



**HAL**  
open science

## Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates

Corentin J Lapeyre, Antony Misdariis, Nicolas Cazard, Denis Veynante,  
Thierry Poinsot

► **To cite this version:**

Corentin J Lapeyre, Antony Misdariis, Nicolas Cazard, Denis Veynante, Thierry Poinsot. Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame*, 2019, 203, pp.255-264. 10.1016/j.combustflame.2019.02.019 . hal-02072920

**HAL Id: hal-02072920**

**<https://hal.science/hal-02072920v1>**

Submitted on 19 Mar 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.




OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible

This is an author's version published in: <http://oatao.univ-toulouse.fr/23414>

**Official URL :** <https://doi.org/10.1016/j.combustflame.2019.02.019>

**To cite this version:**

Lapeyre, Corentin J. and Misdariis, Antony and Cazard, Nicolas and Veynante, Denis and Poinot, Thierry  *Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates.* (2019) *Combustion and Flame*, 203. 255-264. ISSN 0010-2180

Any correspondence concerning this service should be sent to the repository administrator: [tech-oatao@listes-diff.inp-toulouse.fr](mailto:tech-oatao@listes-diff.inp-toulouse.fr)

# Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates

Corentin J. Lapeyre<sup>a,\*</sup>, Antony Misdariis<sup>a</sup>, Nicolas Cazard<sup>a</sup>, Denis Veynante<sup>b</sup>,  
Thierry Poinsot<sup>c</sup>

<sup>a</sup>CERFACS, 42 avenue Gaspard Coriolis, Toulouse 31057, France

<sup>b</sup>EM2C, CNRS, Centrale-Supélec, Université Paris-Saclay, 3 rue Joliot-Curie, Gif-sur-Yvette 91192, France

<sup>c</sup>IMFT, Allée du Professeur Camille Soula, Toulouse 31400, France

## Keywords:

Turbulent combustion

Deep learning

Flame surface density

Direct numerical simulation

This work presents a new approach for premixed turbulent combustion modeling based on convolutional neural networks (CNN).<sup>1</sup> We first propose a framework to reformulate the problem of subgrid flame surface density estimation as a machine learning task. Data needed to train the CNN is produced by direct numerical simulations (DNS) of a premixed turbulent flame stabilized in a slot-burner configuration. A CNN inspired from a U-Net architecture is designed and trained on the DNS fields to estimate subgrid-scale wrinkling. It is then tested on an unsteady turbulent flame where the mean inlet velocity is increased for a short time and the flame must react to a varying turbulent incoming flow. The CNN is found to efficiently extract the topological nature of the flame and predict subgrid-scale wrinkling, outperforming classical algebraic models.

## 1. Introduction

Deep Learning (DL) [1] is a machine learning strategy at the center of a strong hype in many digital industries. This popularity stems in part from the capacity of this approach to sift efficiently through high-dimensional data inherent in *real world* applications. In conjunction with so-called *Big Data*, or the access to sensing, storage and computing capabilities that yield huge databases to learn from, some challenges *e.g.* in computer vision [2], natural language processing [3] and complex game playing [4] have seen dramatic advancements in the past decade.

Originally developed as a model of the mammal brain [5], Artificial Neural Networks (ANN) have since been optimized for numerical performance, enabling the training of deeper architectures, and eventually putting them at the center of the DL effort. These developments have been traditionally lead by experts in computer cognition, limiting their application to select fields. Modern programming frameworks with high levels of abstraction [6] have however been made available in the past 3 years, in conjunction with powerful hardware such as GPUs to perform fast training. This

has opened the possibility for applications in many other fields, such as physics, where the *causal* nature of DL [7] suggests that complex patterns could also be sought and learned.

DL clearly belongs to methods devoted to the analysis of *data*. In the field of fluid mechanics and of combustion, where *models* *i.e.* the Navier–Stokes equations are known, evaluating the possible impacts of DL is difficult. In this area, what is obviously needed is a mixed *models/data* approach. Data-driven strategies are by nature approximations, suggesting significant challenges when used on problems for which deterministic equations are available. The low hanging fruits are therefore expected to be sub-problems where models do not rely on exact equations but on simple closure assumptions. In this field, DL may work better than standard models, notably when the flow topology is known to inform the estimation.

Recent studies applied to turbulent flows [8–12] have shown that subgrid-scale (SGS) closure models for Reynolds averaged Navier–Stokes (RANS) and Large Eddy Simulation (LES) could be addressed using several machine learning algorithms, including shallow ANNs. However, advancements offered by DL methods have in large part stemmed from pattern recognition performed by *deep* ANNs [1], which are still mostly absent from the fluid mechanics literature, as shown in a recent review [13]. Convolutional Neural Networks (CNNs) are interesting compared to ANNs because they introduce the notion of *parameter sharing*: instead of having to learn the relationships between input and output everywhere sep-

\* Corresponding author.

E-mail address: [lapeyre@cerfacs.fr](mailto:lapeyre@cerfacs.fr) (C.J. Lapeyre).

<sup>1</sup> Code and data for the deep learning in this work is available at [https://gitlab.com/cerfacs/code-for-papers/2018/arXiv\\_1810.03691](https://gitlab.com/cerfacs/code-for-papers/2018/arXiv_1810.03691).

arately, CNNs learn small filters that apply over the entire image. This reduces the number of connections (hence of learnable parameters) per layer, and offers the possibility to stack many layer efficiently. Additionally, on Nevertheless, some deep residual networks have been built, and it was shown that they could accurately recover state-of-the-art turbulent viscosity models on homogeneous isotropic turbulence [14].

In the combustion community, the determination of the SGS contribution to the filtered reaction rate in reacting flows LES is an example of closure problem that has been daunting for a long time. Indeed, SGS interactions between the flame and turbulent scales largely determines the flame behavior, and modeling them is an important factor to obtain overall flame dynamics. Many turbulent modeling approaches are based on a reconstruction of the SGS wrinkling of the flame surface and the so-called *flamelet* assumption [15]. Under this assumption, the mean turbulent reaction rate can be expressed in terms of *flame surface area* [16,17]. Indeed, the idea that turbulence convects, deforms and spreads surfaces [18] can be applied to a premixed flame front in a turbulent flow. The evaluation of the amount of flame surface area due to unresolved flame wrinkling is the core of all models based on flame surface areas in the last 50 years [15], both for RANS [19–22] and LES [23,24]. CNNs could be a natural fit for this task, which consists in recognizing geometrical topologies of the flame, and learning to associate an under-resolved wrinkling level to each. This is akin to many of the recognition challenges involved in image recognition, one of the domains that CNNs are known to excel at.

This paper explores this question and proposes *a priori* tests of a *deep* CNN-based model for the SGS contribution to the reaction rate of premixed turbulent flames. It is organized as follows: in Section 2, the theoretical aspects of the study are presented. They are inspired from the context of flame surface density models, but are reformulated in the framework of machine learning algorithms. Section 3 describes the DNS performed to produce the data needed to train the neural network. Section 4 describes the design, implementation and training procedure of a CNN for the flame surface density estimation problem at hand. The data produced in the previous section is used to train a CNN. Once the training process has converged, this network is frozen into a function that is used on new fields to predict flame surface density in Section 5. The method is meant to be used in this fashion: once the training has been performed using DNS data, no additional DNS is needed to use the model on new configurations. In the last section, the accuracy of the trained network is compared to several classical models from the literature, and the specific challenges of evaluating learning approaches are discussed.

## 2. Theoretical modeling

### 2.1. Flame surface density models

LES relies on a spatial filtering to split the turbulence spectrum and remove the non-resolved scales. For each quantity of interest  $Q$  from a well resolved flow field, the low-pass spatial filter  $F_\Delta$  with width  $\Delta$  yields:

$$\overline{Q}(\mathbf{x}, t) = \int_{\mathcal{V}} F_\Delta(\mathbf{x} - \mathbf{x}') Q(\mathbf{x}', t) d\mathbf{x}' \quad (1)$$

where  $\overline{\cdot}$  denotes the filtering operation. We will limit this study to perfectly premixed combustion where a progress variable  $c$  for adiabatic flows is defined as:

$$c = \frac{T - T_u}{T_b - T_u} \quad (2)$$

with subscripts  $u$  and  $b$  referring to unburnt and burnt gases, respectively. A balance equation can be written for  $c$  [15], by defining

a density weighted (or Favre) filtering  $\tilde{Q} = \overline{\rho Q} / \overline{\rho}$  for every quantity  $Q$ . Filtering the progress variable equation written in a propagative form (G-equation, [25]) assuming locally flame elements gives [24]:

$$\frac{\partial \tilde{\rho} \tilde{c}}{\partial t} + \nabla \cdot (\tilde{\rho} \tilde{\mathbf{u}} \tilde{c}) + \nabla \cdot (\tilde{\rho} \tilde{\mathbf{u}} \tilde{c} - \tilde{\rho} \tilde{\mathbf{u}} \tilde{c}) = \rho_u S_L^0 \overline{\Sigma} \quad (3)$$

where the right hand side term incorporates filtered diffusion and reaction terms into a single  $c$ -isosurface displacement speed assimilated to laminar flame speed  $S_L^0$ , and where  $\rho_u$  is the fresh gases density.  $\overline{\Sigma} = |\overline{\nabla c}|$  is the generalized *flame surface density* [23], and cannot be obtained in general from resolved flame surfaces. Indeed, when filtering  $c$  to  $\tilde{c}$ , surface wrinkling decreases, resulting in less total  $c$ -isosurface. One popular method to model  $\overline{\Sigma}$  is to introduce the *wrinkling factor*  $\Xi$  that compares the total and resolved generalized flame surfaces. The right-hand side term of Eq. (3) is then rewritten as:

$$\rho_u S_L^0 \overline{\Sigma} = \rho_u S_L^0 \Xi |\nabla \tilde{c}| \quad (4)$$

$$\text{where } \Xi = \frac{\overline{\Sigma}}{|\nabla \tilde{c}|} \quad (5)$$

Fractal approaches such as introduced by Gouldin et al. [26] suggest a relationship between  $\overline{\Sigma}$  and  $|\nabla \tilde{c}|$  of the form:

$$\overline{\Sigma} = \left( \frac{\Delta}{\eta_c} \right)^{D_f - 2} |\nabla \tilde{c}| \quad (6)$$

where  $D_f$  is the fractal dimension of the flame surface, and  $\eta_c$  is the inner cutoff scale below which the flame is no longer wrinkled. The  $\eta_c$  length scales with the laminar flame thickness  $\delta_L^0$  [27,28].

More recent work, based on flame/vortex interactions and multi-fractal analysis [29] suggests a different form (modified to recover Eq. (6) at saturation [30]):

$$\overline{\Sigma} = \left( 1 + \min \left[ \frac{\Delta}{\delta_L^0} - 1, \Gamma_\Delta \left( \frac{\Delta}{\delta_L^0}, \frac{u'_\Delta}{S_L^0}, Re_\Delta \right) \frac{u'_\Delta}{S_L^0} \right] \right)^\beta |\nabla \tilde{c}| \quad (7)$$

where  $\beta$  is a generalized parameter inspired from the fractal dimension. The  $\Gamma_\Delta$  function is meant to incorporate the strain induced by the unresolved scales between  $\Delta$  and  $\eta_c$ . Extensions of this model have also been proposed to compute the parameter  $\beta$  dynamically [30,31]. From a machine learning standpoint, these all correspond to predicting the same output  $\overline{\Sigma}$ , but using several input variables:  $(\tilde{c}, \Delta/\delta_L^0, u'_\Delta/S_L^0)$ . More variables could be included to further generalize the approach, e.g. information about the chemical state, since the machine learning framework does not require a strict physical formulation.

### 2.2. Reformulation in the machine learning context

Flame surface density estimation can be seen as the issue of relating the input field  $\tilde{c}$  to a matching output field  $\overline{\Sigma}$ . Supervised learning of this task can be implemented as follows:

- in a first phase, a dataset generated using a DNS is used, where both  $\tilde{c}$  and  $\overline{\Sigma}$  are known exactly. Models are trained on this data in a supervised manner.
- in a second step, the best trained model is frozen. It is executed in an LES context, where  $\tilde{c}$  is known but not  $\overline{\Sigma}$ .

It is important to note that the knowledge of  $\overline{\Sigma}$ , which comes from a DNS, is only needed during the training phase to generate the target values. The learned model, once trained, can be used on new configurations, without needing to perform the DNS. This is shown in Section 5.

Both expressions (6) and (7) are fully local: the flame surface depends only on the local characteristics of turbulence ( $u'_\Delta$ ), on the grid size ( $\Delta$ ) and on the laminar flame characteristics ( $\delta_L^0$  and  $S_L^0$ ).

These functions are of the form:

$$\bar{\Sigma} = f(\bar{c}, \mathbf{u}, \dots) \quad (8)$$

$$f : \mathbb{R}^k \mapsto \mathbb{R}$$

where  $k$  is the number of local variables considered. A generalized DL approach however could use more data by extracting topological information from the flow. In this study, we investigate the capability of spatial convolution to read the vector  $\bar{c}$  of values over an entire subdomain  $\Omega$ , and to produce a prediction for the matching field of  $\bar{\Sigma}$  over  $\Omega$ . The function  $f_{CNN}$  therefore performs:

$$\bar{\Sigma} = f_{CNN}(\bar{c}) \quad (9)$$

$$f_{CNN} : \mathbb{R}^{\Omega} \mapsto \mathbb{R}^{\Omega}$$

The CNN input is 3D matrix, meaning  $\Omega$  must be a 3D regular grid. The neural network architecture chosen in this work (Fig. 4) imposes a single constraint on  $\Omega$ : each dimension must be a multiple of 4. Other than that, there are no constraints on the input to the network, and it can be used on arbitrarily large domains.

The nature of  $f_{CNN}$  differs from classical SGS models which use only local information to infer the subgrid reaction rate: the CNN explores the flow around each point to construct subgrid quantities. Convolutions are promising for this task for several reasons:

- convolutions are an efficient strategy to obtain approximations of any order of derivatives of a scalar field [32];
- flames are not local elements but complex structures that spread over several mesh points. Analyzing these structures using algebraic (pointwise) models [29,33] is challenging. The spatial analysis offered by successive convolutions may enable to better understand the global topology of the flame and therefore permit a better estimation of the unresolved structures;
- recent advances in training convolutional neural networks have lead to a high availability of these methods;
- convolutions enable to train models on large inputs *via* parameter sharing. This implies that the parameter  $n$  in Eq. (9) can be high, even though the dimensionality of the problem increases with the cube of  $n$ . This contrasts with other classical machine learning approaches, including non-convolutional neural networks, which would quickly become impractically large on so many inputs.

### 3. Building the training database

#### 3.1. Direct numerical simulations of premixed flames

In order to obtain  $|\nabla\bar{c}|$  and  $\bar{\Sigma}$  fields needed to train the CNN, two DNS of a methane-air slot burner are used. Their instantaneous snapshots are treated to produce  $c$  and  $\nabla c$ , and filtered (see Section 3.2).

The fully compressible explicit code AVBP is used to solve the filtered multi-species 3D Navier–Stokes equations with simplified thermochemistry on unstructured meshes [34,35]. A Taylor–Galerkin finite element scheme called TTGC [36] of third-order in space and time is used. Inlet and outlet boundary conditions are treated using an NSCBC approach [37] with transverse terms corrections [38]. Other boundaries are treated as periodic.

Chemical kinetics of the reactions between methane and air at 1 bar are modeled using a global 2-step scheme fitted to reproduce the flame propagation properties such as the flame speed, the burned gas temperature and the flame thickness [39]. This simplified chemistry description is sufficient to study the dynamics of premixed turbulent flames. Fresh gases are a stoichiometric mixture with flame speed  $S_L^0 = 40.5$  cm/s and thermal flame thickness

**Table 1**

Parameters for the two DNS simulations performed to produce training data for the CNN.

|      | $u_{rms}/\bar{u}$ | Snapshots | $u'/S_L^0$ | Da |
|------|-------------------|-----------|------------|----|
| DNS1 | 5%                | 50        | 1:23       | 7  |
| DNS2 | 10%               | 50        | 2:47       | 15 |

0.34 mm. The mesh is a homogeneous cartesian grid with constant element size  $dx = 0.1$  mm, ensuring 7–9 points in the pre-heat zone and 4–5 in the reaction zone. Flame speed and thickness were found to be conserved within 5% on a laminar 1D flame. The domain size is 512 cells in the  $x$  direction and 256 cells in the  $y$  and  $z$  ones, for a total of 33.6 million cells. It is periodic in the  $y$  and  $z$  directions, and fed by a profile of fresh and burnt gases in the  $x = 0$  plane (Fig. 1). The inlet is set with a double hyperbolic tangent profile in the  $y$  direction, with a central flow of fresh gases enclosed in slower burnt gases coflows. Inlet temperatures are 300 and 2256 K in the fresh and burnt gases, respectively. Inlet velocities are  $u_{in} = 10$  and  $u_{coflow} = 0.1$  m/s. The characteristic width of the shear layer, as defined by Pope [40], is  $\delta_m = 0.34$  mm, with a corresponding Reynolds number of  $Re_m \approx 200$ .

- The central flow is a fresh stoichiometric mixture of methane and air.
- The coflow is a slow stream of burnt gases, identical in temperature and mixture to the product of the complete combustion of the central flow.
- Turbulence is injected in the fresh gases only. Simulations are performed with either 5% or 10% turbulence injected according to a Passot–Pouquet spectrum [41] with an integral length scale  $l_F = 2$  mm, yielding  $l_F/\delta_L^0 \approx 6$ . The fresh gas injection channel has a height  $h = 8$  mm ( $h/\delta_L^0 \approx 25$ ).

Table 1 describes the two DNS simulations performed in this study and used to train the CNN. The global Damköhler number is estimated as  $Da = u'/S_L^0 \times l_F/\delta_L^0$ .

DNS1 and DNS2 are steady-state simulations, run for 14 ms each. The first 4 ms are transient and discarded, leading to 2 datasets of 10 ms each, with a full field saved every 0.2 ms. This ensures that the fresh gases have traveled approximately 20 mesh points between each snapshot, yielding significant changes in flame shape and therefore diversity in the training data for the CNN.

#### 3.2. Dataset

Two meshes are used in this study:

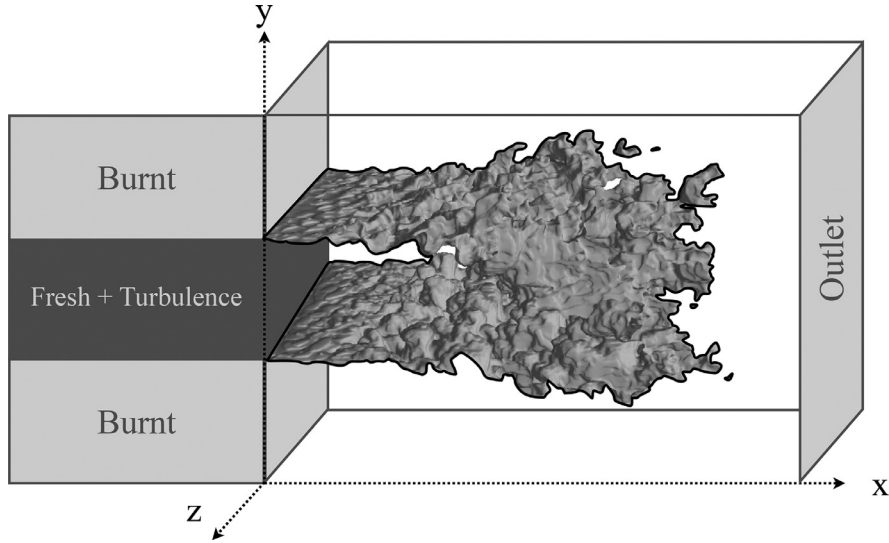
- a DNS mesh used to perform the reactive simulations, which contains  $512 \times 256 \times 256$  cells.
- an “LES” mesh, which represents the same domain but 8 times coarser in every direction, *i.e.*  $64 \times 32 \times 32$  cells.

Fine solutions are produced on the DNS mesh using the Navier–Stokes solver, and then filtered according to Eq. (1) and downsampled on the lower resolution LES mesh. In order to perform this filtering operation, a Gaussian filter is implemented. Its *width* is defined as the multiplying factor on the maximum gradient  $|\nabla c|$ , *i.e.*:

$$\Delta = \frac{\max |\nabla c|}{\max |\nabla \bar{c}|} dx \quad (10)$$

computed on a 1D laminar DNS. The resulting function is therefore written in discrete form as:

$$F_{\Delta}(n) = \begin{cases} e^{-\frac{1}{2}(\frac{n}{\sigma})^2} & \text{if } n \in [1, N] \\ 0 & \text{otherwise} \end{cases} \quad (11)$$



**Fig. 1.** Physical domain used for the DNS. At the inlet, a double hyperbolic tangent profile is used to inject fresh gases in a sheet  $\approx 8$  mm high, surrounded by a slower coflow of burnt gases. Top-bottom (along  $y$ ) and left-right (along  $z$ ) boundaries are periodic. Yellow isosurface is a typical view of  $T = 1600$  K for DNS2.

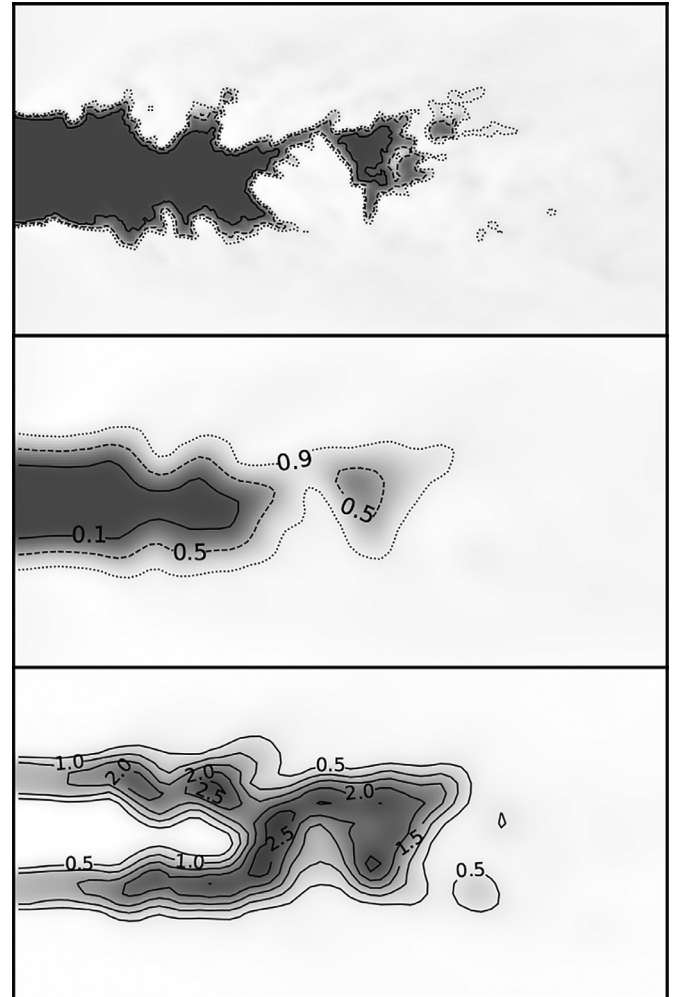
and then normalized by its sum  $\sum_{n \in [0, N]} F_{\Delta}(n)$ . Here,  $\sigma = 26$  and  $N = 31$  are optimized to obtain a filter width  $\Delta = 8dx \approx 2.3 \delta_f^0 \approx l_F/2.5$ .

Data is often normalized when dealing with machine learning tasks, e.g by subtracting the mean and dividing by the standard deviation of the dataset. In the context of the methodology presented here, these values are not known *a priori* on a new combustion setup, and only the DNS can yield the information for the output data. The overarching goal of the approach presented here is to apply the technique to cases where a DNS cannot be performed, hence the need for the network to learn features that are not specifically tailored to a single setup. To achieve this, the input and target fields must be normalized in a fashion that is reproducible *a-priori*.

To reach this goal, the input field  $\bar{c}$  is normalized by construction in Eq. (2). Indeed, for premixed combustion this field goes from 0 in the fresh gases to 1 in the burnt flow. The output flame surface density value  $\bar{\Sigma}$  however spans from 0 far from the flame (both in fresh and burnt gases) to a maximum value that depends on the amount of SGS wrinkling of the flame. The maximum value of  $\bar{\Sigma}$  on a laminar 1D flame is used to normalize this field:  $\bar{\Sigma}_{lam}^{max}$ . The normalized target value writes:

$$\bar{\Sigma}^+ = \frac{\bar{\Sigma}}{\bar{\Sigma}_{lam}^{max}} \quad (12)$$

and does not exceed 1 in areas where the flame is not wrinkled at the subgrid scale. Values exceeding 1 suggest unresolved flame surface. Figure 2 shows a typical instantaneous snapshot of the configuration in the  $(x-y)$  plane:  $\bar{\Sigma}^+$  varies between  $\approx 1$  near the inlet, where turbulence injection has not yet wrinkled the flame, and a maximum of  $\approx 3$  in some local pockets. This shows how the instantaneous field requires specific FSD estimation locally. The DNS field is used to produce input and output fields of lower resolution, which in turn are used to train the neural network. The complete training strategy is shown in Fig. 3. The DNS field of  $c$  is filtered to produce  $\bar{c}$  and  $\bar{\Sigma}^+$ , then sampled on the 8 times coarser LES mesh. These two fields are then sampled on  $X \in \mathbb{R}^{n^3}$ ,  $n = 16$  and fed to the neural network as input/output training.



**Fig. 2.**  $(x-y)$  slice view of the last field from DNS2 (“snapshot 0” in Fig. 5). Fully resolved progress variable  $c$  (top). From this data, the input of the neural network  $\bar{c}$  (middle) and target output to be learned  $\bar{\Sigma}^+$  (bottom) are produced.

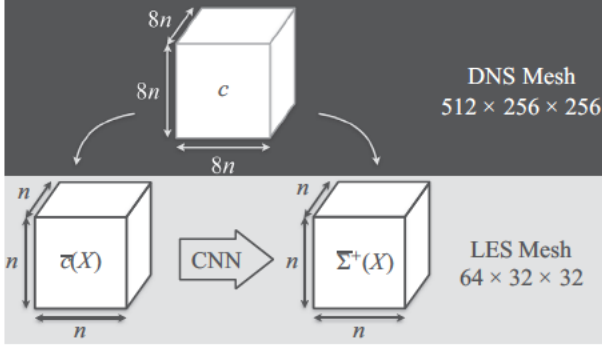


Fig. 3. Training strategy to evaluate SGS wrinkling of a premixed flame: the DNS field of  $c$  is filtered to produce  $\bar{c}$  and  $\Sigma^+$  on an LES mesh that is 8 times coarser than the DNS. The CNN is then trained on the dataset to approximate the function  $\Sigma^+ = f_{CNN}(\bar{c})$ .

## 4. Training the CNN to perform FSD estimation

### 4.1. Neural network approaches for scalar field generation

The objective of the neural network implementation in this study is as follows:

- read on input a 3D field of  $\bar{c}$ , defined on a domain  $\mathbb{R}^\Omega$
- train to produce a 3D field of  $\Sigma^+$

While neural networks were introduced over half a century ago, the recent spike in interest around 2012 was fueled in part by their success on the ImageNet image classification challenge [2]. Image classification is a task where the input is of high dimension (e.g. a 2D, color image), and the output is of lower dimension, typically the number of possible classes (1000 in the case of ImageNet). The network is therefore expected to perform dimensionality reduction, and there is no sense of locality between the input and output.

In this study, a different task is needed: the input field  $\bar{c}$  must be mapped at every mesh point with a matching FSD value, yielding a total  $\Sigma^+$  field of the same dimension as the input. While traditional machine learning approaches might treat this problem

point by point, the neural network can perform an analysis of the full field in a single inference. In order for this to be efficient, the output must not yield a single node value for  $\Sigma^+$ , but a full field.

This task is closer to a machine learning task called image segmentation: an input image is read and analyzed, and an output of same dimension classifies each pixel. This task has also gained traction in the community over the same period, notably with the PASCAL-VOC challenge [42]. In this problem, the classification task must be performed without losing the locality of the information. To this end, fully convolutional neural networks [43] were introduced, showing significant progress in segmentation tasks. Reconstructing the output however took a complex assembly of multiple scales, and later a new architecture which automates this multi-scale analysis called U-Net [44] was introduced. This structure has been expanded upon, yielding deeper networks that are more accurate for complex segmentation tasks [45,46].

### 4.2. Neural network architecture

Review of the literature from Section 4.1 suggest that a U-net architecture [44] offers a basis for the task at hand in this study. U-nets however are optimized for 2D image segmentation. This implies that:

- The input is expected to be a 3 channel (RGB color) 2D matrix representing an image. To adapt to the current case, a single channel will be used on input, but as a full 3D scalar field  $\bar{c}$  over the subdomain  $\mathbb{R}^\Omega$ .
- The output of a U-net is meant for a classification task. Output activation functions are therefore used to represent a categorical distribution, and in the present case must be swapped with rectified linear unit activation to better fit the regression task at hand [1].

The U-net-inspired architecture chosen for this study, adapted for the regression task at hand is shown in Fig. 4. It consists of a fully convolutional neural network with a downsampling and an upsampling path. At each downsampling step, two padded 3D convolutions with  $3 \times 3 \times 3$  kernel are applied, each followed by a batch normalization (BN) and a rectified linear unit (ReLU) activation. In addition, a  $2 \times 2 \times 2$  maxpooling operation is applied and

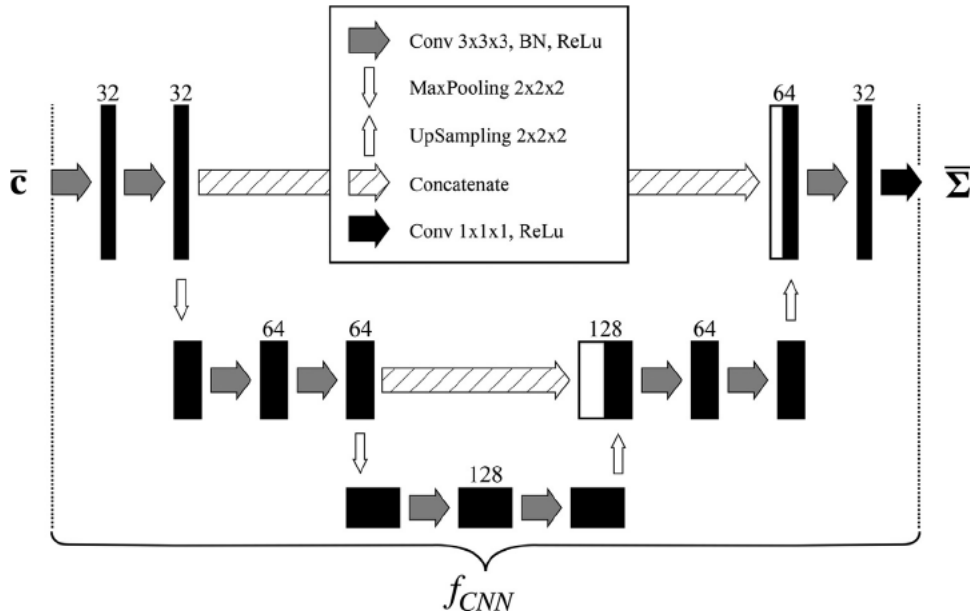


Fig. 4. Structure of the U-net – inspired network used in this study. A total of 13 layers are used. Integers above convolutional layers show the number of filters for each convolution.

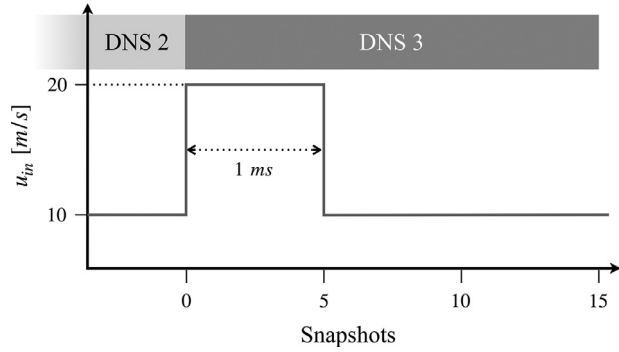


Fig. 5. Inlet velocity versus time (1 snapshot every 0.2 ms) for DNS3, continued from DNS2.

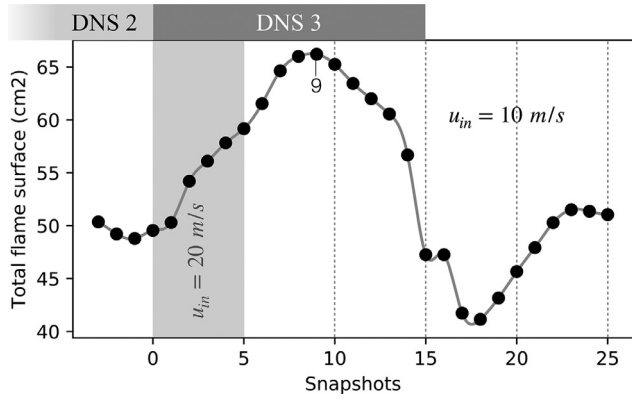


Fig. 6. Total flame surface in the domain versus time during DNS3. Test set spans snapshots 1 through 15. A view of the field from snapshot 9 is shown in Fig. 8.

the number of feature channels is doubled. The upsampling path is a mirrored version of the downsampling path, with a similar structure: it includes 3D transposed convolutions instead of 3D convolutions, and a  $2 \times 2 \times 2$  upsampling operation to recover the initial dimensions. Additionally, according to the U-net structure, skip-connections link layers with equal resolution of each path. In order to perform a regression task the final layer, a 3D transposed convolution with  $1 \times 1 \times 1$  kernel was used, with a ReLu activation to prevent the network from predicting negative outputs. In total, the network consists of 1,414,145 trainable parameters, corresponding

Table 2

Data split for the network training and testing in this study. All columns are expressed in terms of DNS snapshot numbers (1 every 0.2 ms), in sequence.

|      | Training    | Validation  | Testing     |
|------|-------------|-------------|-------------|
| DNS1 | 1–40        | 41–50       | $\emptyset$ |
| DNS2 | 1–40        | 41–50       | $\emptyset$ |
| DNS2 | $\emptyset$ | $\emptyset$ | 1–15        |

to all the weights that need to be adjusted in the network. In the following, the network described here is simply referred to as the CNN.

### 4.3. Training the CNN

The data from the two DNS described in Section 3.2 (Table 1) is used to train the CNN. In machine learning, the data is classically split in three categories:

- the *training* set, used to optimize the weights of the network;
- the *validation* set, used to evaluate the error during training on a set that has not been observed. This enables to detect the point where the network starts overfitting to the training set, and additional training starts to increase the error on the validation set;
- the *testing* set, kept completely unseen during training, and only used *a posteriori* once the training is converged to assess the performance of the full approach.

Training and validation datasets are often taken from the same distribution, and are simply different samples. Ideally, the testing dataset should be taken from a slightly different distribution, in order to show that the underlying features of the data have been learned, and that they can be generalized to new cases. In this study, two DNS with similar setups (DNS1 and DNS2) that lead to similar flames with some variability introduced by different turbulent intensities are used to produce the training and validation sets, by splitting their data (Table 2). In order to obtain a testing set from a different distribution, a dedicated simulation DNS3 is performed, as described in Section 5.

Additionally, data augmentation during training was found to increase the quality of the results. Each training sample is a random  $16 \times 16 \times 16$  crop from the 3D fields, and random  $90^\circ$  rotations and mirror operations are applied since the model should

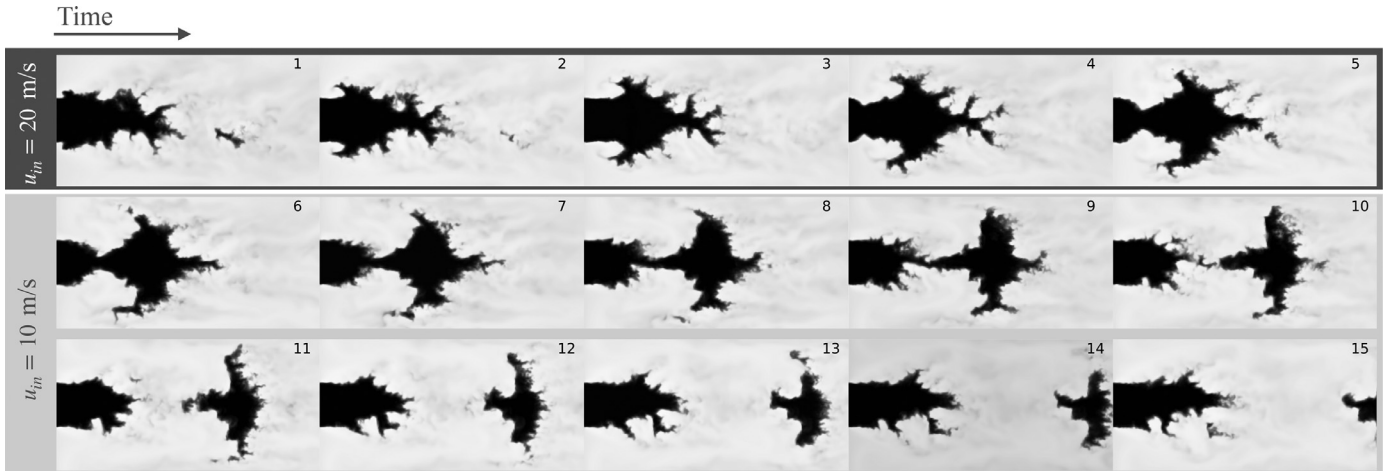


Fig. 7. View of  $c$  in the  $(x-y)$  plane at  $z=0$  for all snapshots (1–15) of DNS3. Black ( $c=0$ ) to white ( $c=1$ ) shows transition from unburnt to burnt gases, respectively. For this DNS, inlet velocity of the fresh gases is doubled for 1 ms (5 snapshots), then set back to its original value for 2 ms (10 snapshots), when the detached pocket of burnt gases reaches the exit.



have no preferential orientation and the network must learn an isotropic function. A training step is performed on a mini-batch of 40 such cubes in order to average the gradient used for optimization and smooth the learning process. The ADAM [47] optimizer is used on a mean-squared-error loss function over all output pixels of the prediction compared to the target. A total of 100 of these mini-batches are observed before performing a test on the validation set to evaluate current train and validation error rates. Each of these 100 mini-batch runs is called an *epoch*. The learning rate, used to weight the update value given by the gradient descent procedure, is initially set to 0.01, and decreased by 20% every 10 epochs. The network converges in  $\approx 150$  epochs, for a total training time of 20 min on an Nvidia Tesla V100 GPU. On this dedicated processor, the dataset is indeed much smaller than typical DL challenge datasets, yielding comparatively short training times.

## 5. Using the CNN to evaluate subgrid scale wrinkling

### 5.1. DNS3: a simulation tailored for testing

Once the training data has been generated (Section 3) and the CNN has been fully trained on it (Section 4), the network is frozen, and can be used to produce predictions of  $\bar{\Sigma}^+$  based on new fields of  $\bar{c}$  unseen during training. To verify the capacity of the CNN to generalize its learning, a new, more difficult case (DNS3) was used. DNS3 is a short-term transient started from the last field of DNS2, where inlet velocity is doubled, going from 10 to 20 m/s for 1 ms (5 snapshots), and then set back to its original value for 2 more ms (Fig. 7). The RMS value of injected turbulence remains constant at  $u' = 1$  m/s. This sudden change leads to a very different, unsteady flow (Fig. 7) where a “mushroom”-type structure is generated [48] and where turbulence varies very strongly and rapidly. It is a typical situation encountered in chambers submitted to combustion instabilities, and is now used to evaluate the CNN.

This *a priori* estimation of  $\bar{\Sigma}^+$  on new fields of  $\bar{c}$  with a trained and frozen network is referred to as *inference* in machine learning, and it is again performed here on the GPU. As explained in Section 2.2, due to the fully convolutional nature of the chosen network,  $\Omega$  need not be of the size  $n^3$ : the network can be directly executed on a 3D flow field of any size, regardless of the size that it was trained on. Inference is therefore performed on each full-field snapshot in a single pass. This has the strong advantage that there is no overlapping region between inference areas, in which the predictions can be of poorer quality [14]. Inference time is 12 ms for each  $64 \times 32 \times 32$  LES field observed.

Figure 6 displays the total flame surface in the domain versus time during DNS3. Fig. 7 shows all the temporal snapshots of  $c$  during DNS3, used for testing the CNN. As the inlet speed is doubled, more mass flow enters the domain and the total flame surface increases. After the mass flow is set back to its initial value at snapshot 5, the flame surface continues to increase until snapshot  $\approx 9$ , which matches the highly wrinkled aspect of the flame as seen in Fig. 8. The mass flow then decreases below its original level, when the unburnt gas pocket exits the domain, starting at snapshot 15. The flame then grows back to its stable length and total area near snapshot 23. Snapshots after number 15 were not included in the testing dataset DNS3: indeed, no significant difference was observed, and this quasi-stable state is less challenging for the generalization of the trained network.

The objective of the network is to predict a value of  $\bar{\Sigma}^+$  at every node and for every instantaneous snapshot that is as close as possible to the true value computed in the DNS  $\bar{\Sigma}_{target}^+$ . Figure 9(a) shows the overall point by point agreement on the full test set,

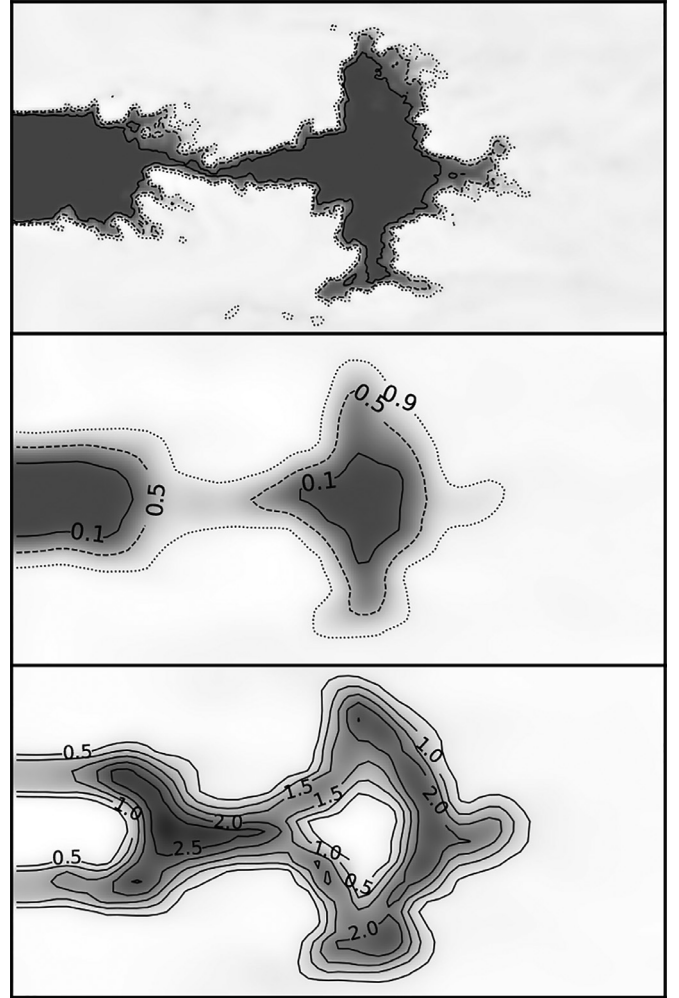


Fig. 8.  $(x - y)$  slice view of snapshot 9 from DNS3. Fully resolved progress variable  $c$  (top). From this data, the input of the neural network  $\bar{c}$  (middle) and target output to be learned  $\bar{\Sigma}^+$  (bottom) are produced.

and demonstrates that the network recovers well the overall trend in the data. In order to better appreciate the error, Fig. 9(b) plots the Root Mean Squared Error (RMSE) of the prediction for bins of points sharing a predicted value in 0.1-wide windows. This shows that the maximum RMSE occurs for the higher values of  $\bar{\Sigma}^+$ , and Fig. 9(c) indicates that some snapshots experience rare extreme RMSE values that can reach 0.4. These events are however limited, and the majority of errors are in the  $[0 - 0.2]$  range. This is a normalized value directly comparable to  $\bar{\Sigma}^+$ , which is valued at 1 in unwrinkled flame fronts and  $\approx 2.5$  in highly wrinkled areas (Fig. 8). From this we conclude that the transient data of DNS3 performs very well on the testing set in a statistical sense.

### 5.2. Comparison with algebraic models

One issue with learning techniques, including CNNs, is that they are the result of an optimization process where a mean squared error of a loss function is minimized. The resulting loss function value is hard to interpret in a physical sense. In order to give a baseline to compare the accuracy of the technique with, the model of Charlette et al. [29] was implemented with a parameter value  $\beta = 0.5$ . This efficiency function assumes flame-turbulence equilibrium to evaluate the amount of sub-grid scale wrinkling, ultimately yielding  $\Xi$ . Eq. (5) gives the relationship with  $\bar{\Sigma}$ , and therefore in

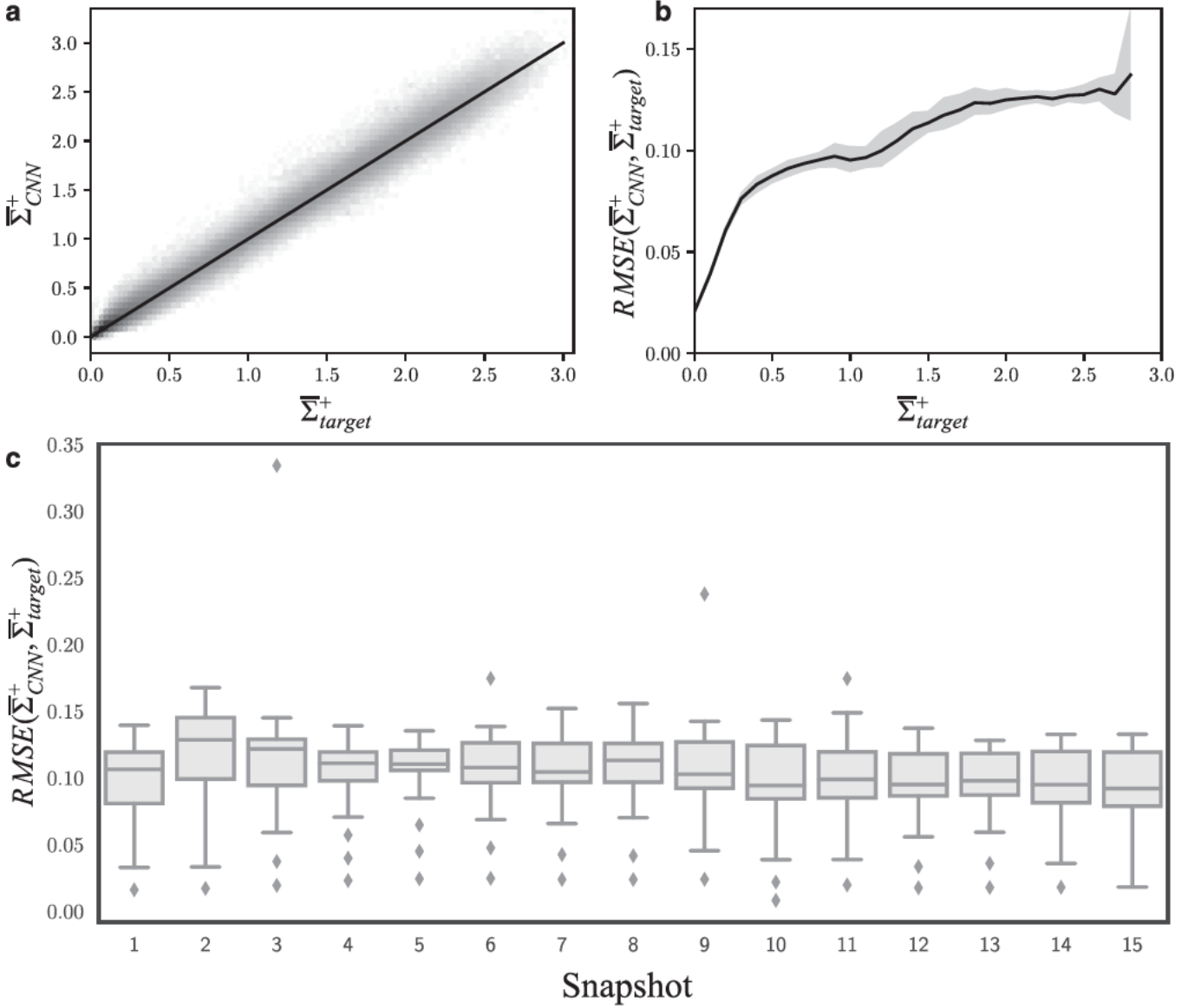


Fig. 9. Data from the predictions of the CNN applied to the test dataset, *i.e.* the 15 snapshots of DNS3. (a) Scatterplot of  $\bar{\Sigma}^+$  predicted by the CNN vs target value extracted from the DNS. Grey line indicates  $y = x$ , the points with 0 error. (b) Root Mean Squared Error (RMSE) of the  $\bar{\Sigma}^+$  predictions versus target DNS value. Dark line is the mean value of RMSE over all 15 snapshots, light-grey shows 95% confidence interval around this value. (c) Boxplot view of RMSE distribution for each snapshot. Central bar: median value; grey box: first to last quartile; vertical line: 95% occurrences; contour: kernel density plot estimate for full data.

order to compare this method to the output of the CNN the following field is built:

$$\bar{\Sigma}_{Charlette}^+ = \frac{\mathbb{E}|\nabla\bar{c}|}{\bar{\Sigma}_{lam}^{max}} \quad (13)$$

with the same normalization factor as Eq. (12),  $\bar{\Sigma}_{lam}^{max}$ . Note that, as explained with Eq. (7), this form has been modified [30] so that when “saturated” (*i.e.* for high SGS velocities) it recovers the fractal model [26] of Eq. (6), with a fractal dimension of  $D_f = 2.5$ . The prediction made by this fractal model can also be compared to the present results in the form:

$$\bar{\Sigma}_{Fractal}^+ = \frac{\mathbb{E}_{Fractal}|\nabla\bar{c}|}{\bar{\Sigma}_{lam}^{max}} \quad (14)$$

Figure 10 displays the detailed views of the total flame area integrated in 8 mm slices (1 LES mesh cell) along axial position for

each snapshot of DNS3. Figure 11 displays the mean absolute prediction error of flame surface versus axial position  $x$  for all DNS3 snapshots, *i.e.* a condensed view of Fig. 10. It is clear from this graph that the CNN performs with very high accuracy compared to the other models. At the inlet, where turbulence is present but has not yet wrinkled the flame front, the fractal model overestimates significantly the amount of flame surface. This is due to the fact that the flame front is not yet wrinkled by turbulent structures. Turbulence and flame motions have not yet reached an equilibrium, usually assumed in the derivation of algebraic flame surface models. The CNN however has learned that the large-scale topology of the flow is not wrinkled, and therefore that the flame is in fact not wrinkled, leading to an excellent prediction in this region. Downstream, the Charlette model significantly underpredicts the flame surface (*cf.* Fig. 10). This suggests that the Charlette model is far from its saturation value of Eq. (6), hence the SGS turbulence is underestimated in this configuration.

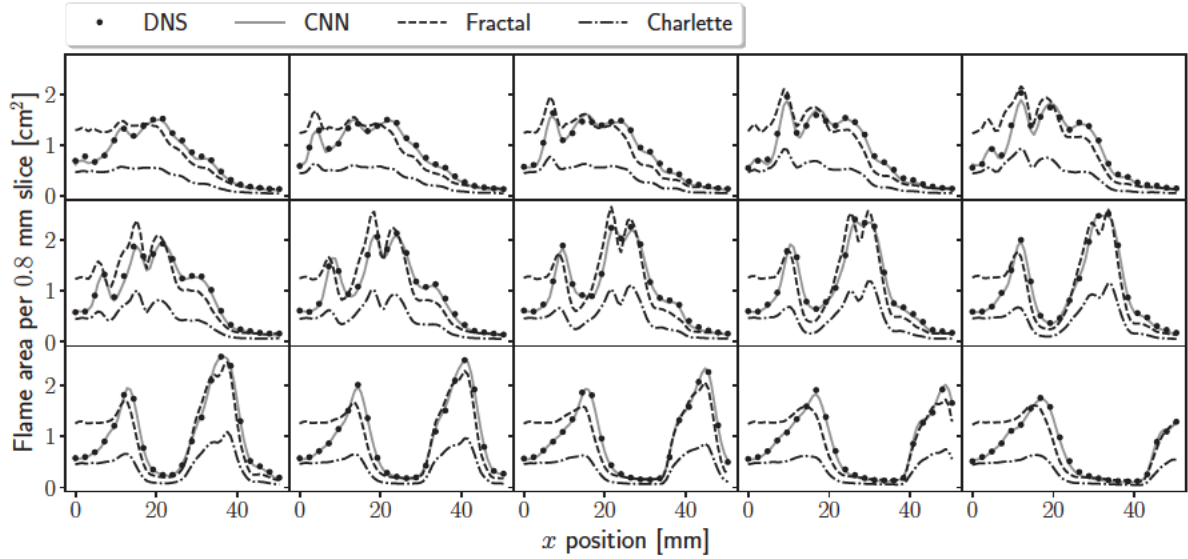


Fig. 10. Total flame surface at each  $x$  location for all snapshots of DNS3.

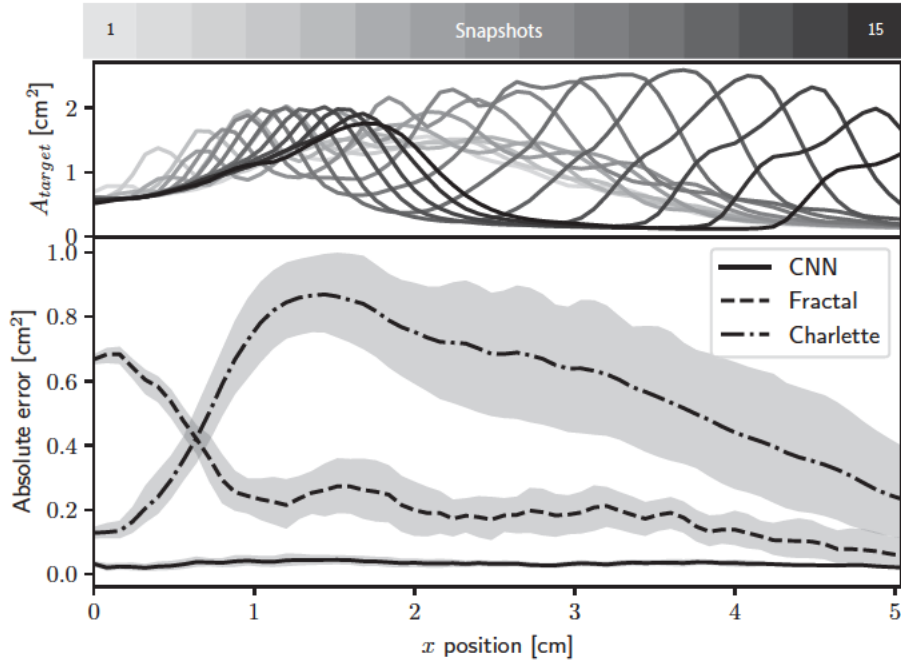


Fig. 11. Top: Total flame surface at location  $x$  (integrated in the  $y-z$  plane) from DNS3 at each instant with snapshots ordered by increasing darkness; Bottom: Absolute error of CNN and Charlette predictions for the same  $x$  locations. Grey area shows 95% confidence interval.

## 6. Conclusions

In this study, a convolutional neural network (CNN) inspired from a U-net architecture was used to predict SGS flame surface density for a premixed turbulent flame on an underresolved mesh typical of large eddy simulation. Results show excellent agreement for this case between the prediction by the neural network and the baseline value from the DNS. Some algebraic models from the literature are shown for comparison and significantly under-perform compared to the CNN. This suggests that including topological information to estimate unresolved flame wrinkling helps to achieve better accuracy. The advantage of the present technique is that this extraction was not hand-designed: with little effort, the training process leads to automatic and multi-scale extraction to assist the prediction.

This study demonstrates the capacity of a deep learning approach to learn to predict SGS contribution to flame wrinkling, as well as to generalize this task to a different test case than the training set. This work can therefore be seen as a proof of concept, demonstrating that an SGS combustion model can be reformulated as a machine learning problem, and accurately learned using a CNN. To the best of the author's knowledge, this is the first application of CNNs to turbulent combustion modeling.

This new technique poses some specific challenges, unlike traditional models, that should be investigated in future work to expand the capacity of the learned network to generalize to new configurations and flame regimes. One could be a thorough comparison, both *a priori* (as done here) and *a posteriori*, with so-called dynamic models [30], considered as state-of-the-art in the community. Another could be to explore the robustness of the model – or

adapt the approach accordingly – to different conditions typically encountered in realistic configurations. This could include e.g. flame thickening factor, pressure, inlet temperature, or equivalence ratio variations. Indeed, in its current form, the method needs to be either tested or retrained as soon as these parameters vary, but success of CNN generalization in the literature suggest that single architectures could be trained over significant ranges of these quantities.

## Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## References

- [1] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, *Deep learning*, 1, MIT Press, Cambridge, 2016.
- [2] A. Krizhevsky, I. Sutskever, G.E. Hinton, *Imagenet classification with deep convolutional neural networks*, *Advances in Neural Information Processing Systems* (2012), pp. 1097–1105.
- [3] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A.A. Kalyanpur, A. Lally, J.W. Murdock, E. Nyberg, J. Prager, et al., *Building watson: an overview of the deepqa project*, *AI Mag.* 31 (3) (2010) 59–79.
- [4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., *Mastering the game of go without human knowledge*, *Nature* 550 (7676) (2017) 354.
- [5] W.S. McCulloch, W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, *Bull. Math. Biophys.* 5 (4) (1943) 115–133.
- [6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, 2015, *TensorFlow: large-scale machine learning on heterogeneous systems*, software available from tensorflow.org.
- [7] H.W. Lin, M. Tegmark, D. Rolnick, *Why does deep and cheap learning work so well?* *J. Stat. Phys.* 168 (6) (2017) 1223–1247.
- [8] K. Duraisamy, Z.J. Zhang, A.P. Singh, *New approaches in turbulence and transition modeling using data-driven techniques*, 53rd AIAA Aerospace Sciences Meeting (2015), p. 1284.
- [9] J. Ling, A. Kurzawski, J. Templeton, *Reynolds averaged turbulence modelling using deep neural networks with embedded invariance*, *J. Fluid Mech.* 807 (2016) 155–166.
- [10] A. Volland, G. Balarac, C. Corre, *Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures*, *J. Turbul.* 18 (9) (2017) 854–878.
- [11] R. Maulik, O. San, *A neural network approach for the blind deconvolution of turbulent flows*, *J. Fluid Mech.* 831 (2017) 151–181.
- [12] M. Schoepflein, J. Weatheritt, R. Sandberg, M. Talei, M. Klein, *Application of an evolutionary algorithm to LES modelling of turbulent transport in premixed flames*, *J. Comput. Phys.* 374 (2018) 1166–1179.
- [13] K. Duraisamy, G. Iaccarino, H. Xiao, *Turbulence modeling in the age of data*, *Ann. Rev. Fluid Mech.* 51 (2019) 357–377.
- [14] A.D. Beck, D.G. Flad, C.-D. Munz, *Deep neural networks for data-based turbulence models*, arXiv preprint arXiv:1806.04482.
- [15] T. Poinso, D. Veynante, *Theoretical and numerical combustion*, 2011 Third ed. RT Edwards, Inc., ([www.cerfacs.fr/elearning](http://www.cerfacs.fr/elearning)).
- [16] F.E. Marble, J.E. Broadwell, *The coherent flame model for turbulent chemical reactions*, Technical report trw-9-pu, 1977.
- [17] S. Candel, D. Veynante, F. Lacas, E. Maistret, N. Darabiha, T. Poinso, *Coherent flame model: applications and recent extensions*, in: B. Larrouturou (Ed.), *Advances in Combustion Modeling*, Series on advances in mathematics for applied sciences, World Scientific, Singapore (1990), pp. 19–64.
- [18] S. Pope, *The evolution of surfaces in turbulence*, *Int. J. Eng. Sci.* 26 (5) (1988) 445–469.
- [19] K.N.C. Bray, J.B. Moss, *A unified statistical model of the premixed turbulent flame*, *Acta Astron.* 4 (1977) 291–319.
- [20] N. Peters, *Laminar flamelet concepts in turbulent combustion*, *Symp. (Int.) Combust.* 21 (1986) 1231–1250.
- [21] J.M. Duclos, D. Veynante, T. Poinso, *A comparison of flamelet models for premixed turbulent combustion*, *Combust. Flame* 95 (1993) 101–117.
- [22] G. Bruneaux, T. Poinso, J.H. Ferziger, *Premixed flame-wall interaction in a turbulent channel flow: budget for the flame surface density evolution equation and modelling*, *J. Fluid Mech.* 349 (1997) 191–219.
- [23] M. Boger, D. Veynante, H. Boughanem, A. Trouvé, *Direct numerical simulation analysis of flame surface density concept for large eddy simulation of turbulent premixed combustion*, *Symp. (Int.) Combust.* 27 (1998) 917–927.
- [24] R. Knikker, D. Veynante, C. Meneveau, *A dynamic flame surface density model for large eddy simulation of turbulent premixed combustion*, *Phys. Fluids* 16 (2004) L91–L94.
- [25] A.R. Kerstein, W. Ashurst, F.A. Williams, *Field equation for interface propagation in an unsteady homogeneous flow field*, *Phys. Rev. A* 37 (7) (1988) 2728–2731.
- [26] F. Gouldin, K. Bray, J.Y. Chen, *Chemical closure model for fractal flamelets*, *Combust. Flame* 77 (1989) 241.
- [27] T. Poinso, D. Veynante, S. Candel, *Quenching processes and premixed turbulent combustion diagrams*, *J. Fluid Mech.* 228 (1991) 561–605.
- [28] O.L. Gulder, G.J. Smallwood, *Inner cutoff scale of flame surface wrinkling in turbulent premixed flames*, *Combust. Flame* 103 (1995) 107–114.
- [29] F. Charlette, D. Veynante, C. Meneveau, *A power-law wrinkling model for LES of premixed turbulent combustion. Part I – non-dynamic formulation and initial tests*, *Combust. Flame* 131 (2002) 159–180.
- [30] G. Wang, M. Boileau, D. Veynante, *Implementation of a dynamic thickened flame model for large eddy simulations of turbulent premixed combustion*, *Combust. Flame* 158 (11) (2011) 2199–2213.
- [31] F. Charlette, C. Meneveau, D. Veynante, *A power-law flame wrinkling model for LES of premixed turbulent combustion. Part II: dynamic formulation*, *Combust. Flame* 131 (1–2) (2002) 181–197.
- [32] W.K. Pratt, *Digital image processing: PIKS scientific inside*, 4, Wiley-interscience Hoboken, New Jersey, 2007.
- [33] O. Colin, F. Ducros, D. Veynante, T. Poinso, *A thickened flame model for large eddy simulations of turbulent premixed combustion*, *Phys. Fluids* 12 (7) (2000) 1843–1863.
- [34] T. Schönfeld, M. Rudgyard, *Steady and unsteady flows simulations using the hybrid flow solver AVBP*, *AIAA J.* 37 (11) (1999) 1378–1385.
- [35] L. Selle, G. Lartigue, T. Poinso, R. Koch, K.U. Schildmacher, W. Krebs, B. Prade, P. Kaufmann, D. Veynante, *Compressible large-eddy simulation of turbulent combustion in complex geometry on unstructured meshes*, *Combust. Flame* 137 (4) (2004) 489–505.
- [36] O. Colin, M. Rudgyard, *Development of high-order Taylor–Galerkin schemes for unsteady calculations*, *J. Comput. Phys.* 162 (2) (2000) 338–371.
- [37] T. Poinso, S. Lele, *Boundary conditions for direct simulations of compressible viscous flows*, *J. Comput. Phys.* 101 (1) (1992) 104–129.
- [38] V. Granet, O. Vermorel, T. Leonard, L. Gicquel, T. Poinso, *Comparison of non-reflecting outlet boundary conditions for compressible solvers on unstructured grids*, *AIAA J.* 48 (10) (2010) 2348–2364.
- [39] B. Franzelli, E. Ribet, L.Y. Gicquel, T. Poinso, *Large eddy simulation of combustion instabilities in a lean partially premixed swirled flame*, *Combust. Flame* 159 (2) (2012) 621–637.
- [40] S.B. Pope, *Turbulent flows*, Cambridge University Press, 2000.
- [41] T. Passot, A. Pouquet, *Numerical simulation of compressible homogeneous flows in the turbulent regime*, *J. Fluid Mech.* 181 (1987) 441–466.
- [42] M. Everingham, L.V. Gool, C.K.I. Williams, J. Winn, A. Zisserman, *The Pascal Visual Object Classes (VOC) challenge*, *Int. J. Comput. Vis.* 88 (2) (2010) 303–338.
- [43] J. Long, E. Shelhamer, T. Darrell, *Fully convolutional networks for semantic segmentation*, *IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 3431–3440.
- [44] O. Ronneberger, P. Fischer, T. Brox, *U-net: convolutional networks for biomedical image segmentation*, *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer (2015), pp. 234–241.
- [45] L.C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, *Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs*, *IEEE Trans. Pattern Anal. Mach. Intell.* 40 (4) (2018) 834–848.
- [46] Z. Liu, X. Li, P. Luo, C.-C. Loy, X. Tang, *Semantic image segmentation via deep parsing network*, in: *Proceedings of the IEEE International Conference on Computer Vision* (2015), pp. 1377–1385.
- [47] D.P. Kingma, J. Ba, *A method for stochastic optimization*, in: *Proceedings of the 3rd International Conference on Learning Representations* (2015).
- [48] T. Poinso, A. Trouvé, D. Veynante, S. Candel, E. Esposito, *Vortex driven acoustically coupled combustion instabilities*, *J. Fluid Mech.* 177 (1987) 265–292.