to generate diversified outputs.[80–83] Thus, the predictive performance of the model is degraded significantly when mode collapse happens. To solve this issue, we have tested various common GAN loss functions: standard,[75] Wasserstein,[84] Wasserstein with gradient penalization,[85] and least-squares.[86] Training was only successful when using the least-squares loss function (LS–GAN) and the utility of the LS–GAN has been reported in many interdisciplinary sectors.[87–89] The generator G and the discriminator D are then finally trained by stochastic gradient descent with the ADAM optimizer.[90]

Regarding the generator G, the network processes the input x at its input layer. It is followed by a fully connected layer (FCL) with 30 neurons, which is the value used in all the layers in G. Then, three residual blocks are stacked repeatedly. A residual block consists of three sets of FCL and rectified linear unit (ReLU) activation function.[91] A residual skip connection is added to the main path before a final ReLU activation. A mixture density (MD) block is then connected to infer$\{\alpha, \mu, \sigma\}$. A softmax activation function is used for the prediction of the mixing coefficients α due to the requirement that they sum to unity.[92,93] The mathematical definitions of ReLU and softmax functions are provided in the Appendix, along with the non-negative exponential linear unit (nnelu) function. The nnelu function is used to ensure that the standard deviation σ always has a positive value.[94] A sampling layer is finally used to generate the sampled $\bar{\dot{\omega}}$ from the Gaussian mixture distribution. The ultimate output of the proposed model is a stochastic distribution sampled from the Gaussian mixture distribution, while the output of our previous work[26] was a set of statistical moments $\{\mu, \sigma\}$ defining a Gaussian distribution that was not yet sampled from. The schematic of the generator network is depicted in Fig. 6.

The network architecture of discriminator D was determined through trial and error. The mode collapse can be caused depending on the size of the discriminator network. In addition, if the D network is too large, vanishing gradient problems can occur.[95] For a GAN, the vanishing gradient problem denotes a situation where a discriminator close to optimality cannot provide enough feedback to the generator, making the training slow or stop completely. We have tried different sizes of the network and we figured out that a sufficiently large discriminator avoids both mode collapse and vanishing gradients for our case. Five residual blocks and 100 neurons are used to construct the D network.

For the training dataset, we have used five different time snapshots, which are $t = 1.8\tau, 1.9\tau, 2.0\tau, 2.1\tau, 2.2\tau$. For each snapshot, we built seven filtered snapshots using filter widths of $\Delta/\delta_{th} \approx 0.57$, $1.14, 1.71, 2.29, 3.43, 4.57, 9.14$. Table III describes which snapshots are included in the training and testing datasets. For the testing dataset, we selected the last time step 2.2τ and the LES filter widths $\Delta/\delta_{th} \approx 1.71, 3.43$. Therefore, what we aim to test is the extrapolation capability with respect to time and interpolation capability with respect to the filter width.

TABLE III. Datasets for training and testing.

Datasets	Time step t	LES filter width Δ/δ_{th}
Training	$1.8\tau, 1.9\tau, 2.0\tau, 2.1\tau$	0.57, 1.14, 2.29, 4.57, 9.14
Testing	2.2τ	1.71, 3.43

The hyperparameters employed in this study have been summarized in Table IV. The total training time was taken to be 32 h running on a Titan-X graphics processing unit (GPU) equipped workstation. In order to implement the deep learning model in practice, the Python library *Tensorflow probability*[96] has been utilized on top of *Tensorflow*.[97]

For the purpose of examining the effect of the number of input parameters entering the model, we have created two scenarios:

TABLE IV. List of hyperparameters used in this study. The values of learning rate, β_1, β_2, and ϵ are the default values of the Adam optimizer implemented in *Tensorflow*.

	Generator	Discriminator
Number of neurons at layers	30	100
Number of residual blocks	3	5
Number of trainable parameters	9660	152 901
Learning rate	0.001	0.001
β_1	0.9	0.9
β_2	0.999	0.999
$\hat{\epsilon}$	1×10^{-7}	1×10^{-7}
Weight decay	None	None
Optimizer	Adam	Adam
Loss function	Squared L2 norm	Squared L2 norm
Number of mini-batch size of training	1024	

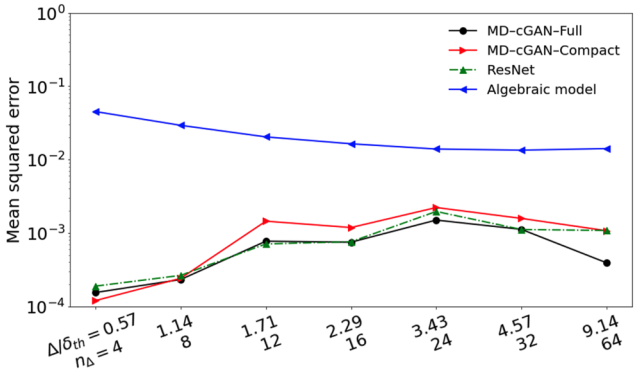

FIG. 7. Mean squared errors between predicted $\bar{\dot{\omega}}$ and $\bar{\dot{\omega}}$ from filtered DNS for different filter widths. The exact values are listed in Table V in the Appendix.

MD–cGAN–Full and MD–cGAN–Compact, which correspond to the proposed deep learning model with a total of 11 input parameters and the model with five parameters. Furthermore, as a comparison to MD–cGAN–Compact, we also set up a case without the mixture density network and with five input parameters in order to demonstrate the effect of the ability to estimate the probability density based on the Gaussian mixture model. The other parts of the network are equivalent to the network depicted in Fig. 6. The residual skip connection in this neural network, which allows a deeper neural network, is primarily responsible for its ability to infer the target parameter. This case is henceforth referred to as ResNet (abbreviated from residual neural network). The ResNet network architecture does not use conditional adversarial training, mixture density blocks, or sampling layers. Instead, it models the target parameter directly using only ResNet blocks. The hyperparameters for the ResNet block are the same as the ones used in MD–cGAN methods.

IV. RESULTS

In this section, we test the generalization capability of the proposed deep learning model on the unseen time snapshot for the last time step ($\tau = 2.2$)and untrained filter width $\Delta/\delta_{th} \approx 1.71, 3.43$ (equivalent to $n_\Delta = 12, 24$), as described in Sec. III.

Figure 7 shows the comparison of the quantitative performances of four cases, such as MD–cGAN–Full, MD–cGAN–Compact, ResNet, and algebraic model. As a basis of comparison against the deep learning models, we introduce a well-known FSD model from the literature, which was proposed by Fureby.[70,71] Details of the algebraic model can be found in the Appendix. Values of mean squared error (MSE) between the filtered DNS and predicted values are illustrated in Fig. 7 showing all the LES filter widths including the training and testing datasets. The exact values of MSE are provided in Table V in the Appendix.

It can be noticed that MD–cGAN–Full performs slightly better than MD–cGAN–Compact and is noticeably more accurate than the algebraic model. Note that the y axis is in logarithmic scale. For both deep learning models, the test results for the untrained LES filter widths ($\Delta/\delta_{th} \approx 1.71, 3.43$) do not deviate from the results of the trained LES filter widths. In addition, the trend of the errors indicates that the error levels increase as an LES filter width increases until $\Delta/\delta_{th} \approx 3.43$ and then decreases again. It also shows that ResNet performs similarly with two other deep learning cases. The inference times taken by the models are compared in Fig. 8, normalized by the time taken by the algebraic model. It is shown that MD–cGAN–Full, MD–cGAN–Compact, and ResNet require a similar time cost but MD–cGAN–Full shows the biggest demand time. We found that the MD-cGAN models take ~50 times longer to perform inference than algebraic models. It is important to note that we performed the inference calculations on a GPU. The inference time for *a posteriori* analysis could be reduced by using central processing units (CPUs) instead. The differences in performance between GPU-based and CPU-based inference are discussed in depth in Ref. 98.

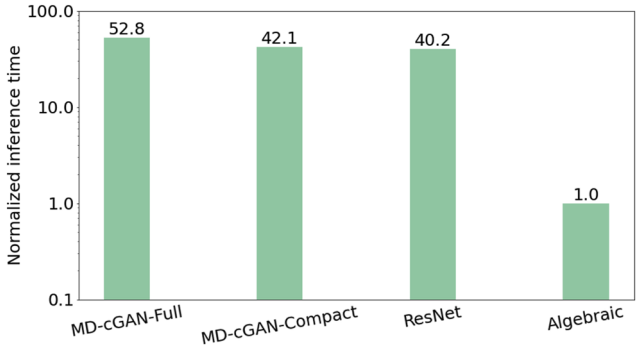

FIG. 8. Plot of inference time normalized by the time taken by the algebraic model. Note that the y-axis is in the logarithmic scale.

Figure 9 shows the comparison of the correlation plots of predicted $\bar{\dot{\omega}}$ against $\bar{\dot{\omega}}$ from the filtered DNS for the LES filter widths $\Delta/\delta_{th} \approx 1.14, 1.71, 2.29, 3.43, 4.57$. A hexbin plot has been used to show the density of the points onto the 100 × 100 discretized grid using a grey-colored scale. As shown in Fig. 7, scatters with higher variance are observed for large filter widths, for example, $\Delta/\delta_{th} \approx 3.43, 4.57$. In contrast, it displays a tight spread of the prediction for small filter widths, for example, $\Delta/\delta_{th} \approx 0.57, 1.14$, which was to be expected because, at these small filter sizes, there is a negligible subgrid flame structure present.

It is found that the MD–cGAN–Compact shows little difference in performance compared to MD–cGAN–Full. This fact implies that the information latent in the compact set of features, which is represented by the parameters $\tilde{c}$, $|\nabla\tilde{c}|$, $|\nabla^2\tilde{c}|$, u'_Δ, Δ, is dominant. In other words, the other parameters $|\tilde{u}_i|$, $|\nabla\tilde{u}_i|$, $|S_{ij}|$, $|\omega_{ij}|$, $\tilde{\kappa}$, and $\tilde{a}_T$ can be neglected to predict the filtered reaction rate $\bar{\dot{\omega}}$. This shows that feature importance analysis successfully reduces the number of input features without compromising on model accuracy.

In order to demonstrate the learning capability of the MD–cGAN–Compact qualitatively, slices of the $\bar{\dot{\omega}}$ field from the filtered DNS and predicted by the models are presented in Fig. 10. It is shown that MD–cGAN–Compact can reproduce the exact values as well as the geometrical structures of the flame front even though it is tested on the untrained LES filter widths, whereas the algebraic model predicts considerably lower values than the ground truth.

Further quantitative analysis to manifest the stochastic predictive ability of the proposed model has been conducted. We generated 100 sample fields using the MD–cGAN–Full and the MD–cGAN–Compact and then we computed their mean fields and variance fields, respectively.

In Fig. 11, a quantitative comparison is attempted by showing the total reaction rate of the mean field inside the total volume. It is also found that the proposed deep learning model performs well from this global point of view.

Despite the MD–cGAN–Compact model providing relatively good prediction capability, there is some deviation from ground truth when looking at the averaged profiles along an axial direction. Figures 12(a) and 12(b) show the comparison of the spatial evolution of both model predictions by plotting the mean $\bar{\dot{\omega}}$ values averaged on the transverse yz-plane as a function of the x-coordinate, generating 100 samples of predictions. The profiles inferred from the trained filter widths match the profiles from the filtered DNS well for both models. However, a deviation has been found in the estimation by the MD–cGAN–Compact for the untrained filter width $\Delta/\delta_{th} \approx 3.43$. It appears that the unused parameters in the compact

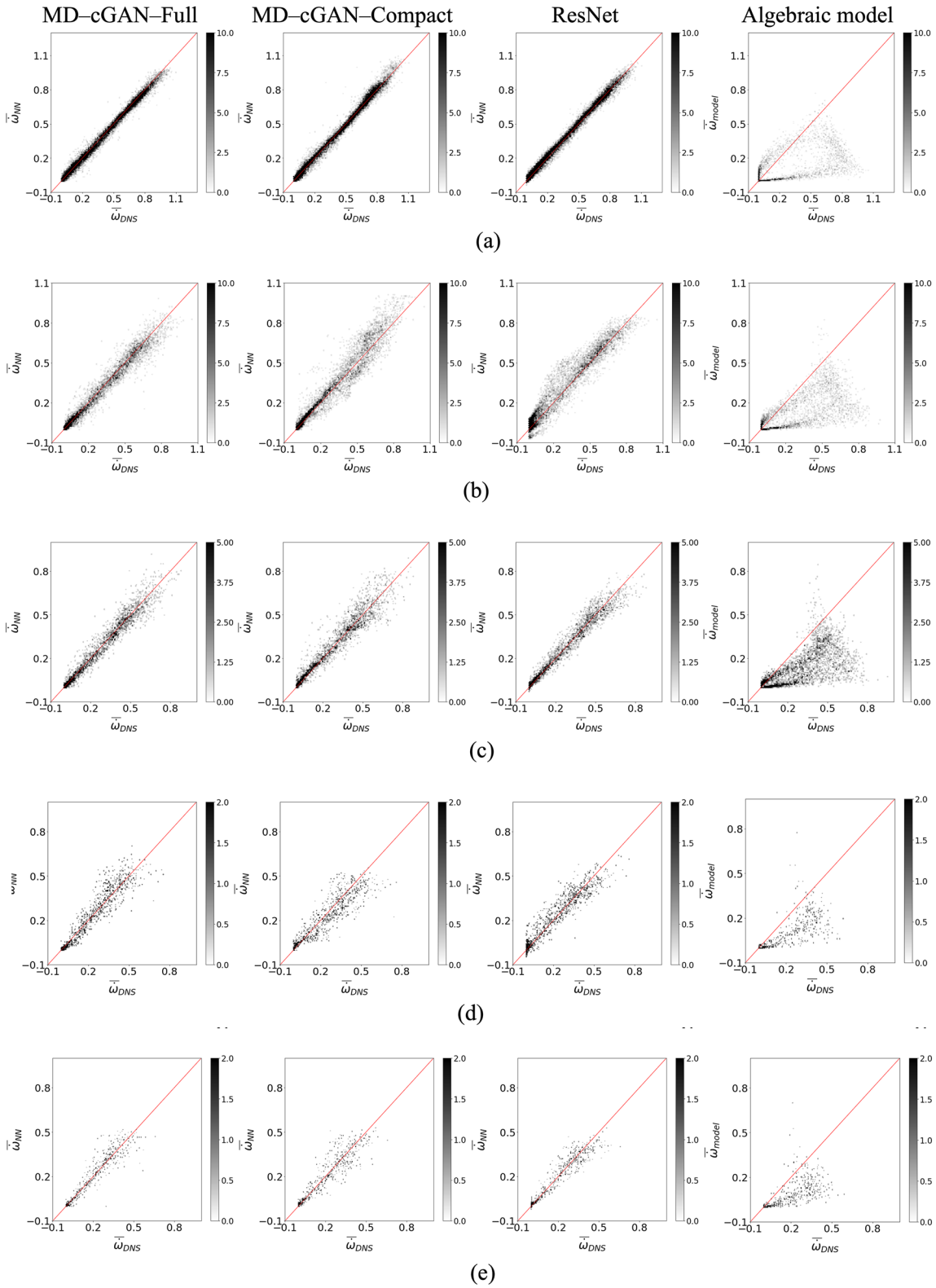

FIG. 9. Hexbin plots showing the correlation between the predicted filtered reaction rate $\overline{\omega}$ and $\overline{\omega}$ from filtered DNS for different filter widths. ※untrained LES filter width. (a) $\Delta/\delta_{th} = 1.14(n_\Delta = 8)$. (b) $\Delta/\delta_{th} = 1.71(n_\Delta = 12)$※. (c) $\Delta/\delta_{th} = 2.29(n_\Delta = 16)$. (d) $\Delta/\delta_{th} = 3.43(n_\Delta = 24)$※. (e) $\Delta/\delta_{th} = 4.57(n_\Delta = 32)$.

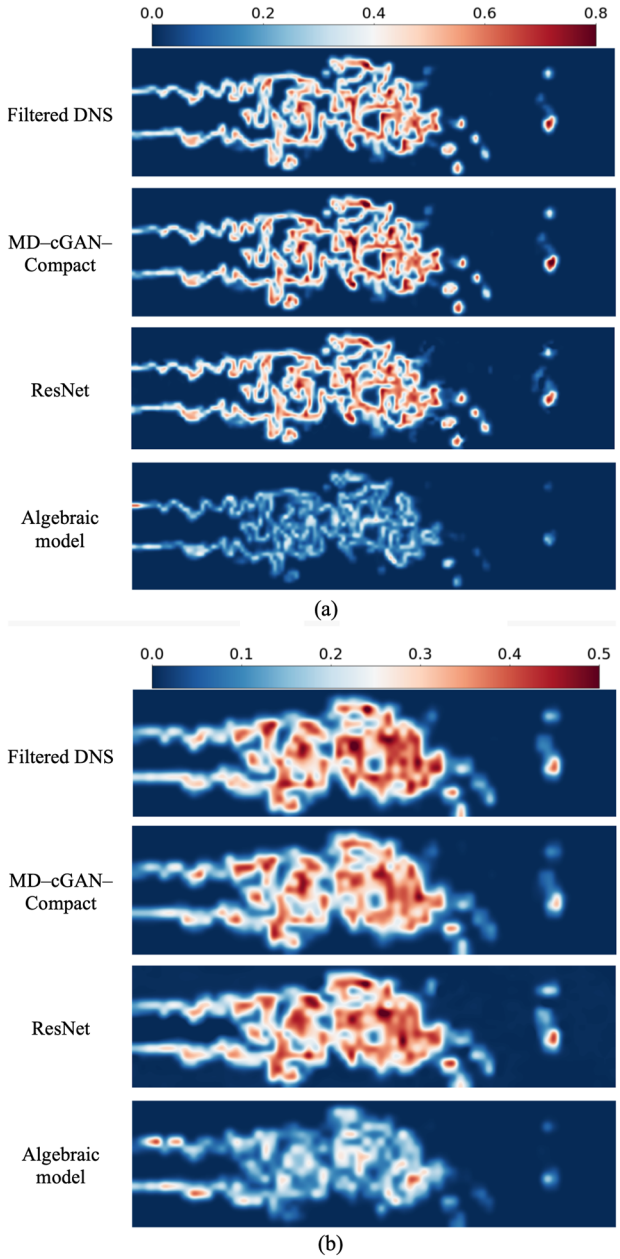

FIG. 10. Center slice view of true and predicted filtered reaction rate $\overline{\dot{\omega}}$ for the unseen time snapshot and untrained LES filter widths. (a) $\Delta/\delta_{th} = 1.71 (n_\Delta = 12)$. (b) $\Delta/\delta_{th} = 3.43 (n_\Delta = 24)$.

parameter set more accurately interpolate to the parameter space represented by the untrained filter widths. In addition, Fig. 12(c) illustrates the probabilistic nature of the proposed model, in which the statistical mean profile predicted by the MD–cGAN–Compact and its bandwidth of confidence are presented for the untrained filter width $\Delta/\delta_{th} \approx 3.43$, as an example. It shows that the range of the prediction can cover the profile computed from the filtered DNS.

Figure 13 depicts the density distributions of the mean $\overline{\dot{\omega}}$ fields predicted by the MD–cGAN–Compact together with its 95% confidence interval and the density distributions of the filtered DNS as a comparison. The reaction zone Z is selected such as to investigate only the regions where the flame is located. The region is chosen to cover only regions where $\overline{\dot{\omega}}$ is greater than 10% of its maximum value

$$Z = \left\{\overline{\dot{\omega}} \geq 0.1 \times \max\left(\overline{\dot{\omega}}\right)\right\}. \tag{12}$$

The same definition has been used to identify the reaction zone in Ref. 29. The MD–cGAN–Compact model exhibits a comparable magnitude of errors in the prediction of $\overline{\dot{\omega}} \in Z$, and the accuracy level of prediction for untrained filter widths is similar to the one for the trained filter widths. It can be noted that the stochastic spreads of the predictions increase as the LES filter width increases.

The comparison between MD–cGAN–Compact and ResNet reveals the benefits of the mixture density network. Figures 13(a) and 13(d) indicate that the distributions from ResNet are comparable to MD–cGAN–Compact at the trained LES filter widths. However, ResNet anticipates different tendencies of density distribution compared with filtered DNS with respect to the untrained LES filter widths in Figs. 13(b) and 13(c). Interestingly, ResNet, a deep neural network that is trying to estimate the probability density of the target parameter directly by itself, provides a comparable predictive power to the other methods for the training LES filter widths, whereas for the untrained LES filter widths, the predictive power significantly decreases. It is found that the probabilistic deep learning approach that models the joint distribution of the target parameter based on a linear combination of multivariate Gaussian distributions demonstrates an improvement in the generalization capability for the unseen data from untrained time snapshots and LES filter widths.

V. CONCLUSIONS

A probabilistic deep learning model has been investigated as a method of turbulent premixed combustion modeling in the context of large-eddy simulation (LES). A slot burner jet flame DNS has been used for the training database and the DNS is spatially filtered for a large range of filter widths. By combining a Gaussian mixture density model and a conditional generative adversarial network, the filtered reaction rate evaluated at different LES filter widths is predicted with very good accuracy. In addition, we used a feature importance analysis that allows us to reduce the number of input parameters to the model with no impact on its predictive ability, making the model simpler and cheaper to evaluate and easier to interpret physically. The generalization capability of the proposed deep learning model tested on an unseen time step and untrained filter widths was studied *a priori* and the proposed deep learning model was able to reproduce the filtered reaction rate source term distributions quite accurately.

We aim to further develop our model by testing and validating it with various DNSes that have different turbulent conditions and fuels, such as higher *Ka* numbers and non-unity Lewis number cases. We also plan to study the model's ability to generalize to different flame configurations through *a posteriori* analysis. Probabilistic machine learning models have not been widely used for LES

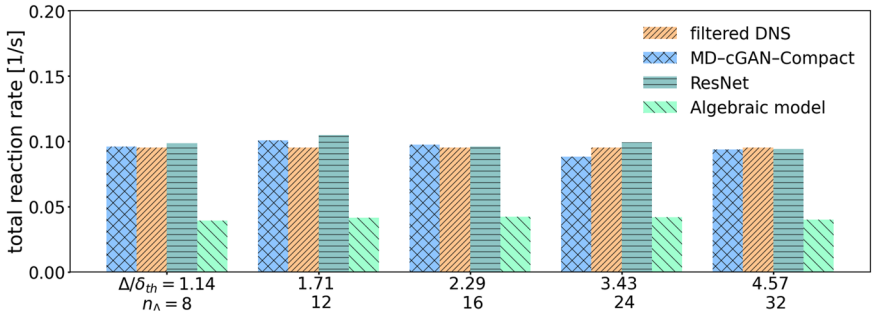

FIG. 11. Total reaction rate integrated inside the domain volume for different filter widths.

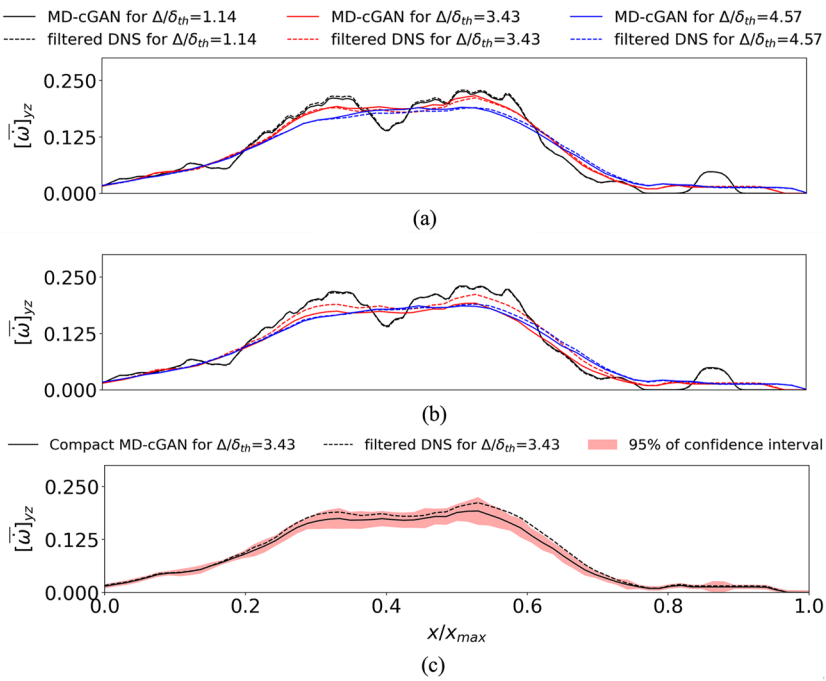

FIG. 12. Evolutions of the filtered reaction rate $\bar{\dot{\omega}}$ from the mean field (a), (b) and mean profile predicted by the MD–cGAN–Compact shown with its 95% confidence interval (c), averaged on the transverse plane along the streamwise x-direction. (a) MD–cGAN–Full. (b) MD–cGAN–Compact. (c) Predicted profiles for $\Delta/\delta_{th} = 3.43$ ($n_\Delta = 24$).

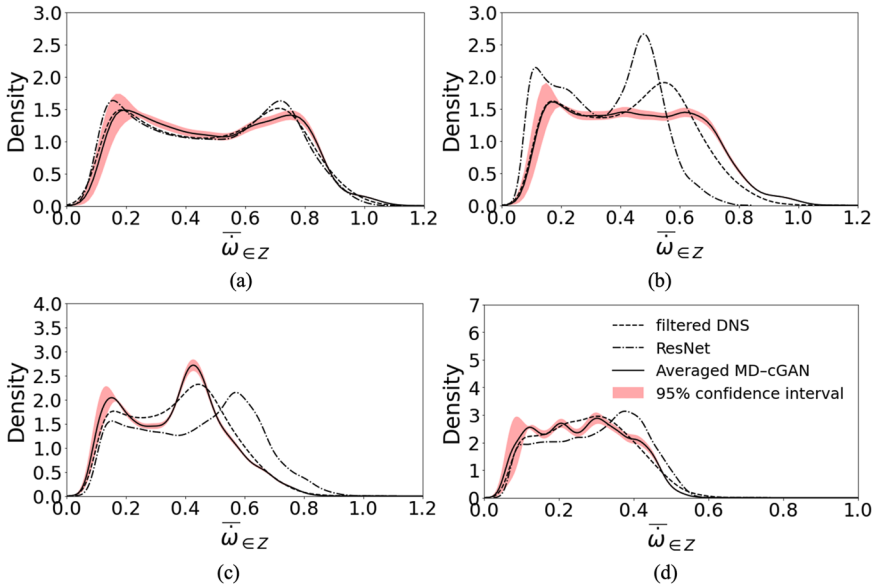

FIG. 13. Comparison of distribution density of the filtered reaction rate $\bar{\dot{\omega}}$ for different filter widths computed from the filtered DNS (dashed), the MD–cGAN–Compact (solid) and its 95% confidence interval (shade), and ResNet (dashed dot). ※ untrained LES filter width. (a) $\Delta/\delta_{th} = 1.14 (n_\Delta = 8)$ (b) $\Delta/\delta_{th} = 1.71$ $(n_\Delta = 12)$※ (c) $\Delta/\delta_{th} = 3.43 (n_\Delta = 24)$※ (d) $\Delta/\delta_{th} = 4.57$ $(n_\Delta = 32)$.

TABLE V. Mean squared error (MSE) of the models for different filter widths.

MSE	MD–cGAN–Full	MD–cGAN–Compact	ResNet	Algebraic model
$\Delta/\delta_{th} = 0.57 (n_\Delta = 4)$	1.540×10^{-4}	1.196×10^{-4}	1.882×10^{-4}	4.516×10^{-2}
$\Delta/\delta_{th} = 1.14 (n_\Delta = 8)$	2.317×10^{-4}	2.405×10^{-4}	2.638×10^{-4}	2.934×10^{-2}
$\Delta/\delta_{th} = 1.71 (n_\Delta = 12)$[a]	7.714×10^{-4}	1.447×10^{-3}	7.086×10^{-4}	2.041×10^{-2}
$\Delta/\delta_{th} = 2.29 (n_\Delta = 16)$	7.484×10^{-4}	1.185×10^{-3}	7.589×10^{-4}	1.643×10^{-2}
$\Delta/\delta_{th} = 3.43 (n_\Delta = 24)$[a]	1.498×10^{-3}	2.221×10^{-3}	1.963×10^{-3}	1.394×10^{-2}
$\Delta/\delta_{th} = 4.57 (n_\Delta = 32)$	1.119×10^{-3}	1.578×10^{-3}	1.114×10^{-3}	1.345×10^{-2}
$\Delta/\delta_{th} = 9.14 (n_\Delta = 64)$	3.916×10^{-4}	1.072×10^{-3}	1.081×10^{-3}	1.414×10^{-2}

[a]untrained LES filter width.

modeling, but they could provide powerful ways to capture the statistical nature of SGS variables. In the future, we could explore using probabilistic models to represent filtered density functions in PDF methods. In addition, it is important to investigate the uncertainty quantification of these models to improve their interpretability.

ACKNOWLEDGMENTS

This research is funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr which we gratefully acknowledge. dtec.bw is funded by the European Union – NextGenerationEU. We also gratefully acknowledge the financial support from Deutsche Forschungsgemeinschaft (DFG) through DFG-Project Nos. PF443-9/1 and KL1456/5-1. The authors are also thankful to ITIS for financial support.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Junsu Shin: Conceptualization (lead); Data curation (equal); Formal analysis (lead); Investigation (lead); Methodology (lead); Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead); Writing – review & editing (lead). **Victor Xing**: Conceptualization (supporting); Data curation (equal); Resources (lead); Writing – original draft (supporting). **Michael Pfitzner**: Funding acquisition (lead); Supervision (equal). **Corentin Lapeyre**: Data curation (supporting); Supervision (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

APPENDIX A: ALGEBRAIC FSD MODELS

We employ an FSD model proposed by Fureby[71] and modified in Ref. 70. This algebraically formulated FSD model reads

$$\begin{aligned}
\overline{\dot{\omega}}_{Alg} &= \Xi |\nabla \tilde{c}| - \frac{\partial}{\partial x_k}\left(\rho D \frac{\partial c}{\partial x_k}\right), \\
\Xi &= \left(1 + \frac{\Delta}{\varepsilon_i}\right)^{D-2}, \ \varepsilon_i = \frac{s_L \Delta}{u'_\Delta \Gamma}, \\
\Gamma &= 0.75 \cdot \exp\left[-1.2\left(\frac{u'_\Delta}{s_L}\right)^{-0.3}\right] \cdot \left(\frac{\Delta}{\delta_L}\right)^{2/3}, \\
D &= \frac{2.05}{u'_\Delta / s_L + 1} + \frac{2.35}{s_L / u'_\Delta + 1},
\end{aligned} \tag{A1}$$

where the LES filter width is Δ. In order to compute the filtered reaction rate only, the diffusion term filtered from the DNS data is subtracted from the model expression.

APPENDIX B: ACTIVATIONS FUNCTIONS USED

In this paper, we use the following activation functions:

$$\begin{aligned}
ReLU(x_i) &= \max(0, x_i), \\
softma(x_i) &= \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}, \\
nnelu(x_i) &= 1 + \begin{cases} x_i, & \text{if } x_i \geq 0, \\ e^{x_i} - 1, & \text{if } x_i < 0, \end{cases}
\end{aligned} \tag{B1}$$

where the input vector to these functions is represented by x_i.

REFERENCES

[1]T. Nilsson, H. Carlsson, R. Yu, and X.-S. Bai, "Structures of turbulent premixed flames in the high Karlovitz number regime – DNS analysis," Fuel **216**, 627–638 (2018).
[2]J. Driscoll, "Turbulent premixed combustion: Flamelet structure and its effect on turbulent burning velocities," Prog. Energy Combust. Sci. **34**(1), 91–134 (2008).
[3]S. Luca, A. Attili, E. lo Schiavo, F. Creta, and F. Bisetti, "On the statistics of flame stretch in turbulent premixed jet flames in the thin reaction zone regime at varying Reynolds number," Proc. Combust. Inst. **37**(2), 2451–2459 (2019).
[4]B. Fiorina, R. Vicquelin, P. Auzillon, N. Darabiha, O. Gicquel, and D. Veynante, "A filtered tabulated chemistry model for LES of premixed combustion," Combust. Flame **157**(3), 465–475 (2010).
[5]M. Pfitzner, J. Shin, and M. Klein, "A multidimensional combustion model for oblique, wrinkled premixed flames," Combust. Flame **241**, 112121 (2022).
[6]L. Berger, A. Attili, and H. Pitsch, "Synergistic interactions of thermodiffusive instabilities and turbulence in lean hydrogen flames," Combust. Flame **244**, 112254 (2022).

[7]P. E. Lapenna *et al.*, "Data-driven subfilter modelling of thermo-diffusively unstable hydrogen–air premixed flames," Combust. Theory Modell. **25**(6), 1064–1085 (2021).
[8]J. D. Regele, E. Knudsen, H. Pitsch, and G. Blanquart, "A two-equation model for non-unity Lewis number differential diffusion in lean premixed laminar flames," Combust. Flame **160**(2), 240–250 (2013).
[9]S. B. Pope, *Turbulent Flows* (Cambridge University Press, 2000).
[10]V. Raman and H. Pitsch, "A consistent LES/filtered-density function formulation for the simulation of turbulent flames with detailed chemistry," Proc. Combust. Inst. **31**(2), 1711–1719 (2007).
[11]P. J. Colucci, F. A. Jaberi, P. Givi, and S. B. Pope, "Filtered density function for large eddy simulation of turbulent reacting flows," Phys. Fluids **10**(2), 499–515 (1998).
[12]D. C. Haworth, "Progress in probability density function methods for turbulent reacting flows," Prog. Energy Combust. Sci. **36**(2), 168–259 (2010).
[13]X. Zhang and A. Natarajan, "Gaussian mixture model for extreme wind turbulence estimation," Wind Energy Sci. **7**(5), 2135–2148 (2022).
[14]P. Jia *et al.*, "Modelling synthetic atmospheric turbulence profiles with temporal variation using Gaussian mixture model," Mon. Not. R. Astron. Soc. **480**(2), 2466–2474 (2018).
[15]H. Klem, G. M. Hocky, and M. McCullagh, "Size-and-shape space Gaussian mixture models for structural clustering of molecular dynamics trajectories," J. Chem. Theory Comput. **18**(5), 3218–3230 (2022).
[16]H. Wu, W. Chen, and Z. Jiang, "Gaussian mixture models for diatomic gas–surface interactions under thermal non-equilibrium conditions," Phys. Fluids **34**(8), 082007 (2022).
[17]M. Akram, M. Hassanaly, and V. Raman, "An approximate inertial manifold (AIM) based closure for turbulent flows," AIP Adv. **12**(7), 075118 (2022).
[18]A. A. Mishra, K. Duraisamy, and G. Iaccarino, "Estimating uncertainty in homogeneous turbulence evolution due to coarse-graining," Phys. Fluids **31**(2), 025106 (2019).
[19]P. Trisjono and H. Pitsch, "Systematic analysis strategies for the development of combustion models from DNS: A review," Flow, Turbul. Combust. **95**(2–3), 231–259 (2015).
[20]J. A. Langford and R. D. Moser, "Optimal LES formulations for isotropic turbulence," J. Fluid Mech. **398**, 321–346 (1999).
[21]S. Yellapantula, B. A. Perry, and R. W. Grout, "Deep learning-based model for progress variable dissipation rate in turbulent premixed flames," Proc. Combust. Inst. **38**(2), 2929–2938 (2021).
[22]S. Yao, B. Wang, A. Kronenburg, and O. T. Stein, "Modeling of sub-grid conditional mixing statistics in turbulent sprays using machine learning methods," Phys. Fluids **32**(11), 115124 (2020).
[23]M. T. Henry de Frahan, S. Yellapantula, R. King, M. S. Day, and R. W. Grout, "Deep learning for presumed probability density function models," Combust. Flame **208**, 436–450 (2019).
[24]Z. X. Chen *et al.*, "Application of machine learning for filtered density function closure in MILD combustion," Combust. Flame **225**, 160–179 (2021).
[25]C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinsot, "Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates," Combust. Flame **203**, 255–264 (2019).
[26]J. Shin, Y. Ge, A. Lampmann, and M. Pfitzner, "A data-driven subgrid scale model in large eddy simulation of turbulent premixed combustion," Combust. Flame **231**, 111486 (2021).
[27]J. Ren, H. Wang, K. Luo, and J. Fan, "A priori assessment of convolutional neural network and algebraic models for flame surface density of high Karlovitz premixed flames," Phys. Fluids **33**(3), 036111 (2021).
[28]Z. M. Nikolaou, C. Chrysostomou, L. Vervisch, and S. Cant, "Progress variable variance and filtered rate modelling using convolutional neural networks and flamelet methods," Flow, Turbul. Combust. **103**(2), 485–501 (2019).
[29]V. Xing, C. Lapeyre, T. Jaravel, and T. Poinsot, "Generalization capability of convolutional neural networks for progress variable variance and reaction rate subgrid-scale modeling," Energies **14**(16), 5096 (2021).
[30]A. Seltz, P. Domingo, L. Vervisch, and Z. M. Nikolaou, "Direct mapping from LES resolved scales to filtered-flame generated manifolds using convolutional neural networks," Combust. Flame **210**, 71–82 (2019).
[31]K. Wan, C. Barnaud, L. Vervisch, and P. Domingo, "Chemistry reduction using machine learning trained from non-premixed micro-mixing modeling: Application to DNS of a syngas turbulent oxy-flame with side-wall effects," Combust. Flame **220**, 119–129 (2020).
[32]J. Shin, M. Hansinger, M. Pfitzner, and M. Klein, "A priori analysis on deep learning of filtered reaction rate," Flow, Turbul. Combust. **109**, 383 (2022).
[33]C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2015), pp. 1–9.
[34]K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2016), pp. 770–778.
[35]K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Curran Associates, Inc., 2014).
[36]M. Bode *et al.*, "Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows," Proc. Combust. Inst. **38**(2), 2617–2625 (2021).
[37]M. Ihme, W. T. Chung, and A. A. Mishra, "Combustion machine learning: Principles, progress and prospects," Prog. Energy Combust. Sci. **91**, 101010 (2022).
[38]I. Goodfellow *et al.*, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, edited by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger (Curran Associates, Inc., 2014).
[39]S. Lee and D. You, "Data-driven prediction of unsteady flow over a circular cylinder using deep learning," J. Fluid Mech. **879**, 217–254 (2019).
[40]M. Hassanaly, A. Glaws, K. Stengel, and R. N. King, "Adversarial sampling of unknown and high-dimensional conditional distributions," J. Comput. Phys. **450**, 110853 (2022).
[41]K. Fukami, R. Maulik, N. Ramachandra, K. Fukagata, and K. Taira, "Probabilistic neural network-based reduced-order surrogate for fluid flows," arXiv:2012.08719 (2020).
[42]M. Pfitzner, "A new analytic pdf for simulations of premixed turbulent combustion," Flow, Turbul. Combust. **106**(4), 1213–1239 (2021).
[43]M. Pfitzner and M. Klein, "A near-exact analytic solution of progress variable and pdf for single-step Arrhenius chemistry," Combust. Flame **226**, 380–395 (2021).
[44]M. Pfitzner and P. Breda, "An analytic probability density function for partially premixed flames with detailed chemistry," Phys. Fluids **33**(3), 035117 (2021).
[45]N. Chakraborty and S. Cant, "Unsteady effects of strain rate and curvature on turbulent premixed flames in an inflow–outflow configuration," Combust. Flame **137**(1–2), 129–147 (2004).
[46]N. Chakraborty, M. Klein, and R. S. Cant, "Stretch rate effects on displacement speed in turbulent premixed flame kernels in the thin reaction zones regime," Proc. Combust. Inst. **31**(1), 1385–1392 (2007).
[47]N. Chakraborty, D. Alwazzan, M. Klein, and R. S. Cant, "On the validity of Damköhler's first hypothesis in turbulent Bunsen burner flames: A computational analysis," Proc. Combust. Inst. **37**(2), 2231–2239 (2019).
[48]J. Lai, M. Klein, and N. Chakraborty, "Direct numerical simulation of head-on quenching of statistically planar turbulent premixed methane-air flames using a detailed chemical mechanism," Flow, Turbul. Combust. **101**(4), 1073–1091 (2018).
[49]G. P. Smith *et al.*, Gri-mech 3.0, http://combustion.berkeley.edu/gri-mech/, 1999.
[50]T. Schoenfeld and M. Rudgyard, "Steady and unsteady flow simulations using the hybrid flow solver AVBP," AIAA J. **37**(11), 1378–1385 (1999).
[51]L. Selle *et al.*, "Compressible large eddy simulation of turbulent combustion in complex geometry on unstructured meshes," Combust. Flame **137**(4), 489–505 (2004).
[52]O. Colin and M. Rudgyard, "Development of high-order Taylor–Galerkin schemes for LES," J. Comput. Phys. **162**(2), 338–371 (2000).
[53]T. J. Poinsot and S. K. Lelef, "Boundary conditions for direct simulations of compressible viscous flows," J. Comput. Phys. **101**(1), 104–129 (1992).
[54]R. H. Kraichnan, "Diffusion by a random velocity field," Phys. Fluids **13**(1), 22 (1970).

[55]T. Passot and A. Pouquet, "Numerical simulation of compressible homogeneous flows in the turbulent regime," J. Fluid Mech. **181**(1), 441 (1987).
[56]V. A. Sabelnikov *et al.*, "Dissipation and dilatation rates in premixed turbulent flames," Phys. Fluids **33**(3), 035112 (2021).
[57]N. Peters, *Turbulent Combustion* (Cambridge University Press, 2000).
[58]R. Borghi, "On the structure and morphology of turbulent premixed flames," *Recent Advances in the Aerospace Sciences* (Springer, 1985), pp. 117–138.
[59]R. Sankaran, E. R. Hawkes, C. S. Yoo, and J. H. Chen, "Response of flame thickness and propagation speed under intense turbulence in spatially developing lean premixed methane–air jet flames," Combust. Flame **162**(9), 3294–3306 (2015).
[60]H. Wang, E. R. Hawkes, J. H. Chen, B. Zhou, Z. Li, and M. Aldén, "Direct numerical simulations of a high Karlovitz number laboratory premixed jet flame—An analysis of flame stretch and flame thickening," J. Fluid Mech. **815**, 511–536 (2017).
[61]M. Klein, H. Nachtigal, M. Hansinger, M. Pfitzner, and N. Chakraborty, "Flame curvature distribution in high pressure turbulent Bunsen premixed flames," Flow, Turbul. Combust. **101**(4), 1173–1187 (2018).
[62]R. S. Rogallo and P. Moin, "Numerical simulation of turbulent flows," Annu. Rev. Fluid. Mech. **16**(1), 99–137 (1984).
[63]A. Leonard, "Energy cascade in large-eddy simulations of turbulent fluid flows," Adv. Geophys. **18**, 237–248 (1975).
[64]M. Klein, N. Chakraborty, and S. Ketterl, "A comparison of strategies for direct numerical simulation of turbulence chemistry interaction in generic planar turbulent premixed flames," Flow, Turbul. Combust. **99**(3–4), 955–971 (2017).
[65]T. Echekki and J. H. Chen, "Unsteady strain rate and curvature effects in turbulent premixed methane-air flames," Combust. Flame **106**(1–2), 184–202 (1996).
[66]D. N. Reshef *et al.*, "Detecting novel associations in large data sets," Science **334**(6062), 1518–1524 (2011).
[67]S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in Neural Information Processing Systems* edited by I. Guyon, U. von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017).
[68]J. Wang, X. He, B. Wang, and X. Zheng, "Shapley additive explanations of multigeometrical variable coupling effect in transonic compressor," J. Eng. Gas Turbines Power **144**(4), 041015 (2022).
[69]J. Heyse, A. Mishra, and G. Iaccarino, "Data driven and physics constrained perturbations for turbulence model uncertainty estimation," in *AAAI 2021 Spring Symposium on Combining Artificial Intelligence and Machine Learning with Physics Sciences* (AAAI, 2021), pp. 1–7.
[70]T. Ma, O. T. Stein, N. Chakraborty, and A. M. Kempf, "A posteriori testing of algebraic flame surface density models for LES," Combust. Theory Modell. **17**(3), 431–482 (2013).
[71]C. Fureby, "A fractal flame-wrinkling large eddy simulation model for premixed turbulent combustion," Proc. Combust. Inst. **30**(1), 593–601 (2005).
[72]M. Schoepplein, J. Weatheritt, R. Sandberg, M. Talei, and M. Klein, "Application of an evolutionary algorithm to LES modelling of turbulent transport in premixed flames," J. Comput. Phys. **374**, 1166–1179 (2018).
[73]D. J. Hand, G. J. McLachlan, and K. E. Basford, "Mixture models: Inference and applications to clustering," Appl. Stat. **38**(2), 384 (1989).
[74]C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, 2006).
[75]M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv:1411.1784 (2014).
[76]J. Oskarsson, "Probabilistic regression using conditional generative adversarial networks," Linköping University, 2020.
[77]Y. Boget, "Adversarial regression. Generative adversarial networks for non-linear regression: Theory and assessment," arXiv:1910.09106 (2019).
[78]Y.-L. Min, S.-J. Hong, H. Kim, and S.-I. Lee, "*Generative adversarial network for robust regression using continuous dataset*," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)* (IEEE, 2020), pp. 1209–1211.
[79]K. Aggarwal, M. Kirchmeyer, P. Yadav, S. S. Keerthi, and P. Gallinari, "Benchmarking regression methods: A comparison with CGAN," arXiv:1905.12868 (2019).
[80]N. Kodali, J. Abernethy, J. Hays, and Z. Kira, "On convergence and stability of GANs," arXiv:1705.07215 (2017).
[81]H. Thanh-Tung and T. Tran, "Catastrophic forgetting and mode collapse in GANs," in *2020 International Joint Conference on Neural Networks (IJCNN)* (IEEE, 2020), pp. 1–10.
[82]S. Lala, M. Shady, A. Belyaeva, and M. Liu, "Evaluation of mode collapse in generative adversarial networks," in *2018 IEEE High Performance Extreme Computing Conference* (IEEE, 2018), pp. 1–9.
[83]T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *30th Conference on Neural Information Processing Systems (NIPS)* (Neural Information Processing Systems, 2016).
[84]M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," arXiv:1701.07875 (2017).
[85]I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," arXiv:1704.00028 (2017).
[86]X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," arXiv:1611.04076 (2016).
[87]X. Zhu, Y. Liu, Z. Qin, and J. Li, "Data augmentation in emotion classification using generative adversarial networks," arXiv:1711.00648 (2017).
[88]C. Wang, Y. Cao, S. Zhang, and T. Ling, "A reconstruction method for missing data in power system measurement based on LSGAN," Front. Energy Res. **9**, 1–13 (2021).
[89]M. Faraz and H. Khaloozadeh, "*Multi-step-ahead stock market prediction based on least squares generative adversarial network*," in *2020 28th Iranian Conference on Electrical Engineering (ICEE)* (IEEE, 2020), pp. 1–6.
[90]D. P. Kingma and J. Ba, "Adam: A method for stochastic Optimization," arXiv:1412.6980 (2014).
[91]V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning* (Omnipress, 2010), pp. 807–814.
[92]J. S. Bridle, "Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition," *Neurocomputing* (Springer, Berlin, Heidelberg, 1990), pp. 227–236.
[93]R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," Neural Comput. **3**(1), 79–87 (1991).
[94]D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," arXiv:1511.07289 (2015).
[95]M. Wiatrak, S. V. Albrecht, and A. Nystrom, "Stabilizing generative adversarial networks: A survey," arXiv:1910.00927 (2019).
[96]O. Duerr, B. Sick, and E. Murina, *Probabilistic Deep Learning: With Python, Keras and TensorFlow Probability* (Simon & Schuster, 2020).
[97]M. Abadi *et al.*, "TensorFlow: Large-Scale machine learning on heterogeneous distributed systems," arXiv:1603.04467 (2016).
[98]E. Trautner, "Deep neural networks for combustion modeling: Optimizing CPU-based inference for OpenFOAM," M.S. thesis, Ostbayerische Technische Hochschule, Regensburg, 2019.