Deep learning for subgrid-scale modeling in large eddy simulations of turbulent premixed combustion

Victor Xing

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Dynamique des fluides

Présentée et soutenue par :

M. VICTOR XING

le vendredi 18 novembre 2022

Titre :

Apprentissage automatique profond pour la modélisation de sous-maille en simulations aux grandes échelles de combustion prémélangée turbulente

---

Ecole doctorale :

Mécanique, Energétique, Génie civil, Procédés (MEGeP)

Unité de recherche :

Centre Européen de Recherche et Formation Avancées en Calcul Scientifique (CERFACS)

Directeur(s) de Thèse :

M. THIERRY POINSOT

M. CORENTIN LAPEYRE

Rapporteurs :

M. ANTONIO ATTILI, UNIVERSITY OF EDINBURGH

M. DENIS VEYNANTE, CENTRALESUPELEC GIF SUR YVETTE

Membre(s) du jury :

MME PASCALE DOMINGO, CORIA, Président

M. CORENTIN LAPEYRE, CERFACS, Membre

M. MICHAEL PFITZNER, UNIVERSITAT DER BUNDESWEHR MUNICH, Membre

M. THIERRY POINSOT, TOULOUSE INP, Membre

Abstract

In a century defined by climate change and data abundance, combustion is moving towards new opportunities created by the digital revolution. Large eddy simulations (LES) of full-scale practical combustion systems are becoming tractable, but their predictive power hinges on the accuracy of the subgrid-scale (SGS) models that account for unresolved turbulent combustion physics. Deep learning (DL) has recently been used to train data-driven SGS models that achieve excellent accuracy in *a priori* tests. Yet, there are still hardly any applications of DL SGS models in LES of practical combustion systems. This work investigates three elements that must be addressed to enable the adoption of deep learning in practical LES of turbulent premixed combustion: evaluating DL models on high Reynolds test cases, ensuring their ability to generalize beyond their training configuration, and implementing a computationally efficient integration of DL models to high-performance LES solvers. Three DL models that gradually include each of these elements are developed. They are based on U-Net convolutional neural networks (CNNs) trained on downsampled filtered snapshots of direct numerical simulations. First, a model for the total flame surface density is trained on the R2 high Reynolds turbulent jet flame. Excellent *a priori* generalization to higher Reynolds numbers and to LES snapshots is observed, and insights on the inner workings of the model are provided. Second, a CNN model for the SGS variance of the progress variable is trained on a statistically planar turbulent flame. With a Pfitzner source term formulation and a beta probability density function closure, it is able to accurately predict *a priori* the SGS variance and the filtered reaction rate on the R2 jet flame, thus demonstrating its ability to generalize to new configurations. Third, the AVBP-DL coupling strategy is developed to enable DL models to be queried by the AVBP solver with negligible computational overhead. Finally, the Masri vented obstructed explosion test case is used to test *a posteriori* a CNN model for the SGS wrinkling factor trained on the statistically planar turbulent flame. The model predicts the right peak overpressure, but this results from a compensation between excessive wrinkling in the initial laminar phase and insufficient wrinkling in the critical turbulent propagation stage. Several attempts to correct this behavior are then explored.

Keywords: deep learning, large eddy simulation, subgrid-scale model, numerical combustion.

Résumé

Dans un siècle défini par le changement climatique et l'abondance de données, la combustion se dirige vers de nouvelles opportunités créées par la révolution numérique. Les simulations aux grandes échelles (Large Eddy Simulations, LES) de systèmes de combustion à échelle réelle deviennent réalisables, mais leur capacité prédictive se base sur la précision de modèles de sous-maille (Subgrid-Scale, SGS) qui tiennent compte de l'activité de combustion turbulent non résolue. L'apprentissage automatique profond (Deep Learning, DL) a récemment été utilisé pour entraîner des modèles SGS basés sur les données qui atteignent une excellent précision lors de tests *a priori*. Toutefois, il n'y a toujours presque pas d'applications de modèles DL SGS à des LES de systèmes de combustion industriels. Ces travaux s'intéressent à trois éléments qui doivent être étudiés pour permettre l'adoption du DL dans des LES de combustion turbulente prémélangée : l'évaluation de modèles DL sur des cas tests à haut Reynolds, l'assurance de leur capacité à généraliser au-delà de leur configuration d'entraînement, et l'implémentation d'une intégration efficace de modèles DL à des solveurs LES haute performance. Trois modèles DL incluant graduellement chacun de ces éléments sont développés. Ils ont basés sur des réseaux de neurones convolutionnels (Convolutional Neural Networks, CNNs) U-Nets entraînés sur des instantanés filtrés et déraffinés de simulations numériques directes. Premièrement, un modèle pour la densité totale de surface de flamme est entraîné sur la flamme de jet turbulente à haut Reynolds R2. Une excellente généralisation *a priori* à de plus hauts nombres de Reynolds et à des instantanés LES est observée, et des aperçus sur le fonctionnement interne du modèle sont proposés. Dans un second temps, un modèle CNN pour la variance SGS de la variable de progrès est entraîné sur une flamme plane turbulente statistiquement stationnaire. Avec une formulation Pfitzner du terme source et une fermeture beta densité de probabilité, il est capable de prédire *a priori* avec précision la variance SGS et le taux de réaction filtré sur la flamme de jet R2, démontrant ainsi sa capacité à généraliser à de nouvelles configurations. Troisièmement, la stratégie de couplage AVBP-DL est développée pour permettre à des modèles DL d'être interrogés par le solveur AVBP avec un surcoût de calcul négligeable. Enfin, le cas test d'explosion aérée et obstruée Masri est utilisé pour tester *a posteriori* un modèle CNN pour le facteur de plissement SGS entraîné sur la flamme plane turbulente statistiquement stationnaire. Le modèle prédit la bonne supression maximale, mais ceci résulte d'une compensation entre un plissement excessif lors de la phase initiale laminaire et d'un plissement insuffisant durant l'étape critique de propagation turbulente. Plusieurs tentatives de correction de ce comportement sont ensuite explorées.

Mots-clés: apprentissage automatique profond, simulation aux grandes échelles, modèle de sous-maille, combustion numérique.

Acknowledgements

This scientific journey was made possible thanks to my two PhD supervisors, Corentin Lapeyre and Thierry Poinsot. Thank you for your continued support during these 3+ demanding years, and for allowing me to freely explore the fascinating intersection of numerical combustion and deep learning. Your complementary points of view were invaluable to overcome the twists and turns that shape the course of a PhD. Thierry, your feedback is infallibly accurate, and I am grateful for the trust you have put in the arcane arts of deep learning. Corentin, thank you for introducing me to this discipline, and for upholding its merits amid the combustion community. Your keen intuition has often been instrumental in the progress of my PhD, and I always enjoyed our long conversations on deep learning and more.

I am deeply thankful to Denis Veynante, Antonio Attili, Michael Pfitzner, and Pascale Domingo for their participation in the jury of the defense and the high-quality discussion that ensued. Special thanks to Denis and Antonio for their thorough review of this manuscript. I have also had the pleasure to discuss my work with you several times over the course of the PhD, and these discussions were always a great source of inspiration.

Many thanks to the other members of the combustion community I have exchanged with during my PhD. In particular, thank you to Michael Pfitzner and Junsu Shin for their interest in my work and the collaboration we undertook.

To all my coworkers at Cerfacs, thank you for making this lab a special place. Victor and Jessica, it was a great pleasure to work with you on the DNS comparison, and I hope our efforts will be rewarded. Anass, honestly I do not think I could have made AVBP-DL work without you. Dorian, our epic discussions always pushed the edge of my understanding of deep learning. Luciano, GEM-CNN was an amazing learning experience I was glad to share with you. Soumyo and Elsa, I found much needed solace in your support during our shared lockdown experience with Masri. Gab, your HPC wizardry is only matched by your kindness. Antoine, thank you for the brief stay in the COOP group. Nicolas, Camille, Antony, Bastien, I will keep fond memories of the deep learning office next to the patio. Quentin, thank you for being the venting term to my overpressure, and special thanks to the other members of the LEFEX group: Thomas, Olivier, Jean-Jacques, Benjamin, Francis, Loïc. Thank you to the other PhDs who are too many to all list: Aurélien, Lionel, Ekhi, Jon, Antoine, Thomas, Minh, ... I am also grateful to the CSG team for their flawless reliability and kindness, as well as Michèle, Brigitte, Nathalie for their administrative support.

I was fortunate to meet many wonderful people who have made my years in Toulouse unforgettable: Pierre, who was brave enough to make the trip from Bordeaux to attend the defense, and my other former housemates Alexis, Paul, Arnaud, Frederik; Nico Urien, for the countless coffees and chess games; Clovis and Alyssa, for the board game afternoons; my cycling companions Arthur, Markus, Antony, Etienne, Victor, Marion, Paul, and others at the Repaire; PAM and Hugo, who have achieved the incredible feat of being world-class researchers, climbers, and ping-pong players. Thank you to all my dear friends elsewhere who have supported me: Bo, Louis, Jean, Mathilde, Irina, Son, Marianne, ...

Enfin, je suis infiniment reconnaissant à ma famille pour son soutien indéfectible. Julia, Papa, Maman, vous avez été à mes côtés dans mes succès comme dans les moments difficiles, et je ne vous remercierai jamais assez pour tout ce que vous avez fait pour moi.

Contents

Nomenclature

Abbreviations

BN	Batch Normalization
CDSM	Charlette Dynamic Saturated model with Mouriaux corrections
CFD	Computational Fluid Dynamics
CFL	Courant–Friedrichs–Lewy
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CWIPI	Coupling With Interpolation Parallel Interface
DL	Deep Learning
DNS	Direct Numerical Simulation
ERF	Effective Receptive Field
F-TACLES	Filtered Tabulated Chemistry for LES
FSD	Flame Surface Density
GAN	Generative Adversarial Network
GNN	Graph Neural Network
GPU	Graphics Processing Unit
HIT	Homogeneous Isotropic Turbulence
JPDF	Joint Probability Density Function
KDE	Kernel Density Estimate
KPP	Kolmogorov-Petrovsky–Piskunov
LES	Large Eddy Simulation
ML	Machine Learning
MLP	Multilayer Perceptron
MPMD	Multiple Program Multiple Data
MS	Medium Scale
NSCBC	Navier-Stokes Characteristic Boundary Conditions
ODE	Ordinary Differential Equation
PB-CNN	Pfitzner beta PDF CNN
PDE	Partial Differential Equation
PDF	Probability Density Function
ReLU	Rectified Linear Unit
SGS	Subgrid Scale
SS	Small Scale
TFLES	Thickened Flame LES
TKE	Turbulent Kinetic Energy
TRZ	Thin Reactions Zone
TTGC	Two-step Taylor Galerkin scheme
VAE	Variatioal Autoencoder
VKP	von Karman–Pao
VRLES	Very Refined LES

Deep learning variables

$\boldsymbol{b}$	Bias
c	Number of channels
$\boldsymbol{G}$	Convolution filter
g	Activation function
H	Feature map height
h	Convolution kernel height
L	Loss function
m	Batch size
n	Network depth
p	Probability density function
$\boldsymbol{W}$	Weight matrix
W	Feature map width
w	Convolution kernel width
$\boldsymbol{x}$	Input
$\hat{\boldsymbol{y}}$	Prediction
$\boldsymbol{y}$	Output
β	Batch normalization shift parameter
γ	Batch normalization scale parameter
η	Learning rate
$\boldsymbol{\theta}$	Network parameters
μ	Batch normalization mean
σ	Batch normalization standard deviation
$\mathcal{B}$	Batch normalization layer
$\mathcal{C}$	Convolutional layer
$\mathcal{D}$	Dense layer
$\mathcal{L}$	Layer

Numerical variables

s	Overlap size
G	Filter kernel/impulse response
h	Mesh size
n	Number of computing nodes
R	Downsampling ratio
x	Generic spatial coordinate
Δ	Filter size
θ	Fraction of DL overhead in the temporal iteration time
κ	Wavenumber
$\mathcal{G}$	Filter frequency response

Operators

$\bar{\cdot}$	Generic filtering
$\check{\cdot}$	Thickening

$\langle\cdot\rangle$	Statistical mean
∇	Vector differentiation
$\odot$	Elementwise operation
$\widetilde{\cdot}$	Favre-averaged filtering

Physical variables

A	Flame surface area
$\check{c}$	Thickened progress variable
c	Progress variable
c_p	Specific heat capacity at constant pressure
D_k	Mass diffusivity of species k
Da	Damköhler number
E	Efficiency function
E_a	Activation energy
e_s	Specific sensible energy
e_t	Total specific non-chemical energy
F	Thickening factor
h_s	Specific sensible enthalpy
$\underline{\underline{\boldsymbol{I}}}$	Identity tensor
$\boldsymbol{J}$	Diffusive flux vector
Ka	Karlovitz number
K	Reaction rate constant
k	Turbulent kinetic energy
Le	Lewis number
$\mathcal{M}_k$	Symbol for species k
M	Number of reactions
N	Number of species
N_c	Number of points in the thickened flame front
Pr	Prandtl number
P	Pressure
$\boldsymbol{q}$	Heat flux vector
Q	Reaction rate of progress
q_R	Specific heat of reaction
Re	Reynolds number
R	Perfect gas constant
Re_λ	Taylor-scale Reynolds number
$\underline{\underline{\boldsymbol{S}}}$	Strain rate tensor
Sc	Schmidt number
s_c	Consumption speed
s_d	Displacement speed
s_L	Unstrained laminar flame speed
T	Temperature
t	Time
$\boldsymbol{u}$	Bulk velocity vector

u_η	Kolmogorov velocity
$\boldsymbol{V}$	Diffusion velocity vector
$\boldsymbol{V}_c$	Correction velocity in the Hirschfelder-Curtiss approximation
W	Molecular weight
$[X]$	Molar concentration
X	Mole fraction
Y	Mass fraction
Z	Mixture fraction
α	Thermal diffusivity
β	Charlette model exponent
γ	Heat capacity ratio
Γ	Gamma function
δ	Diffusive laminar flame thickness
δ_L	Thermal laminar flame thickness
$\delta_{L,c}$	Progress variable laminar flame thickness
$\hat{\Delta}$	Test-filter size (Charlette model)
Δh_f^0	Specific formation enthalpy
Δ_m	Averaging filter size (Charlette model)
ϵ	Dissipation rate
ζ	Flame interaction sensor
η	Kolmogorov lengthscale
κ	Flame front curvature
λ	Thermal conductivity
λ_g	Taylor microscale
μ	Dynamic viscosity
ν	Kinematic viscosity
$\nu_{k,j}$	Stoichiometric coefficient of species k in reaction j
Ξ	Subgrid-scale wrinkling factor
ξ	Reduced spatial coordinate
ρ	Density
$\overline{\Sigma}$	Total flame surface density
$\overline{\Sigma}^+$	Normalized total flame surface density
σ	Maximum total flame surface density on a filtered 1D laminar flame
Σ	Flame surface density
$\underline{\underline{\boldsymbol{\tau}}}$	Viscous stress tensor
τ	Eddy turnover time
τ_η	Kolmogorov time
τ_F	Flame time scale
ϕ	Equivalence ratio
ψ	Generic physical quantity
$\boldsymbol{\omega}$	Vorticity
$\dot{\omega}$	Mass production rate
$\dot{\omega}_T$	Heat release rate
ω^+	Normalized vorticity magnitude

$\mathcal{A}$	Pre-exponential constant
$\mathcal{E}$	Energy spectrum function

Subscripts and superscripts

$'$	Fluctuation
b	Burnt
F	Fuel
O	Oxydizer
t	Turbulent
u	Unburnt

Chapter 1

Introduction

Contents

1.1 Future challenges in combustion

The 21st century will present critical challenges for combustion. Mitigating the long-term impact of climate change hinges on an urgent transformation of our societies. The rise of global surface temperatures is one of the key physical markers of climate change, with cascading consequences on natural catastrophes, sea level rise, biodiversity, and food security. Increasing temperatures starting from the second half of the 20th century are fully explained by higher concentrations of atmospheric greenhouse gases due to anthropogenic activities. Quoting from the Summary for Policymakers of the Working Group I contribution on the Intergovernmental Panel on Climate Change Sixth Assessment Report [256]:

> *This Report reaffirms with high confidence the AR5 finding that there is a near-linear relationship between cumulative anthropogenic CO_2 emissions and the global warming they cause. Each 1000 $GtCO_2$ of cumulative CO_2 emissions is assessed to likely cause a* 0.27 °C *to* 0.63 °C *increase in global surface temperature with a best estimate of* 0.45 °C.

Figure 1.1 illustrates this near-linear relationship, along with five shared socioeconomical pathways for the future. Limiting the global surface temperature increase to 1.5 °C or 2 °C calls for immediate action to curb cumulative CO_2 emissions within a strict carbon budget.

Energy is the lifeblood of modern societies, providing power, heating, and transportation. Yet, energy consumption accounts for three quarters of global greenhouse gas emissions [311]. Reducing global emissions must involve a prompt reform of the energy sector. In 2021, fossil fuels still made up 80% of the energy consumed in the world (Figure 1.2). Faced with this conundrum, it appears that improving the efficiency of the combustion systems involved in oil-, gas-, and coal-powered energy processes is a fundamental concern for the near future.

Today, numerical simulations play a key role in the design of combustion applications as they enable cost-efficient, rapid design iterations. They discretize turbulent reactive flows on a computational grid and predict their evolution by iterative time advancement. Two

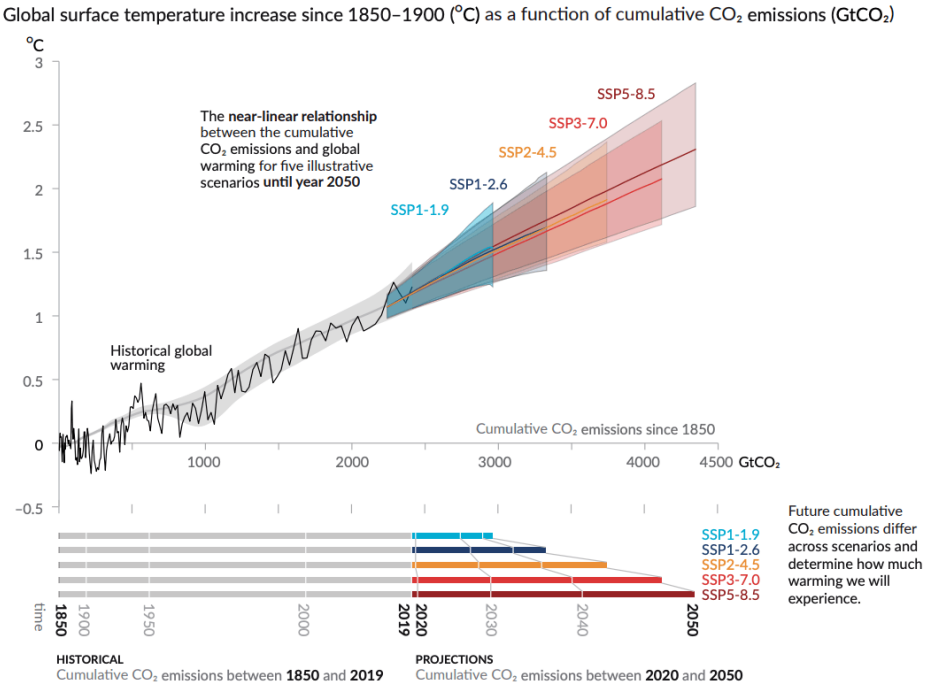


Figure 1.1: Relationship between cumulative CO_2 emissions and increase in global surface temperature from 1850 to 2019. From IPCC 2021, Figure SPM.10 [256].

elements make numerical combustion particularly computationally intensive. First, all practical combustion applications are turbulent, since turbulence greatly enhances the rate of energy transfer. Turbulent flows contain small-scale eddies that are central to the physics of turbulent reactive flows. Second, combustion involves chemical reactions happening at sub-millimetric scales. Accurate turbulent combustion simulations must therefore account for complex physical phenomena that are several orders of magnitude smaller than the extent of large combustion systems.

Large eddy simulations (LES) are able to fully resolve turbulent reactive flows up to the resolution of the computational grid. They rely on subgrid-scale (SGS) models to reproduce the effect of unresolved phenomena. Thanks to advances in hardware efficiency, numerical methods, and SGS modeling, LES is becoming a viable solution to simulate large-scale combustion systems. Still, most state-of-the-art LES computations are performed on reduced-scale configurations [26, 59, 88, 224, 287] and industrial-scale cases are still out of reach. Working towards the improvement of the computational efficiency of LES and the accuracy of SGS models at coarse resolutions could unlock full-scale LES of realistic combustors.

The decarbonization of the energy sector will also create new opportunities for combustion. To manage the intermittency of solar and wind power production, energy storage infrastructure will need to grow. Electric cars are poised to dominate the passenger vehicle market, with European and Californian bans on the sale of new internal combustion engine cars from 2035. These applications rely on battery electricity storage, typically in lithium-ion batteries today. Stationary battery storage is projected to grow from 2 GW in 2017 to 235 GW in 2030 [297]. Lithium-ion batteries are prone to thermal runaway, a sudden

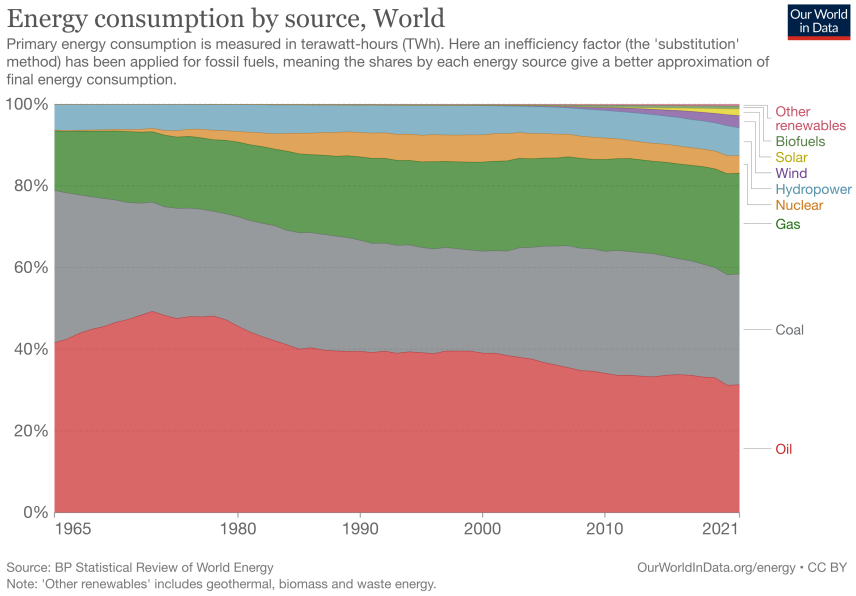


Figure 1.2: Breakdown of global energy consumption by source. From Our World In Data [312] licensed under CC BY 4.0.

increase in temperature that can lead to spontaneous combustion [94]. Understanding this mechanism via numerical simulations is paramount to the safety of transportation and power generation.

Hydrogen is a carbon-free fuel that could play a vital role in a decarbonized society. An electrical grid reliant on renewable energies will need long-term storage to tackle interseasonal variations in power supply and demand. In peak production periods, transforming excess electrical energy into chemical energy for long-term storage, a concept known as power-to-gas, could be economically viable if renewable energy penetration reaches 80% [352]. In applications like transportation or heating, hydrogen could be used for electricity storage via fuel cells or be combusted directly, for instance by blending with natural gas. Hydrogen also has the potential to decarbonize high-temperature industrial processes for iron, steel, and cement production. But hydrogen is a dangerous molecule: it is prone to leaking, odorless and colorless, gaseous at ambient conditions, very flammable, and burns extremely quickly due to synergistic interactions with turbulence [27]. Numerical combustion could greatly facilitate the design of safe storage and distribution systems for hydrogen.

Finally, combustion must adapt to a world where data is abundant. The availability of sensor data and low-power computing devices encourages systematic data collection and processing in combustion systems [298]. This could help detect problematic events like combustion instabilities [42] or guide preventive maintenance. Direct numerical simulations of reacting flows now commonly exceed billions of degrees of freedom [27, 209, 395]. Deep learning is a promising solution to extract useful insights from these vast quantities of data.

1.2 Deep learning

Deep learning introduces expressive statistical models (deep neural networks) that are able to automatically learn complex patterns from extensive amounts of data. Early successes of deep learning come from computer vision, where deep neural networks quickly became the best-performing models for image classification [137, 176, 356]. Since then, advances in deep learning have enabled complex vision tasks like generating photorealistic images [155], learning useful representations from videos [18, 227], and rendering large 3D scenes [228]. Deep learning has been successfully applied to a wide range of data modalities. Deep neural networks are now able to synthesize music [73, 145] and speech [306, 334] with long-term coherence. Large language models with more than 10^{11} parameters like GPT-3 [38] and PaLM [60] are trained on huge textual corpora, and produce convincing results in machine translation, text summarization, question answering, and even code generation [50]. Large language models have been shown to perform some form of reasoning on mathematical problems [189], and can explain answer questions with a step-by-step explanation of their logical chain of thought [170]. Lately, multi-modal models that combine various data types have unlocked increasingly complex tasks. Deep learning for speech recognition, the transformation of an audio signal into language, is nearing human-level performance [293]. Models jointly trained on text and image data [292] have opened new creative avenues for text-guided image generation. In the past year only, Stable Diffusion [314], DALL-E 2 [299], Flamingo [5], and Imagen [320] are some examples of generative models that have reached a quality and flexibility in image generation approaching human-level abilities. As an example, an illustration of the theme of this thesis generated by DALL-E 2 is shown in Figure 1.3.

Figure 1.3: Illustration generated by the DALL-E 2 [299] text-to-image generative model from the prompt "*Numerical simulation of an orange photorealistic turbulent flame with a neural network*".

Deep learning is firmly on its way to accelerating scientific progress. AlphaFold deals with the long-standing problem of predicting the 3D structure of proteins from their se-

quence of amino acids significantly faster than traditional computational methods [154]. In high-energy physics, deep learning is routinely used to analyze the enormous quantity of data generated by particle colliders [124]. Deep neural networks can enhance weather forecasts by accurately predicting precipitations within short time horizons [301]. Closer to the applications of this thesis, deep learning is seeing growing adoption in models for computational fluid dynamics [80] and combustion simulations specifically [146].

1.3 Objective and organization of the thesis

This thesis is motivated by the following research question:

How can deep learning be applied to SGS modeling in practical LES of turbulent premixed flames?

The organization of the thesis is summarized in Figure 1.4. Numerical combustion and deep learning are two domains that are presented in Part I. Chapter 2 recalls the theoretical foundations of large eddy simulations of turbulent premixed flames. Chapter 3 covers the deep learning notions underpinning the multiscale convolutional neural networks used throughout the thesis. Chapter 4 then explores the intersection between these two domains, starting from a review of existing deep learning applications to numerical combustion. It highlights three missing elements from the existing literature that are necessary to move towards applying deep learning in practical LES: evaluating models on challenging high Reynolds test cases, choosing different configurations for the training and test sets to assess generalization, and efficiently integrating deep learning models to high-performance LES solvers. These are the three axes that underlie the construction of this thesis. Chapter 4 also presents the methods used to train and evaluate deep learning SGS models, and the overarching modeling framework of this thesis.

Part II presents two SGS models that are evaluated *a priori*. In Chapter 5, a model for the total flame surface density proposed by Lapeyre et al. [183] is evaluated on a highly turbulent jet flame, with a focus on interpreting the predictions and learning process of the model. Chapter 6 trains a model for the SGS progress variable variance and investigates its generalization from an academic training configuration to a different challenging test case.

Part III is devoted to *a posteriori* validation of deep learning models. In practice, it requires a coupling between an LES solver and the model which must solve hardware and software constraints in a computationally efficient manner. This is handled by the AVBP-DL coupling scheme presented in Chapter 7. Finally, Chapter 8 evaluates a model for the SGS wrinkling factor *a posteriori* in the LES of a vented explosion in an obstructed chamber.

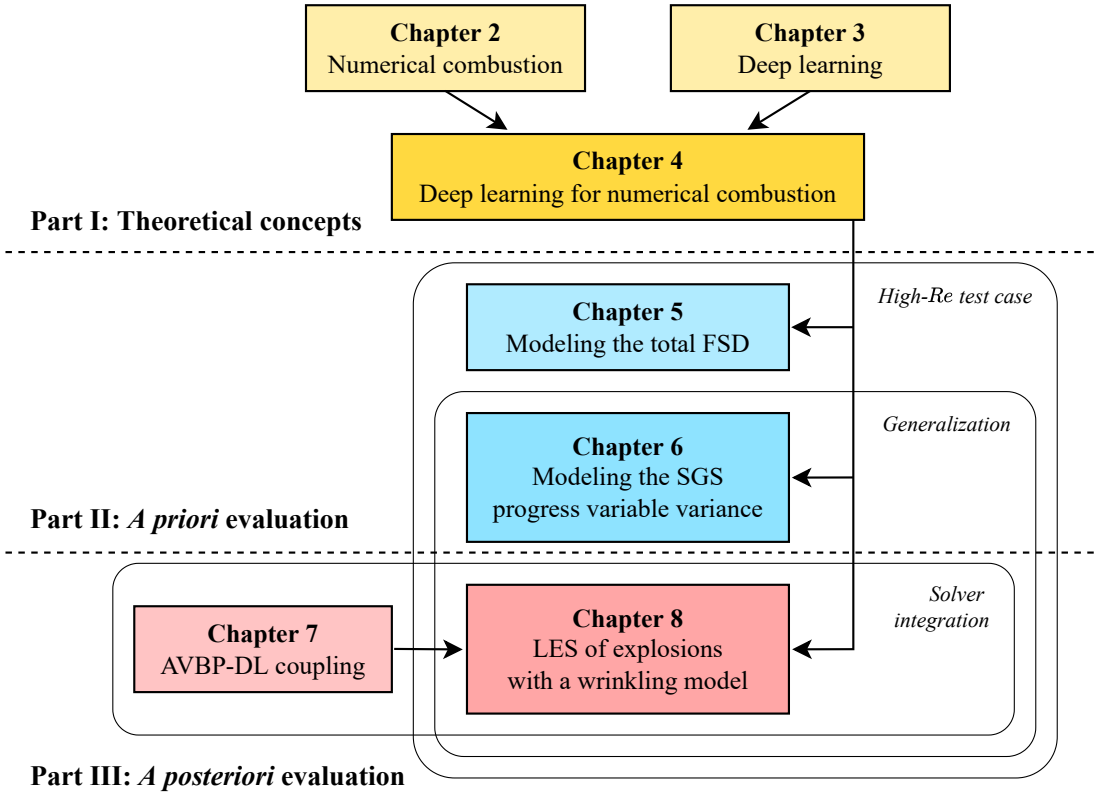


Figure 1.4: Organization of the thesis.

Part I

Theoretical concepts

Chapter 2

Turbulent premixed combustion

Contents

This chapter presents theoretical concepts that underpin the numerical simulations that are carried out in this work. The governing equations and constitutive relations of multispecies reacting flows are first recalled. Notions of laminar premixed combustion, turbulence, turbulent premixed combustion, and large eddy simulations are then introduced. Since each of these topics is too vast to cover exhaustively, classical books are referenced throughout this overview. Combining all these notions leads to an overview of LES for turbulent premixed combustion, with a specific focus on subgrid-scale models to close the filtered reaction rate. Finally, the AVBP solver used for all the numerical simulations of this thesis is presented.

2.1 Governing equations of multispecies reacting flows

Consider a compressible, multispecies, reacting, gaseous mixture of N chemical species. The local composition of the mixture is parametrized by the **species mass fractions** $(Y_k)_{k=1,...,N}$, defined as the ratio of the mass of species k to the total mass of the mixture in a small control volume. By definition, the species mass fractions sum to unity:

$$\sum_{k=1}^{N} Y_k = 1\,. \tag{2.1}$$

From the velocity fields $\boldsymbol{u}_k$ of all the species, a mass-averaged bulk velocity is defined as [178]

$$\boldsymbol{u} = \sum_{k=1}^{N} Y_k \boldsymbol{u}_k\,, \tag{2.2}$$

and is simply called the velocity field of the mixture. The difference between the velocity $\boldsymbol{u}_k$ of an individual species k and the mixture velocity is the **diffusion velocity** $\boldsymbol{V}_k = \boldsymbol{u}_k - \boldsymbol{u}$.

Balance equations for a set of primitive variables composed of:

- the density ρ, which is the local mass of the mixture per unit volume,
- the velocity field $\boldsymbol{u}$,
- the total specific non-chemical energy e_t,
- and the species mass fractions $(Y_k)_{k=1,...,N}$,

fully describe the spatio-temporal evolution of the mixture. In conservative form, they read [185]:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0\,, \tag{2.3}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) = \nabla \cdot \underline{\underline{\boldsymbol{\sigma}}}\,, \tag{2.4}$$

$$\frac{\partial \rho e_t}{\partial t} + \nabla \cdot (\rho e_t \boldsymbol{u}) = \nabla \cdot (\underline{\underline{\boldsymbol{\sigma}}} \cdot \boldsymbol{u} - \boldsymbol{q}) + \dot{\omega}_T\,, \tag{2.5}$$

$$\frac{\partial \rho Y_k}{\partial t} + \nabla \cdot (\rho Y_k \boldsymbol{u}) = -\nabla \cdot \boldsymbol{J}_k + \dot{\omega}_k\,, \quad \text{for } k = 1, ..., N, \tag{2.6}$$

where $\underline{\underline{\boldsymbol{\sigma}}}$ is the Cauchy stress tensor, $\boldsymbol{q}$ the heat flux vector, $\dot{\omega}_T$ the heat release rate, $\boldsymbol{J}_k$ the diffusive flux vector of species k, and $\dot{\omega}_k$ the mass production rate of species k. Radiative heat transfer and volumic body forces such as gravity are omitted.

2.2 Constitutive relations

Equations 2.3 to 2.6 are complemented by the following constitutive relations.

Equation of state All the species are assumed to behave like perfect gases. The partial pressure P_k of species k is related to its density $\rho_k = \rho Y_k$, its atomic weight W_k, the perfect gas constant $R = 8.314\,\mathrm{J\,mol^{-1}\,K^{-1}}$, and the temperature T through the equation of state

$$P_k = \rho_k \frac{R}{W_k} T \,. \tag{2.7}$$

The total thermodynamic pressure P is equal to the sum of partial pressures (Dalton's law), leading to

$$\boxed{P = \rho \frac{R}{W} T \,,} \tag{2.8}$$

with $W = 1/\left(\sum_{k=1}^{N} Y_k/W_k\right)$ the mixture molecular weight.

Stress tensor The Cauchy stress tensor $\underline{\underline{\boldsymbol{\sigma}}}$ is commonly decomposed as the sum of an isotropic term $-P\underline{\underline{\boldsymbol{I}}}$, which is the stress tensor for a fluid at rest, and of a viscous stress tensor $\underline{\underline{\boldsymbol{\tau}}}$ generated by the motion of the fluid [22]:

$$\underline{\underline{\boldsymbol{\sigma}}} = -P\underline{\underline{\boldsymbol{I}}} + \underline{\underline{\boldsymbol{\tau}}} \,. \tag{2.9}$$

The gas is presumed to behave like a Newtonian fluid, meaning that viscous stresses are proportional to the strain rate tensor $\underline{\underline{\boldsymbol{S}}} = (\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)/2$. Assuming that the fluid is isotropic and that the bulk viscosity term is negligible, an expression for $\underline{\underline{\boldsymbol{\tau}}}$ can be derived [22]:

$$\boxed{\underline{\underline{\boldsymbol{\tau}}} = 2\mu \left[\underline{\underline{\boldsymbol{S}}} - \frac{1}{3}(\nabla \cdot \boldsymbol{u})\underline{\underline{\boldsymbol{I}}}\right] \,.} \tag{2.10}$$

μ is the dynamic viscosity, which mainly depends on temperature. Models for μ are discussed in Section 2.8

Total energy Species specific sensible enthalpies $h_{s,k}$ quantify how enthalpy changes with temperature:

$$h_{s,k}(T) = \int_{T_0}^{T} c_{p,k}\,\mathrm{d}T \,, \tag{2.11}$$

with $c_{p,k}$ the species specific heat capacity at constant pressure, and T_0 a reference temperature usually set to $298.15\,\mathrm{K}$ [275]. Specific sensible energies $e_{s,k}$ are defined as

$$e_{s,k} = h_{s,k} - \frac{RT}{W_k} \,. \tag{2.12}$$

The specific sensible energy of the whole mixture is thus equal to

$$e_s = \sum_{k=1}^{N} e_{s,k} Y_k = \int_{T_0}^{T} c_p\,\mathrm{d}T - \frac{RT}{W} \,. \tag{2.13}$$

There are many viable choices for an energy or enthalpy variable [275]. This work will use the **total specific non-chemical energy** e_t which adds the specific sensible and kinetic energies:

$$\boxed{e_t = e_s + \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}\,.} \tag{2.14}$$

Heat flux The heat flux $\boldsymbol{q}$ originates from heat diffusion through conduction and energy transport due to the diffusion of species with different sensible enthalpies. Using Fourier's law for the conduction term, $\boldsymbol{q}$ writes:

$$\boxed{\boldsymbol{q} = -\lambda\nabla T + \sum_{k=1}^{N}\boldsymbol{J}_k h_{s,k}\,,} \tag{2.15}$$

with λ the thermal conductivity. Following common practice [185], heat transfer due to mass diffusion, also called the Dufour effect, is neglected.

Heat release rate The heat release rate $\dot{\omega}_T$ measures the volumic heat released by the production and destruction of species with different formation enthalpies in chemical reactions:

$$\boxed{\dot{\omega}_T = -\sum_{k=1}^{N}\Delta h^0_{f,k}\dot{\omega}_k\,,} \tag{2.16}$$

with $\Delta h^0_{f,k}$ the specific formation enthalpy of species k at the reference temperature T_0.

Diffusive flux The diffusive flux vector $\boldsymbol{J}_k$ of species k is expressed as

$$\boxed{\boldsymbol{J}_k = \rho Y_k \boldsymbol{V}_k\,.} \tag{2.17}$$

The diffusion velocity $\boldsymbol{V}_k$ is influenced by the diffusion of other species induced by composition and pressure gradients as well as thermodiffusion (also known as the Soret effect) which refers to mass diffusion created by a temperature gradient. The Soret effect is known to be crucial in combustion simulations when light molecules such as atomic and molecular hydrogen are included in the chemical mechanisms [87]. In this thesis, only reduced chemical mechanisms for simple hydrocarbon fuels which do not contain such light species will be used. Accordingly, the Soret effect will be omitted without significant repercussions.

The equation for the diffusion velocity $\boldsymbol{V}_k$ thus writes [178]:

$$X_k\boldsymbol{V}_k = -\sum_{j=1}^{N} D_{k,j}\left[\nabla X_j + (X_j - Y_j)\frac{\nabla P}{\rho}\right]\,, \tag{2.18}$$

with $X_k = Y_k W/W_k$ the mole fraction of species k, and $D_{k,j}$ the diffusion coefficient of species j into species k. The pressure term is usually neglected. The multicomponent diffusion matrix $\underline{\underline{\boldsymbol{D}}} = (D_{k,j})_{k,j}$ is not given by the kinetic theory of gases and requires the

resolution of an $N \times N$ system [86] which is prohibitively computationally expensive in usual combustion simulations.

Instead, the Hirschfelder-Curtiss approximation [141] is a common simplifying alternative to estimate $D_{k,j}$. It expresses the diffusion velocity as

$$\boxed{\boldsymbol{V}_k = -\frac{D_k}{X_k}\nabla X_k + \boldsymbol{V}_c\,,} \tag{2.19}$$

where D_k is the diffusion coefficient of species k into the rest of the mixture:

$$D_k = \frac{1 - Y_k}{\sum_{j=1, j\neq k}^{N} X_j / D_{k,j}}\,, \tag{2.20}$$

and $\boldsymbol{V}_c$ is a correction velocity that ensures mass conservation:

$$\boxed{\boldsymbol{V}_c = \sum_{k=1}^{N} D_k \frac{W_k}{W}\nabla X_k\,.} \tag{2.21}$$

Note that there exists a rigorous theoretical justification for the Hirschfelder-Curtiss approximation, since it is possible to exhibit a matrix series that converges towards the multicomponent diffusion matrix, and whose truncation at the zeroth-order leads to the Hirschfelder-Curtiss approximation [113].

Transport coefficients The interplay between the diffusional transport of momentum, species, and heat plays a key role in combustion processes. Their characteristic rates of evolution are the kinematic viscosity $\nu = \mu/\rho$, the species diffusivities D_k, and the thermal diffusivity $\alpha = \lambda/(\rho c_p)$. To compare the magnitude of these three effects, it is convenient to introduce the following non-dimensional numbers:

- the Prandtl number, comparing momentum and heat diffusion

$$Pr = \frac{\nu}{\alpha} = \frac{\mu c_p}{\lambda}\,, \tag{2.22}$$

- the Schmidt number, comparing momentum and species diffusion

$$Sc = \frac{\nu}{D_k}\,, \tag{2.23}$$

- and the Lewis number, comparing heat and species diffusion

$$Le = \frac{\alpha}{D_k} = \frac{\lambda}{\rho c_p D_k}\,. \tag{2.24}$$

These numbers are presumed to be constant in time and space [275].

Species production rate Let M be the number of elementary chemical reactions involved in the system. Reaction j is written in the form:

$$\sum_{k=1}^{N} \nu'_{k,j} \mathcal{M}_k \rightleftharpoons \sum_{k=1}^{N} \nu''_{k,j} \mathcal{M}_k \,, \tag{2.25}$$

where $\nu'_{k,j}$ and $\nu''_{k,j}$ are the stoichiometric coefficients of species k, and $\mathcal{M}_k$ denotes the species k. According to the law of mass action, the rate of progress of a single reaction is proportional to the product of reactant molar concentrations $[X_k] = \rho Y_k / W_k$. The rate of progress Q_j therefore writes:

$$Q_j = K_{f,j} \prod_{k=1}^{N} [X_k]^{\nu'_{k,j}} - K_{r,j} \prod_{k=1}^{N} [X_k]^{\nu''_{k,j}} \,. \tag{2.26}$$

$K_{f,j}$ and $K_{r,j}$ are the forward and reverse rate constants which need to be modeled. Their most common expression is given by the Arrhenius law:

$$\boxed{K = \mathcal{A} T^{\beta_j} \exp\left(-\frac{E_{a,j}}{RT}\right) \,,} \tag{2.27}$$

where $\mathcal{A}$ the pre-exponential constant, β_j the temperature exponent, and $E_{a,j}$ the activation energy are tabulated.

Finally, the mass production rate of species k writes:

$$\boxed{\dot{\omega}_k = W_k \sum_{j=1}^{M} \nu_{k,j} Q_j \,,} \tag{2.28}$$

where $\nu_{k,j} = \nu''_{k,j} - \nu'_{k,j}$ is the net stoichiometric coefficient of species k in reaction j.

2.3 Laminar premixed flames

The fundamentals of premixed combustion, where reactants (fuel and oxidizer) are fully premixed before combustion, are usually laid out in the context of a 1D laminar unstretched flame [178, 185, 275]. Laminar flames indeed play a key role in understanding turbulent premixed flame fronts via the **flamelet** assumption, where local flame elements behave like 1D laminar flamelets, as detailed in Section 2.5. In the following, the subscripts and superscripts F, u, b will denote quantities related to the fuel, unburnt state, and burnt state, respectively.

2.3.1 Phenomenology of a 1D laminar premixed flame

The structure of a 1D laminar premixed flame is illustrated in Figure 2.1. In this simple case, chemical reactions are modeled as a single global irreversible reaction Reactants → Products. Chemical activity is concentrated in a narrow region marked by the reaction rate peak, where the temperature is sufficiently high for the reaction to occur (Equation 2.27). This

is the **reaction zone**, which is controlled by the balance between diffusive transport of the reactants that enter this zone and the burning rate of these reactants. This reaction zone naturally propagates towards the unburnt state, leaving behind a burnt mixture of combustion products and reactants, depending on the stoichiometry, at the adiabatic flame temperature T_b. For a lean mixture with unity Lewis numbers, T_b is given by:

$$T_b = T_u + \frac{q_R Y_F^u}{c_p} \tag{2.29}$$

where $q_R = -\dot{\omega}_T/\dot{\omega}_F$ is the specific heat of reaction and c_p has been assumed to be constant.

Incoming reactants are preheated by the thermal conduction of the heat released by the reaction. This forms a **preheat zone** upstream of the reaction zone where temperature gradually rises from the unburnt temperature to the reaction zone temperature. This region is piloted by the interaction between convection and diffusion of the reactants.

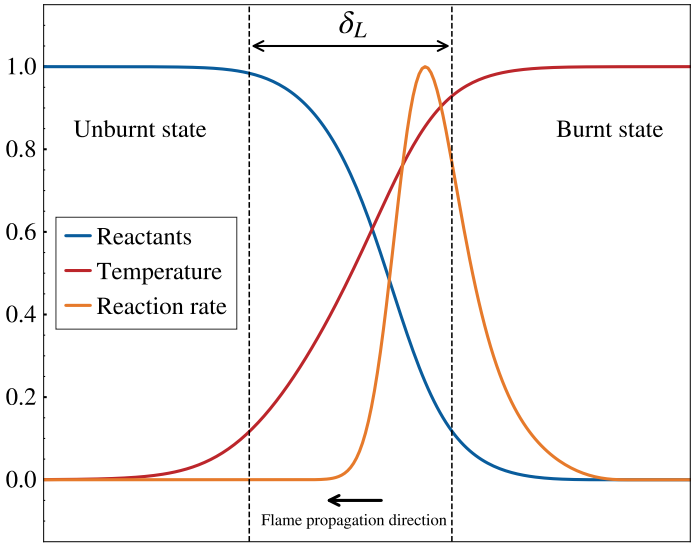


Figure 2.1: Structure of a 1D laminar premixed flame. The temperature, reactant mass fractions, and reaction rate profiles are normalized by their unburnt and burnt values.

Overall, the preheat zone and the reaction zone form a flame front which propagates at a **laminar flame speed** defined by the consumption rate of the fuel [275]:

$$s_L = -\frac{1}{\rho_u Y_F^u} \int_{-\infty}^{+\infty} \dot{\omega}_F \, \mathrm{d}x \, . \tag{2.30}$$

Its **thermal thickness** is defined via the steepest gradient of temperature:

$$\delta_L = \frac{T_b - T_u}{\max\left(dT/dx\right)} \tag{2.31}$$

which is useful to define mesh resolution constraints to properly resolve the flame. Two other definitions of the flame thickness are:

- the diffusive thickness δ, based on the thermal diffusivity α and used in some theoretical developments:
$$\delta = \frac{\alpha_u}{s_L}\,, \tag{2.32}$$
- and the progress variable thickness $\delta_{L,c}$:
$$\delta_{L,c} = \frac{1}{\max\left(dc/dx\right)} \tag{2.33}$$
which can be different from δ_L if c is not defined as a normalized temperature. This definition will be used in Section 4.3 to characterize the thickness of the flame front relevant to the DL model.

s_L and δ_L have square root dependencies on the magnitude of the diffusion and reaction terms [275]:

$$s_L \propto \sqrt{\alpha \mathcal{A}} \tag{2.34}$$
$$,\delta_L \propto \sqrt{\frac{\alpha}{\mathcal{A}}}\,, \tag{2.35}$$

where $\mathcal{A}$ is the pre-exponential constant for the single-step reaction.

2.3.2 Theoretical analysis

A governing equation for the 1D laminar flame is now derived, with more details contained in Poinsot and Veynante [275]. Some simplifying assumptions are made: all species have same a unity Lewis number, reactions are grouped in a single-step irreversible global reaction, and the fuel is the deficient reactant, *i.e.* combustion is lean and the equivalence ratio ϕ is below unity. These hypotheses preserve the main physical phenomena while facilitating a purely analytical approach. The evolution of the flame is controlled by a single parameter, the **progress variable** $c = (T - T_u)/(T_b - T_u) = 1 - Y_F/Y_F^u$. This non-dimensional quantity parametrizes the whole premixed flame structure. Its conservation equation reads:

$$\rho u \frac{\partial c}{\partial x} = \frac{\partial}{\partial x}\left(\rho D \frac{\partial c}{\partial x}\right) - \frac{\dot{\omega}_F}{Y_F^u}\,, \tag{2.36}$$

where $D = \alpha = D_k$ is the diffusivity. A reduced spatial variable ξ is introduced:

$$\xi = \int_0^x \frac{\rho_u s_L}{\rho(y) D(y)}\,\mathrm{d}y\,. \tag{2.37}$$

Noting that

$$\frac{\partial}{\partial x} = \frac{\partial \xi}{\partial x}\frac{\partial}{\partial \xi} = \left(\frac{\rho_u s_L}{\rho D}\right)\frac{\partial}{\partial \xi} \tag{2.38}$$

and using the conservation of mass $\rho u = \rho_u s_L$, Equation 2.36 becomes:

$$\boxed{\frac{\partial c}{\partial \xi} = \frac{\partial^2 c}{\partial \xi^2} - \frac{\rho D}{(\rho_u s_L)^2 Y_F^u}\dot{\omega}_F\,.} \tag{2.39}$$

Grouping the constants into the factor $\Lambda = (\rho_u s_L)^2 Y_F^u$, the fuel production rate $\dot{\omega}_F$ is therefore linked to the non-dimensional source term ω by:

$$\dot{\omega}_F = -\frac{\Lambda}{\rho D}\omega \,. \tag{2.40}$$

2.3.3 Pfitzner source term

While the Arrhenius law (Equation 2.27) is precise, it is also cumbersome and combustion theoreticians have looked to derive simpler expressions for the species reaction rates [96, 275]. Recently, Pfitzner et al. [269, 270] proposed such an expression. They showed that $c_m(\xi) = [1 + e^{-m\xi}]^{-1/m}$ is a solution of Equation 2.39 parametrized by a model coefficient m. It corresponds to the non-dimensional source term

$$\omega_m(c) = (m+1)(1-c^m)c^{m+1} \,. \tag{2.41}$$

The dimensional Pfitzner source term therefore writes:

$$\boxed{\dot{\omega}_{F,m} = -\frac{\Lambda}{\rho D}(m+1)(1-c^m)c^{m+1} \,.} \tag{2.42}$$

This is an alternative to the exponential Arrhenius reaction rate (Equation 2.27) which has a polynomial expression in c and therefore T. It assumes that chemistry can be reduced to a global single-step irreversible reaction.

The laminar flame speed is given in the factor Λ, and for any m value, it is consistent with the definition of the laminar flame speed as a consumption speed:

$$-\frac{1}{\rho_u Y_F^u}\int_{-\infty}^{\infty} \dot{\omega}_{F,m}(x)\,\mathrm{d}x = s_L \,. \tag{2.43}$$

The progress variable flame thickness is inversely proportional to m and has the following analytical expression in ξ space:

$$\delta_{L,c} = \frac{(m+1)^{\frac{m+1}{m}}}{m} \,. \tag{2.44}$$

The Pfitzner formulation can be a substitute to any flame computed with Arrhenius chemistry by matching independently the flame speed with the Λ parameter and the flame thickness with the m parameter. Figure 2.2 compares the fuel consumption rate profiles for Pfitzner and Arrhenius source terms in the example of a lean methane-air mixture at elevated pressure and temperature. It illustrates two possible use cases of the Pfitzner source term. In this example, choosing $m = 5$ provides a good approximation of the Arrhenius source term. This allows complex skeletal mechanisms to be replaced by a single-step mechanism that uses the Pfitzner source term, which could lead to massively increased computational efficiency if only the relationship between the progress variable and the reaction rate is needed. On the other hand, lower values of m can be chosen to thicken the reaction zone. This could allow computational savings by relaxing mesh resolution requirements, especially for LES of

large premixed systems where a perfect description of complex chemistry is not needed [290]. Finally, the polynomial nature of the Pfitzner source term makes it amenable to analytical turbulent closures. This property will be used in Chapter 6.

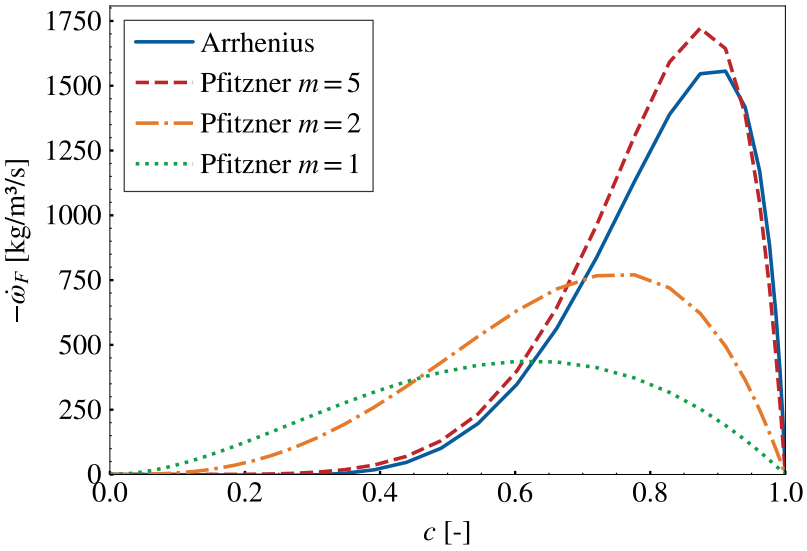


Figure 2.2: Arrhenius and Pfitzner fuel consumption rate for a lean ($\phi = 0.7$) methane-air mixture at $P = 4\,\text{bar}$, $T_u = 800\,\text{K}$. The Arrhenius rate is derived from a skeletal mechanism [208].

2.4 Turbulence

J. O. Hinze proposes the following definition of turbulent flows [140]:

> *Turbulent fluid motion is an irregular condition of flow in which the various quantities show a random variation with time and space coordinates, so that statistically distinct average values can be discerned.*

In this definition, *randomness* should be understood as the variability of the velocity field $\boldsymbol{u}(\boldsymbol{x}, t)$ over repetitions of the same experiment. This is a consequence of the sensitivity of the Navier-Stokes equations 2.3 and 2.4 to small perturbations that inevitably perturb the flow [282]. The stochastic nature of turbulent flows leads to enhanced diffusivities of mass, momentum, and heat, making them particularly desirable in engineering applications where the flow acts as an energy carrier.

A key indicator of the laminar or turbulent nature of a flow is the Reynolds number Re:

$$\boxed{Re = \frac{||\boldsymbol{u}||L}{\nu}}, \tag{2.45}$$

where L is a characteristic size of the flow. At low Reynolds numbers, viscous forces dominate inertial effects, and the flow is laminar[1]. For high Reynolds numbers, inertial effects lead to

[1]Named so because of the image of a flow moving in smooth layers, or *laminae* [393].

the creation of coherent swirls that deviate from the mean motion of the flow. These are **turbulent eddies**.

The theory of turbulence formalized by Kolmogorov [172] involves a continuous spectrum of turbulent eddy scales. As the flow evolves, the largest eddies, whose size depends on the geometry of the flow, break up into smaller eddies to whom they transfer their kinetic energy. These smaller eddies break up into yet smaller eddies and so on, a sequence named the **energy cascade**. As eddies get smaller, viscous forces start to dominate inertial effects, and eventually a balance is reached between the kinetic energy received by larger eddies and the energy dissipated through molecular viscosity. This characterizes the smallest eddies in the energy cascade. Unlike the largest eddies, motion at the smallest scales is independent of the geometry as it is driven by the equilibrium between inertial and dissipative fluxes, and happens over much shorter characteristic time scales.

The kinetic energy dissipation rate $\epsilon = 2\nu\langle \underline{\underline{\boldsymbol{S}}} : \underline{\underline{\boldsymbol{S}}} \rangle$ is constant across the scales of the turbulent cascade. It can thus be estimated by the rate of energy transfer from the large scales, which contain most of the turbulent kinetic energy (TKE) $k = \langle \boldsymbol{u} \cdot \boldsymbol{u} \rangle / 2$. A dimensional analysis leads to

$$\epsilon = \frac{k^{3/2}}{l_t}\,, \tag{2.46}$$

where l_t is the integral lengthscale that characterizes the size of the largest, energy-containing eddies.

Kolmogorov's first similarity hypothesis presumes that for a sufficiently large Reynolds number, there exists a universal equilibrium range, where the statistics of sufficiently small eddies only depend on the viscosity ν, representing dissipative effects, and the kinetic energy dissipation rate ϵ, representing energy transfer from larger eddies. The universal equilibrium range is split into an inertial subrange, where turbulence statistics only depend on ϵ (Kolmogorov's second similarity hypothesis), and a dissipative range that contains the smallest eddies, for which viscous forces overpower inertial effects and lead to viscous dissipation of the TKE. Turbulent eddies in the dissipative range are characterized by the Kolmogorov length, time, and velocity:

$$\eta = \left(\frac{\nu^3}{\epsilon}\right)^{1/4}\,, \tag{2.47}$$

$$\tau_\eta = \left(\frac{\nu}{\epsilon}\right)^{1/2}\,, \tag{2.48}$$

$$u_\eta = (\nu\epsilon)^{1/4}\,. \tag{2.49}$$

Figure 2.3 illustrates the energy cascade on the Pope model turbulence spectrum [282] which plots the energy spectrum function $\mathcal{E}$ as a function of the wavenumber κ. The TKE k is the sum of the contribution of $\mathcal{E}$ over all the wavenumbers:

$$k = \int_0^\infty \mathcal{E}(\kappa)\,\mathrm{d}\kappa\,. \tag{2.50}$$

The spectrum shows that most of the TKE is contained in the largest eddies, hence their

grouping into the energy-containing range. In the inertial subrange, the spectrum evolves proportionally to $\kappa^{-5/3}$.

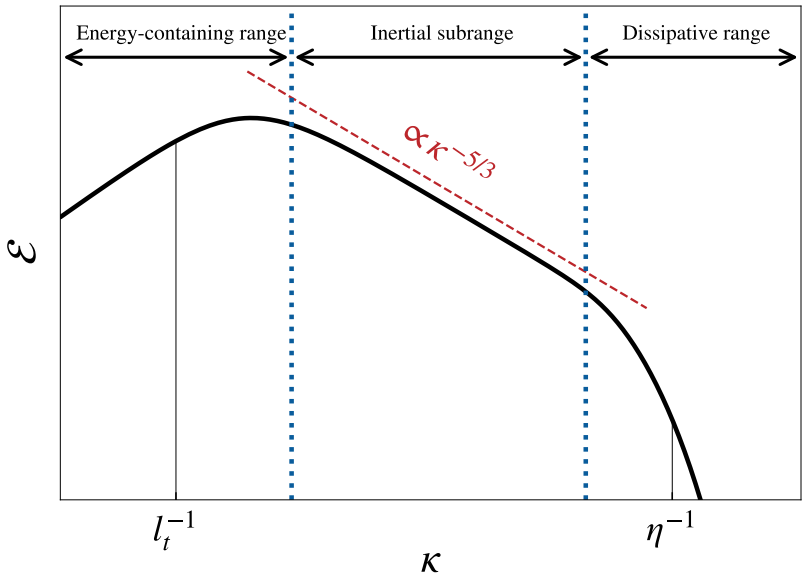


Figure 2.3: Pope model turbulence spectrum in log-log scale [282].

Figure 2.4 is a visual example of a turbulent velocity field. It shows a component of the velocity field in homogeneous isotropic turbulence (HIT). The multi-scale nature of turbulence is clearly apparent by the existence of coherent velocity regions over a wide range of lengthscales.

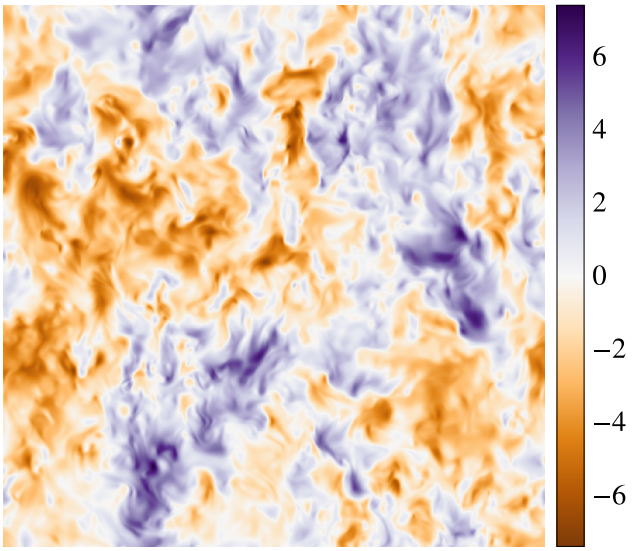


Figure 2.4: x velocity field in the z-normal plane of homogeneous isotropic turbulence ($u' = 3.8\,\mathrm{m\,s^{-1}}$, $l_t = 1.8\,\mathrm{mm}$, $t = 5.7 l_t/u'$).

2.5 Turbulent premixed flames

To understand the effects of turbulence on premixed flame propagation, it is convenient to explore the limit cases where turbulence effects are weak and strong compared to laminar flame advancement. In these conditions, Damköhler's observations on burning velocity measurements on a Bunsen burner led to two hypotheses [70, 77]:

1. In the limit of weak turbulence, where turbulent eddies are larger than the laminar flame thickness, turbulence only stretches and wrinkles the flame front without perturbing its inner structure. Accordingly, the turbulent flame speed s_T should be proportional to the area of the wrinkled flame front:

$$\frac{s_T}{s_L} \sim \frac{A_T}{A_L}\,. \tag{2.51}$$

2. In the limit of strong turbulence, where turbulent eddies are smaller than the laminar flame thickness, turbulence only affects the inner structure of the flame front in the form of an increased turbulent diffusivity D_t:

$$\frac{s_T}{s_L} \sim \sqrt{1+\frac{D_t}{D}}\,. \tag{2.52}$$

The strength of the influence of turbulence on premixed flames is assessed by comparing velocity and length scales that characterized turbulent eddies and laminar flame propagation. Two ratios are therefore introduced: u'/s_L compares the turbulence intensity $u' = \sqrt{2k/3}$, representing the turnover velocity of large eddies, to the laminar flame speed s_L, and l_t/δ compares the integral lengthscale to the diffusive laminar flame thickness $\delta = \nu_u/s_L$.

These ratios have been used to propose regime diagrams that attempt to categorize turbulent flame types with a finer granularity than Damköhler's hypotheses [34, 265–267, 276]. Figure 2.5 shows the commonly used Borghi-Peters diagram [267].

The boundaries of the regime diagram are defined by the turbulent Reynolds number of the large eddies:

$$Re_t = \frac{u' l_t}{\nu_u} = \left(\frac{u'}{s_L}\right)\left(\frac{l_t}{\delta}\right), \tag{2.53}$$

and the Karlovitz number, expressed as the ratio of the flame time $\tau_F = \delta/s_L$ to the Kolmogorov time τ_η:

$$Ka = \frac{\tau_F}{\tau_\eta} = \left(\frac{\delta}{\eta}\right)^2 = \left(\frac{u'}{s_L}\right)^{3/2}\left(\frac{l_t}{\delta}\right)^{-1/2}. \tag{2.54}$$

Boundaries in regime diagrams should be interpreted as transition regions, and not hard delimiters between distinct flame types.

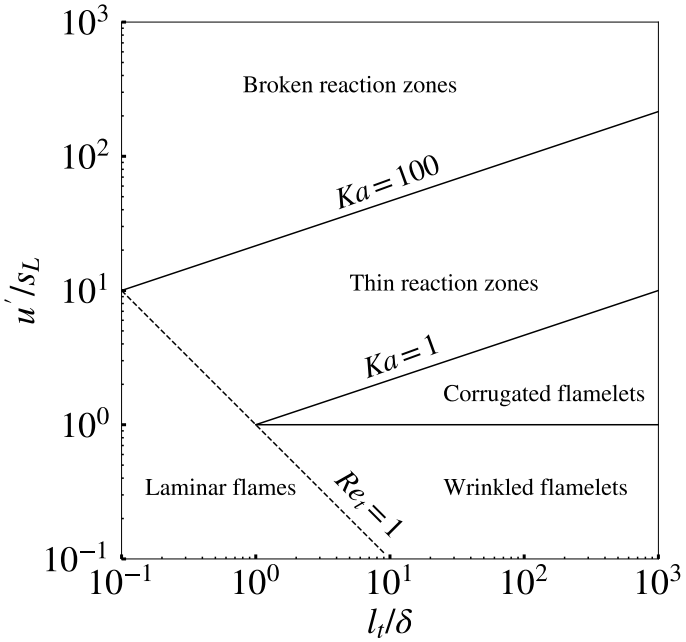


Figure 2.5: Borghi-Peters turbulent premixed combustion diagram.

The Borghi-Peters diagram distinguishes 5 turbulent flame types:

- $Re_t < 1$ corresponds to laminar flames.
- When $u'/s_L < 1$ but the integral lengthscale is sufficiently large, flame propagation is driven by the laminar burning rate rather than turbulent wrinkling. This is the wrinkled flamelet regime which is rarely encountered in practical flows.
- Wrinkled flamelets subject to higher turbulence intensities but still below the limit $Ka = 1$ (Klimov-Williams criterion) become corrugated flamelets. In this regime, $\delta < \eta$, which indicates that the smallest turbulent eddies are larger than the flame thickness and cannot penetrate into the flame front.
- Turbulent flames with a Karlovitz number $1 < Ka < 100$ belong to the thin reaction zones (TRZ) regime. The limit $Ka = 100$ corresponds to a unity Karlovitz number relative to the inner reaction zone thickness, which is approximately 10 times smaller than the diffusive thickness [267]. In this regime, the smallest eddies can therefore enter the flame front and broaden the preheat zone, thus enhancing turbulent transport of preheated gas. However, they are not small enough to enter the inner reaction zone. Peters hypothesized that the reaction zone may become thinner due to stretching from these turbulent eddies, hence the name of this regime.
- The final region is the broken reaction zones regime for $Ka > 100$, where turbulence is so intense that the smallest eddies are able to penetrate the inner reaction layer and disrupt the flame structure.

The Borghi-Peters diagram is derived purely from physical reasoning on orders of magnitude. Recent work has complemented the diagram with measurements from direct numerical simulations (DNS) and experimental results. Figure 2.6 shows the measured regime diagram of Skiba et al. [346]. They observe that broadening of the preheat zone only occurs when the turbulent diffusivity D_t is large enough compared to the preheat zone diffusivity D^*, at about $D_t/D^* = 180$. Below this limit, turbulent eddies do not have enough energy to disrupt the convective-diffusive balance that governs the structure of the preheat zone, even though the smallest eddies are small enough to penetrate it. This new thin flamelets regime is also empirically delimited by $Ka = 60$.

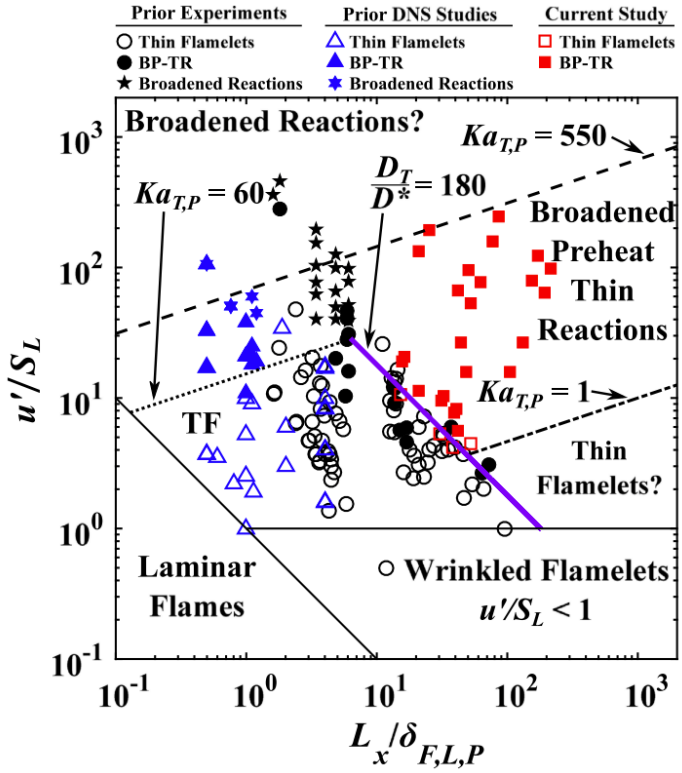


Figure 2.6: Measured premixed combustion regime diagram with DNS and experimental data points. Reprinted from Skiba et al. [346] with permission from Elsevier, license number 5364270406074.

The flamelet concept, which is used in some regions of the regime diagrams, plays a central in turbulent premixed combustion theory. Flamelets are regions of the flame front in which chemical reactions take place in thin layers that are wrinkled but not fragmented by turbulence [265]. Chemical timescales are assumed to be fast compared to turbulent processes so that the effects of turbulence can be treated independently from the chemistry. Under these assumptions, the evolution of thermochemical variables can be tracked by a single scalar quantity, the progress variable c, which increases monotonically from 0 in the unburnt state to 1 in the burnt state. In a flamelet, the relationship between a thermochemical quantity ψ such as temperature and species mass fractions and the progress variable is, on average, the

same as in a laminar flame. In practice, this is estimated by comparing the conditional mean profile $\langle\psi|c\rangle$ to a laminar flame profile [77]. It is also said that the structure of a flamelet is that of a laminar flame.

The extent of the flamelet regime is still a matter of active research and debate [77, 161]. There is evidence that high-Karlovitz number flames up to the upper end of the TRZ can still preserve a flamelet structure [12, 14, 252, 346]. On the other hand, several recent DNS studies have observed thickening of the reaction zone in the TRZ, a marker of non-flamelet behavior which leads to an enhancement of the turbulent consumption speed that cannot be explained by an increase in flame surface [15, 161, 231]. Nonetheless, even in the latter works, turbulent **wrinkling** remains the leading order effect that drives the increase of the turbulent consumption speed [15]. Wrinkling effects can be decoupled from turbulent diffusion and flame structure alterations by rewriting the turbulent consumption speed as [37]:

$$\frac{s_T}{s_L} = I_0 \int \Sigma \, \mathrm{d}n \,, \tag{2.55}$$

where the first term of the right hand side is the stretch factor I_0, and the second term is the integral along n, the normal coordinate to the local flame element, of the **flame surface density** (FSD) $\Sigma = |\nabla c|$. I_0 is usually close to unity unless differential diffusion plays a large role in the reactant balance in the preheat zone, like in hydrogen combustion [11–13], or intense turbulence is able to disrupt the structure of the reaction zone [15].

2.6 Large eddy simulations

For numerical simulations of turbulent premixed combustion, the complex interactions between turbulence and combustion outlined above suggest that the direct resolution of Equations 2.3 to 2.6 should be essential to capture the full extent of the physical phenomena at play. This is done in direct numerical simulations, which are unfortunately intractable for most practical flows. The reason lies in the vast separation of scales in turbulent and reactive flows. While chemical mechanisms require a resolution of the order of the laminar flame thickness $\delta_L \sim 0.1\,\mathrm{mm}$, typical combustion systems extend over $0.1\,\mathrm{m} - 10\,\mathrm{m}$. In addition, Equations 2.46, 2.47, 2.53 show that the separation between the integral lengthscale and the Kolmogorov lengthscale grows with the turbulent Reynolds number as [282]

$$\frac{l_t}{\eta} \propto Re_t^{3/4} \,. \tag{2.56}$$

These separations of scales require the computational domain to be both large and finely discretized, making DNS of high Reynolds number reacting flows very computationally expensive [310].

Large eddy simulations (LES) are a widely used computational alternative to DNS for practical combustion simulations. LES is able to capture fine, unsteady combustion phenomena in complex premixed combustion configurations [262, 263, 369]. In LES, the domain is discretized with a mesh size h which is much larger than the Kolmogorov scale η. The smallest turbulent eddies and chemistry at the scale of the flame front are therefore not

resolved and must be modeled. For inert turbulent flows, the concept of LES is justified by the universal nature of the smallest turbulent eddies in the turbulent cascade (Section 2.4), which makes them amenable to universal modeling. The large-scale features of the flow, which depend on the specific configuration, are fully resolved in a proper LES. For combustion phenomena, LES often exploits universal characterizations of the flame front through the flamelet concept and wrinkling-based models, as will be covered in Section 2.7.3.

The usual theoretical treatment of LES is to consider that it operates on spatially filtered quantities $\bar{\psi}$ [108, 319]. Filtering is represented by a spatial convolution with a low-pass kernel G:

$$\boxed{\bar{\psi}(\boldsymbol{x}) = (G * \psi)(x) = \int_{-\infty}^{+\infty} G(\boldsymbol{x} - \boldsymbol{x}')\psi(\boldsymbol{x}')\,\mathrm{d}\boldsymbol{x}'\,.} \tag{2.57}$$

The cutoff frequency of G is also called the filter size and denoted Δ. It is typically of the order of the mesh size h [319]. G is normalized:

$$\int_{-\infty}^{+\infty} G(x)\,\mathrm{d}x = 1\,, \tag{2.58}$$

and has a finite support. Explicitly filtered LES applies a closed-form filter G on the quantities discretized on the computational mesh, but comes at an increased computational cost. Instead, in implicitly filtered LES[2], variables discretized on the computational grid are already assumed to be intrinsically filtered without the intervention of an explicit filtering operator. Filtering is caused by a combination of the effects of the numerical scheme, subgrid-scale models, and discretization by the computational grid. This is the approach that will be used in this manuscript.

For compressible flows, Favre filtering [89] is used to avoid unnecessary unclosed terms in the filtered mass conservation equation:

$$\boxed{\widetilde{\psi} = \frac{\overline{\rho\psi}}{\bar{\rho}}\,.} \tag{2.59}$$

For both types of filtering, any quantity ψ can be decomposed into a filtered component and a high-frequency component:

$$\psi = \bar{\psi} + \psi' = \widetilde{\psi} + \psi''\,. \tag{2.60}$$

ψ' and ψ'' are called subfilter scale quantities. Some authors prefer to distinguish subfilter scale and subgrid scale quantities, the latter designating the residual obtained from the projection on the computational grid of size h [126]. This subtlety is mostly relevant for explicitly filtered LES [126]. Since $\Delta \sim h$, the term subgrid scale (SGS) will simply be used to denote the residual quantities ψ', ψ'' lost in the filtering operation.

[2]This should not be confused with the concept of implicit LES, which refers to LES where the numerical scheme is responsible for subgrid-scale closure instead of an explicit subgrid-scale model [319].

2.7 Large eddy simulations of turbulent premixed combustion

2.7.1 LES governing equations

Assuming that the LES filter commutes with differentiation, filtering the governing equations 2.3 to 2.6 leads to the LES equations:

$$\frac{\partial \bar{\rho}}{\partial t} + \nabla \cdot (\bar{\rho}\widetilde{\boldsymbol{u}}) = 0\,, \tag{2.61}$$

$$\frac{\partial \bar{\rho}\widetilde{\boldsymbol{u}}}{\partial t} + \nabla \cdot (\bar{\rho}\widetilde{\boldsymbol{u}} \otimes \widetilde{\boldsymbol{u}}) = \nabla \cdot (-\bar{P}\underline{\underline{\boldsymbol{I}}} + \underline{\underline{\bar{\tau}}} + \underline{\underline{\boldsymbol{\tau}}}^{\boldsymbol{t}})\,, \tag{2.62}$$

$$\frac{\partial \bar{\rho}\widetilde{e}_t}{\partial t} + \nabla \cdot (\bar{\rho}\widetilde{e}_t\widetilde{\boldsymbol{u}}) = \nabla \cdot (-\bar{P}\widetilde{\boldsymbol{u}} + \underline{\underline{\bar{\tau}}} \cdot \widetilde{\boldsymbol{u}} - \bar{\boldsymbol{q}} - \boldsymbol{q}^{\boldsymbol{t}}) + \overline{\dot{\omega}}_T\,, \tag{2.63}$$

$$\frac{\partial \bar{\rho}\widetilde{Y_k}}{\partial t} + \nabla \cdot (\bar{\rho}\widetilde{Y_k}\widetilde{\boldsymbol{u}}) = -\nabla \cdot (\bar{\boldsymbol{J}}_k + \boldsymbol{J}_k^{\boldsymbol{t}}) + \overline{\dot{\omega}}_k\,, \quad \text{for } k = 1, ..., N. \tag{2.64}$$

The filtered equations involve many new terms that must be resolved, closed, or neglected. In particular, the filtered energy equation 2.63 was simplified by computing the filtered energy from resolved quantities only: $\widetilde{e}_t = e_s(\widetilde{T}) + \widetilde{\boldsymbol{u}} \cdot \widetilde{\boldsymbol{u}}/2$. Rigorously speaking, this should result in a vast number of unclosed SGS fluxes coming from cross-correlations with convective terms [380]. *A priori* studies on filtered DNS have shown that most of these unclosed terms are typically much smaller than the leading order terms in the filtered energy equation, and that the non-negligible terms that remained to be modeled were the SGS heat flux (velocity-sensible enthalpy correlation) and the SGS turbulence diffusion (velocity-kinetic energy correlation) [254, 272]. Here, their effect is grouped in the term $\boldsymbol{q}^{\boldsymbol{t}}$ which is detailed below.

The new terms in Equations 2.61 to 2.64 are:

- $\underline{\underline{\bar{\tau}}}$, the filtered viscous stress tensor:

$$\underline{\underline{\bar{\tau}}} = \overline{2\mu\left[\underline{\underline{\boldsymbol{S}}} - \frac{1}{3}(\nabla \cdot \boldsymbol{u})\underline{\underline{\boldsymbol{I}}}\right]} \tag{2.65}$$

$$\approx 2\mu(\widetilde{T})\left[\underline{\underline{\widetilde{\boldsymbol{S}}}} - \frac{1}{3}(\nabla \cdot \widetilde{\boldsymbol{u}})\underline{\underline{\boldsymbol{I}}}\right]\,. \tag{2.66}$$

- $\underline{\underline{\boldsymbol{\tau}}}^{\boldsymbol{t}}$, the SGS Reynolds stress tensor:

$$\underline{\underline{\boldsymbol{\tau}}}^{\boldsymbol{t}} = -\bar{\rho}(\widetilde{\boldsymbol{u} \otimes \boldsymbol{u}} - \widetilde{\boldsymbol{u}} \otimes \widetilde{\boldsymbol{u}})\,. \tag{2.67}$$

  It is modeled with the Boussinesq hypothesis, according to which energy transfers from the resolved to the subgrid scales are analogous to molecular diffusion. This postulates the existence of a turbulent viscosity μ_t such that:

$$\underline{\underline{\boldsymbol{\tau}}}^{\boldsymbol{t}} = 2\bar{\rho}\mu_t\left[\underline{\underline{\widetilde{\boldsymbol{S}}}} - \frac{1}{3}(\nabla \cdot \widetilde{\boldsymbol{u}})\underline{\underline{\boldsymbol{I}}}\right]\,. \tag{2.68}$$

  Practical models for μ_t are presented in Section 2.8.

- $\bar{\boldsymbol{q}}$, the filtered heat flux:

$$\bar{\boldsymbol{q}} = -\overline{\lambda \nabla T} + \sum_{k=1}^{N} \overline{\boldsymbol{J}_k h_{s,k}} \tag{2.69}$$

$$\approx -\bar{\lambda} \nabla \widetilde{T} + \sum_{k=1}^{N} \bar{\boldsymbol{J}}_k h_{s,k}(\widetilde{T}) \,. \tag{2.70}$$

- $\boldsymbol{q^t}$, an SGS term modeling the unclosed terms of the filtered energy equation as additional turbulent contributions to the filtered heat flux:

$$\boldsymbol{q^t} = -\lambda_t \nabla \widetilde{T} + \sum_{k=1}^{N} \boldsymbol{J}_k^{\boldsymbol{t}} h_{s,k}(\widetilde{T}) \,, \tag{2.71}$$

  where λ_t is a turbulent conductivity defined as:

$$\lambda_t = \frac{\mu_t c_p}{Pr_t} \,, \tag{2.72}$$

  with a turbulent Prandtl number Pr_t set to 0.6.

- $\bar{\boldsymbol{J}}_k$, the filtered species diffusive flux:

$$\bar{\boldsymbol{J}}_k = \overline{\rho Y_k \left[-\frac{D_k}{X_k} \nabla X_k + \sum_{j=1}^{N} D_j \frac{W_j}{W} \nabla X_j \right]} \tag{2.73}$$

$$\approx \bar{\rho} \left[-\bar{D}_k \frac{W_k}{W} \nabla \widetilde{X}_k + \sum_{j=1}^{N} \bar{D}_j \frac{W_j}{W} \nabla \widetilde{X}_j \right] \,, \tag{2.74}$$

  with $\bar{D}_k \approx \bar{\mu}/(\bar{\rho} Sc_k)$.

- $\boldsymbol{J_k^t}$, the SGS species convective flux:

$$\boldsymbol{J}_k^{\boldsymbol{t}} = \bar{\rho}(\widetilde{\boldsymbol{u} Y_k} - \widetilde{\boldsymbol{u}} \widetilde{Y_k}) \tag{2.75}$$

$$\approx \bar{\rho} \left[-\bar{D}_k^t \frac{W_k}{W} \nabla \widetilde{X}_k + \sum_{j=1}^{N} \bar{D}_j^t \frac{W_j}{W} \nabla \widetilde{X}_j \right] \,, \tag{2.76}$$

  which is modeled through the action of an additional turbulent diffusivity $\bar{D}_k^t = \mu_t/(\bar{\rho} Sc_k^t)$ involving a turbulent Schmidt number Sc_k^t set to 0.6. Subgrid-scale countergradient transport [372] is therefore not modeled.

- $\overline{\dot{\omega}}_k$, the filtered production rate which is the main focus point of turbulent combustion models. Given $\overline{\dot{\omega}}_k$, the filtered heat release rate $\overline{\dot{\omega}}_T$ is derived from Equation 2.16.

In addition, the perfect gas equation of state is considered to hold for the filtered quantities:

$$\bar{P} = \bar{\rho} \frac{R}{W} \widetilde{T} \,. \tag{2.77}$$

2.7.2 Modeling frameworks

The main challenge for LES of turbulent premixed flames is that the mesh resolution is too coarse to properly resolve the flame structure. To properly model turbulent combustion phenomena in spite of this constraint, statistical and geometrical modeling approaches are two viable alternatives.

Probability density function (PDF) methods rely on a statistical description of turbulent and combustion mechanisms via the composition or velocity-composition joint PDF (JPDF) [280]. A transport equation for this JPDF is solved using Monte-Carlo methods. These methods have the benefit of representing convective transport and detailed chemical kinetics without closure, but modeling is required for small-scale mixing [279]. PDF methods have greatly evolved since the theoretical foundations laid by Pope in the 1980s [280], and significant work is dedicated today to improving mixing models, computational efficiency, and numerical accuracy [135]. These methods involve very specific numerical tools, notably Lagrangian Monte-Carlo methods instead of grid-based numerical schemes, which are far remote from the ones used in this manuscript.

Alternatively, several modeling frameworks focus on a geometrical description of the flame front and its surface:

- Level set approaches introduce a level set function G which is used to locate the instantaneous flame surface at $G = G_0$ under flamelet assumptions [237, 273]. Values of G away from the flame front are usually set to the distance to the flame front. The rationale behind this approach is to avoid the numerical difficulties involved in dealing directly with a discontinuous (at the scale of the LES mesh) flame surface by implicitly parametrizing the flame surface with the continuous G function. The G-equation describes the evolution of G:

$$\frac{\partial G}{\partial t} + \boldsymbol{u} \cdot \nabla G = s_L |\nabla G| \,. \tag{2.78}$$

  Adapting the G-equation to LES requires the filter to be applied along the instantaneous flame surface only [273]. The filtered G-equation needs models for the subfilter propagation of the flame front, interactions between flame curvature and molecular transport, and the flame front-conditioned filtered velocity. An additional transport equation for the progress variable may be included to account for resolved transport in the preheat zone [237].

- Flame surface density methods also rely on flamelet assumptions and work from the filtered progress variable balance equation [33]:

$$\frac{\partial \overline{\rho}\widetilde{c}}{\partial t} + \nabla \cdot (\overline{\rho \boldsymbol{u} c}) = \nabla \cdot (\overline{\rho D_c \nabla c}) + \overline{\dot{\omega}}_c = \langle \rho s_d \rangle_s \overline{\Sigma} \approx \rho_u \langle s_c \rangle_s \overline{\Sigma} \,, \tag{2.79}$$

  where D_c is the progress variable diffusivity, $\dot{\omega}_c$ the progress variable source term, s_d the displacement speed, $\overline{\Sigma} = \overline{|\nabla c|}$ is the generalized total flame surface density, and $\langle \psi \rangle_s = \overline{\psi \Sigma} / \overline{\Sigma}$ denotes averaging along the flame surface. This approach fully decouples the flame structure, contained in the $\langle s_c \rangle_s$ term, from the wrinkling of the

flame sheet, determined by $\overline{\Sigma}$. $\overline{\Sigma}$ can be modeled algebraically by a wrinkling model or via a transport equation [278, 363]. For instance, Boger et al. [33] propose an algebraic expression for $\overline{\Sigma}$ in the limit of a thin flame front relative to the filter size:

$$\overline{\Sigma} = 4\sqrt{\frac{6}{\pi}}\Xi\frac{\widetilde{c}(1-\widetilde{c})}{\Delta}\,, \tag{2.80}$$

where the SGS wrinkling factor Ξ remains to be modeled.

- The Filtered Tabulated Chemistry for LES (F-TACLES) formalism [98] relies on a progress variable parametrization of the flame, in which thermochemical quantities are tabulated as a function of the progress variable. To close the filtered LES equations, the filtered progress variable source term, filtered laminar diffusion terms, and various SGS fluxes are supposed to depend only on $\widetilde{c}$ and the filter size Δ and are thus tabulated from filtered 1D laminar flame solutions. A wrinkling model is required to handle SGS wrinkling of the flame front that does not appear in the 1D laminar flames. Recent developments also account for the effect of SGS wrinkling on the flame structure by filtering 2D sinusoidal flames [225]. In the limit of an infinitely thin flame front, F-TACLES recovers the Boger model (Equation 2.80).

- The **thickened flame model for LES** (TFLES) [40, 67] artificially thickens the flame front by both multiplying the diffusivities and dividing the reaction rates by a **thickening factor** F. The scaling relations in Equation 2.34 show that this does not affect the laminar flame speed but increases the laminar thermal flame thickness by F. This enables the flame front to be fully resolved on the LES mesh, and for detailed chemical kinetics to be naturally included. The flame structure may however be affected, especially flame-turbulence interactions in the preheat and reaction zones. Additionally, the thickening operation reduces the surface of the resolved flame front. This is compensated by an **efficiency function** E which accounts for unresolved wrinkling of the thickened flame front. In practice, this consists in applying the following transformations to Equations 2.61 to 2.64:

$$\bar{D}_k \to EF\bar{D}_k\,, \tag{2.81}$$
$$\bar{\lambda} \to EF\bar{\lambda}\,, \tag{2.82}$$
$$\overline{\dot{\omega}}_k \to \frac{E}{F}\dot{\omega}_k(\tilde{\psi})\,, \tag{2.83}$$
$$\overline{\dot{\omega}}_T \to \frac{E}{F}\dot{\omega}_T(\tilde{\psi})\,. \tag{2.84}$$

This results in the following turbulent flame speed s_T and thickened flame thickness $\overset{\vee}{\delta}_L$:

$$\boxed{s_T = Es_L\,,} \tag{2.85}$$
$$\boxed{\overset{\vee}{\delta}_L = F\delta_L\,.} \tag{2.86}$$

This manuscript will adopt the TFLES formalism which has a long track record as a predictive and computationally efficient method for LES of practical premixed combustion

configurations [150, 239, 287, 369, 378]. The main term to be modeled is the efficiency function E which is directly equated to the SGS **wrinkling factor** at the scale Δ:

$$\boxed{E = \Xi_\Delta \,.} \tag{2.87}$$

In the thickening formalism, the scale Δ will be defined in the following section devoted to wrinkling models. SGS wrinkling models can also be applied to the flame surface density and F-TACLES approaches if a transport equation for Σ is not used.

2.7.3 Subgrid-scale wrinkling factor

SGS wrinkling models originate from a flame surface density approach to turbulent combustion modeling. Assuming that reactions take place in an infinitely thin sheet located at $c = c^*$, the surface area of the reaction sheet is given by the co-area formula [90, 242]:

$$A(c^*) = \int_V |\nabla c| \delta(c - c^*) \, \mathrm{d}\mathcal{V} \,, \tag{2.88}$$

where δ is the Dirac delta function. This formula justifies the interpretation of $|\nabla c|$ as the flame surface density (FSD), measuring the area of the flame surface per unit volume. Intuitively, $|\nabla c|$ measures the density of c isosurfaces. Under flamelet assumptions, it is reasonable to presume that iso-surfaces of c remain parallel and that the flame surface area is well approximated by averaging out Equation 2.88 for all c values:

$$A = \int_0^1 A(c) \, \mathrm{d}c = \int_V |\nabla c| \, \mathrm{d}\mathcal{V} \,. \tag{2.89}$$

In the LES formalism, this area can be recovered thanks to the normalization condition of the filter (Equation 2.58):

$$A = \int_V \overline{|\nabla c|} \, \mathrm{d}\mathcal{V} \,. \tag{2.90}$$

Equation 2.90 involves the total FSD $\overline{|\nabla c|}$ which can be integrated to recover the total flame surface A. To close $\overline{|\nabla c|}$, it is often connected to the resolved FSD $|\nabla \bar{c}|$ through the SGS wrinkling factor Ξ_Δ:

$$\boxed{\overline{\Sigma} = \Xi_\Delta |\nabla \bar{c}| \,.} \tag{2.91}$$

Ξ_Δ measures the ratio of the total FSD to the resolved FSD. It is equal to one when flame wrinkling is fully resolved, like in the case of a laminar flame. Ξ_Δ depends on the **combustion filter size** Δ related to the thickness of the LES flame. In the TFLES formalism, Ξ_Δ must account for the artificial thickening of the flame front in addition to LES filtering. The TFLES combustion filter size is thus defined as:

$$\boxed{\Delta = F \delta_L \,.} \tag{2.92}$$

Figure 2.7 illustrates the difference between a true, thin flame front and a thickened flame front. Thickening (as well as filtering) induces a loss of flame surface that is compensated by the SGS wrinkling factor.

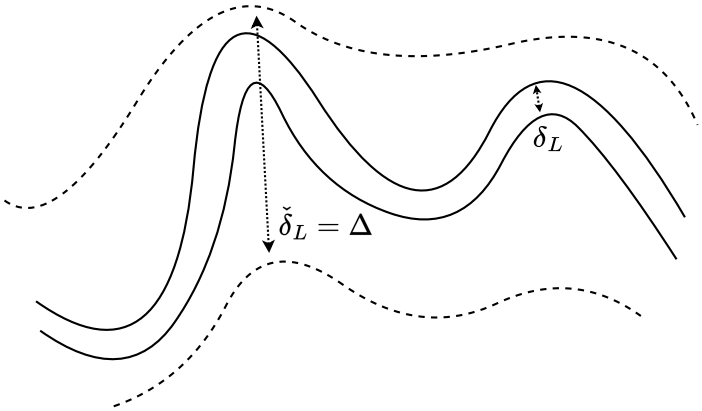


Figure 2.7: True (solid) and thickened (dashed) flame fronts.

2.7.4 Subgrid-scale wrinkling models

Algebraic models for Ξ_Δ have seen extensive developments over the years and have been comparatively reviewed in the literature [44, 214]. They are divided into two families:

- Models based on correlations of the turbulent flame speed [67, 302, 392]. These models leverage Equation 2.55 to express Ξ_Δ as a function of turbulence parameters such as u'/s_L, l_t/δ_L. For instance, Colin et al. [67] propose the expression:

$$\Xi_\Delta = 1 + \alpha\Gamma_\Delta \frac{u'}{s_L}\,, \tag{2.93}$$

  where Γ_Δ accounts for the net straining effect of all vortices smaller than Δ, and α is a model constant computed as:

$$\alpha = \beta_{Colin} \frac{2\ln 2}{3c_{ms}(Re_t^{1/2} - 1)}\,. \tag{2.94}$$

  β_{Colin} is a parameter prescribed by the user, usually set to 0.3.

- Models based on a **fractal** description of the flame front [45, 46, 106, 117, 118, 134, 160, 167, 385]. These will be detailed in the following.

Building from the seminal work of Gouldin et al. [117, 118], fractal models assume that, in a range of physical scales bounded by an inner cutoff η and an outer cutoff L, the flame front is a fractal surface of dimension D such that $2 \leq D \leq 3$. The flame surface area A depends on the measurement scale ϵ as:

$$A(\epsilon) \propto \epsilon^{2-D}\,. \tag{2.95}$$

As shown in Figure 2.8, the fractal nature of the flame is characterized by a constant slope between the cutoff scales in the log-log plot of A versus ϵ. The wrinkling factor Ξ is defined

as the relative amount of missing flame surface at any given scale ϵ:

$$\Xi_\epsilon = \frac{A(\eta)}{A(\epsilon)} \geq 1\,. \tag{2.96}$$

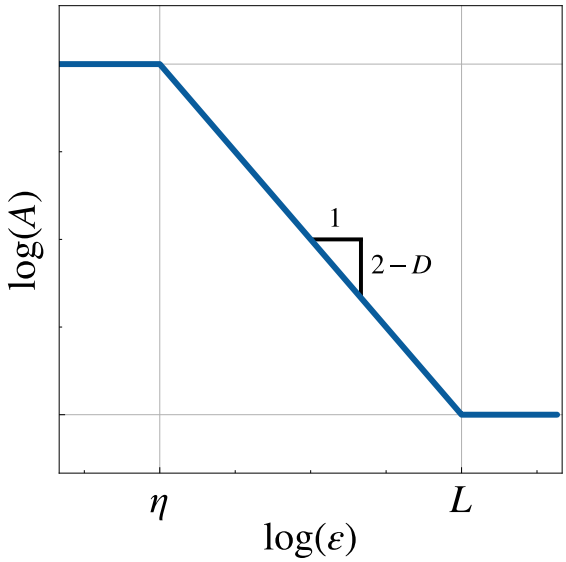


Figure 2.8: Evolution of the area A with respect to the measurement scale ϵ for a fractal surface with cutoff lengths $[\eta, L]$.

Skiba et al. [344] have highlighted the fractal nature of an experimental turbulent flame using a stepping-caliper measurement technique shown in Figure 2.9. The area of a progress variable iso-surface in an experimental flame (top left) is measured using calipers ("rulers") of increasing length (bottom). As the caliper becomes larger, the flame is discretized at a coarser resolution and the surface decreases. Scale invariance denoting a fractal behavior is evidenced by the power-law relationship between the surface area and the caliper size (top right), and the slope of the linear trend in log-space provides the fractal dimension. For a highly resolved flame front (experimental or DNS), it is therefore possible to estimate the fractal dimension and the cutoff scales.

In LES, the exact values of the cutoff scales and the fractal dimension are unknown and need to be modeled. Theoretical scaling arguments based on Damköhler's small- and large-scale limits [267] indicate that D ranges from 7/3 in flamelets to 8/3 in high Karlovitz flames [134]. Experimental measurements lean towards the lower end of this range, with recent results on highly turbulent flames reporting $2.1 \leq D \leq 2.3$ [344]. L corresponds to the size of the largest unresolved wrinkles, which is roughly the turbulence integral lengthscale l_t in RANS [117] and the combustion filter size Δ in LES [46, 166]. This justifies the expression of Δ in TFLES (Equation 2.92), as the thickened flame thickness is also the characteristic size of the largest unresolved wrinkles. η is the size of the smallest wrinkles which is larger or equal to the laminar flame thickness δ_L [46] and scales with the inverse of the Karlovitz number [125, 343].

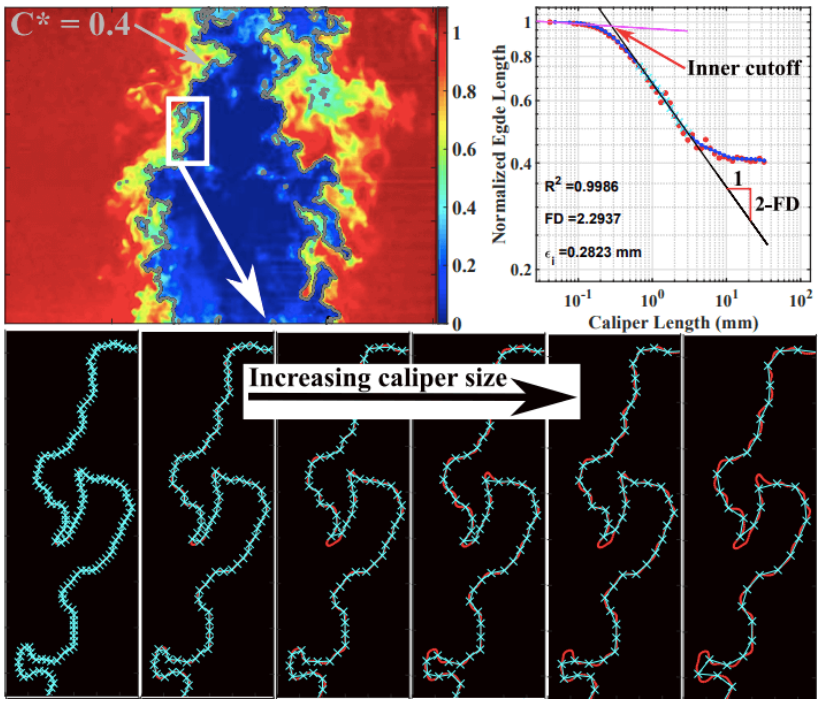


Figure 2.9: Illustration of the fractal nature of an experimental turbulent flame. Reprinted from Skiba et al. [344] with permission from Elsevier, license number 5365931215646.

2.7.5 Charlette model and its variants

One of the most widely-used fractal wrinkling models is the **Charlette model** [45]. In this model, the inner cutoff scale η is chosen as the inverse mean curvature of the flame $|\langle \nabla \cdot \boldsymbol{n} \rangle_s|$ with $\boldsymbol{n}$ the normal vector to the flame front. It is modeled by assuming an equilibrium of the production and destruction of SGS flame surface density, and lower bounded by the laminar flame thickness. The resulting expression for the SGS wrinkling factor is [385]:

$$\Xi_\Delta = \left(1 + \min\left[\frac{\Delta}{\delta_L} - 1, \Gamma_\Delta \frac{u'_\Delta}{s_L}\right]\right)^\beta . \tag{2.97}$$

Γ_Δ is a vortex efficiency function that serves the same purpose as in the Colin model of Equation 2.93. While the Colin model introduced a multiplicative model parameter α, the Charlette model uses a power-law exponent β which is linked to the fractal dimension by $\beta = D - 2$ (Equation 2.95). A constant value $\beta = 0.5$ ($D = 2.5$) is proposed in the original paper and leads to a **static** version of the Charlette model. When u'_Δ is sufficiently large, Equation 2.97 takes on a **saturated** form:

$$\boxed{\Xi_\Delta = \left(\frac{\Delta}{\delta_L}\right)^\beta}, \tag{2.98}$$

where the wrinkling does not depend on the turbulence intensity.

The power-law parameter β can also be determined by a dynamic procedure [46] where it

becomes a spatially and temporally evolving quantity. This avoids the delicate and arbitrary choice of one single value for β, which is often only justified *post hoc* by comparison to DNS or experimental data. It is also supported by empirical evidence highlighting significant spatial and temporal variations of the fractal dimension in turbulent flames [160, 344].

The dynamic procedure introduces a filtering operation $\hat{\bar{Q}}$ at an effective test-filter size[3] $\hat{\Delta} = \gamma\Delta > \Delta$ and an averaging operation $\langle Q \rangle$ over a size $\Delta_m > \hat{\Delta}$. The model assumes that the fractal behavior extends up to the test-filter scale $\hat{\Delta}$, and compares the flame surface at scales[4] Δ and $\hat{\Delta}$ to derive β, as illustrated in Figure 2.10.

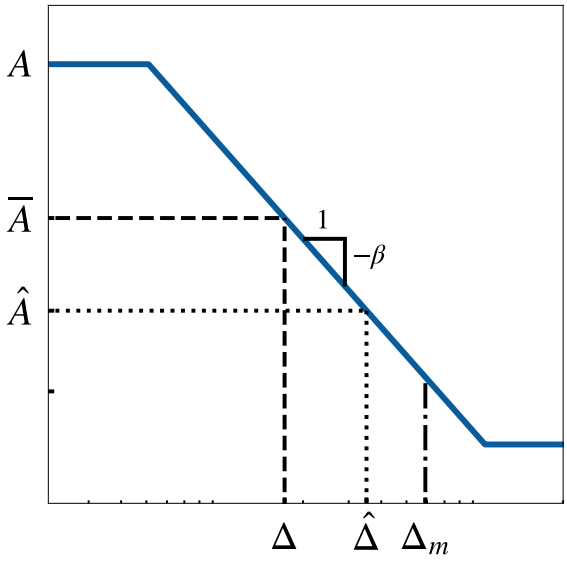


Figure 2.10: Position of the dynamic filtering scales in the fractal cascade. The flame surfaces $\overline{A}$, $\hat{A}$ respectively resolved at the scales Δ, $\hat{\Delta}$ are compared to deduce the total flame surface A.

By equating two expressions of the averaged test-filtered total FSD:

$$\widehat{\langle \Xi_\Delta |\nabla \bar{c}| \rangle} = \langle \Xi_{\hat{\Delta}} |\nabla \hat{\bar{c}}| \rangle \,, \tag{2.99}$$

and assuming that β is uniform over the averaging volume, a closed-form formula for β can be found. The high levels of turbulence seen in practical turbulent configurations mean that Equation 2.97 often takes its saturated form [371] and in this case, the dynamic expression for β is:

$$\boxed{\beta = \frac{\ln\left(\langle \widehat{|\nabla \bar{c}|} \rangle / \langle |\nabla \hat{\bar{c}}| \rangle\right)}{\ln \gamma}} \,. \tag{2.100}$$

[3] Ideal filters like a sharp spectral filter are projection operators, meaning that $\bar{\bar{\psi}} = \bar{\psi}$ [282]. In this case, the composition of two filters of size Δ and $\Delta' \geq \Delta$ is equivalent to filtering at a size Δ'. This is not the case for non-ideal, practical LES filters. The true LES filter is unknown and is often approximated by a Gaussian filter [188, 313]. The composition of two Gaussian filters of size Δ and $\Delta' = c_1\Delta \geq \Delta$ is equivalent to a single Gaussian filter of effective size $\sqrt{1 + c_1^2} > \Delta'$.

[4] Here, the scales are analogous to the caliper lengths in the stepping-caliper measurement of Figure 2.9.

Equation 2.100 involves the ratio $\langle|\widehat{\nabla \bar{c}}|\rangle / \langle|\nabla \hat{\bar{c}}|\rangle$ which can be interpreted as the wrinkling between the scales Δ and $\hat{\Delta}$. After introducing the notations:

$$\Sigma_1 = |\widehat{\nabla \bar{c}}|\,, \tag{2.101}$$

$$\Sigma_2 = |\nabla \hat{\bar{c}}|\,, \tag{2.102}$$

Equation 2.100 yields:

$$\beta = \frac{\ln\left(\langle\Sigma_1\rangle / \langle\Sigma_2\rangle\right)}{\ln \gamma}\,. \tag{2.103}$$

Due to the behavior of the ratio $\langle\Sigma_1\rangle / \langle\Sigma_2\rangle$, this formulation leads to numerical issues when flame fronts:

1. are near domain boundaries,
2. interact with other flame fronts.

These issues are illustrated in the PhD thesis of P. Quillatre [290], where the uncorrected dynamic Charlette model was used in simulations of explosions in semi-confined domains and led to spuriously high, unphysical values of β. The dynamic model was later corrected by S. Mouriaux to handle both problematic cases. These modifications are summarized below, and full details are contained in her paper and PhD thesis [238, 239].

Mouriaux et al. [239] illustrate the presence of numerical issues near domain boundaries in the simple case of a laminar 1D flame front. In this configuration, there is no subgrid-scale wrinkling and β should be equal to zero everywhere. Since $\bar{c}$ is strictly increasing, Σ_1 and Σ_2 simplify to $\widehat{\nabla \bar{c}}$ and $\nabla \hat{\bar{c}}$. The test-filtering and gradient operations must therefore commute to ensure $\beta = 0$. This is the case everywhere except near the boundary, where the test-filtering operator stencil is truncated. A simple solution is to swap the order of the gradient and test-filtering operations in Σ_2:

$$\Sigma_{2,new} = |\widehat{\nabla \bar{c}}|\,. \tag{2.104}$$

This enforces the correct behavior for a laminar flame and does not affect values far from the boundary.

The behavior of Σ_1 and Σ_2 in the presence of flame front interactions is also problematic. When two flame fronts are separated by a distance d smaller than the test-filter size $\hat{\Delta}$, the test-filtering operation blends the two filtered fronts into a test-filtered progress variable profile $\hat{\bar{c}}$ that is flat in the interaction zone, leading to small values of Σ_2. In contrast, test-filtering the filtered gradients $|\nabla \bar{c}|$ that peak inside the filtered fronts leads to uniformly large values of Σ_1. The resulting β field would therefore be high even in the absence of unresolved wrinkling.

To correct this behavior, a flame interaction sensor ζ is introduced. It is built by comparing the directions of the normal vectors to the filtered and test-filtered flame fronts, noted $\boldsymbol{n}$ and $\boldsymbol{N}$ respectively. The normal vectors are aligned when there are no flame front interactions, and are not aligned when flame front interact at the test-filter level. ζ is hence defined as:

$$\zeta = \begin{cases} 1, & \text{if } \boldsymbol{n} \cdot \boldsymbol{N} < 0.9, \\ 0, & \text{otherwise.} \end{cases} \tag{2.105}$$

ζ is then test-filtered to match the characteristic scale of flame-front interactions $\hat{\Delta}$. In regions where $\hat{\zeta} = 1$, the expression

$$\Sigma_3 = \widehat{\boldsymbol{n} \cdot \boldsymbol{N} |\nabla \bar{c}|} \tag{2.106}$$

is used instead of Σ_2 to account for the interactions.

In the end, the final expression for Σ_2 that accounts for domain boundary and flame front interaction corrections is:

$$\boxed{\Sigma_{2,corr} = (1 - \hat{\zeta})\Sigma_{2,new} + \hat{\zeta}\Sigma_3 \,.} \tag{2.107}$$

The dynamic Charlette model in its various forms has been applied to LES of jet flames [326, 377, 385], ignition kernels [239, 386], stratified non-swirling burners [226, 285], the PRECCINSTA swirled burner [371, 378], explosions in semi-confined domains [376], and light-around in an annular combustor [287]. It has also seen numerous incremental improvements over the years [239, 285, 385] and stands today as a state-of-the-art model for the SGS wrinkling factor. The dynamic saturated Charlette model with Mouriaux corrections is abbreviated as the CDSM model, and will be used in Chapters 5 and 8 as a baseline to benchmark DL models.

2.8 Numerical simulations with the AVBP solver

In this manuscript, DNS and LES simulations will be performed using the AVBP solver [327, 328]. AVBP is an explicit massively parallel code that solves the compressible, multispecies, reacting Navier-Stokes equations (both Equations 2.3 to 2.6 and Equations 2.61 to 2.64) on unstructured grids with a cell-vertex formulation. Abundant details on the numerical methods used in AVBP are contained in the PhD thesis of N. Lamarque [180]. The numerical scheme that will be used throughout this work is the Two-step Taylor Galerkin scheme (TTGC) [68]. It is a finite element centered scheme with explicit two-step integration in time. It is third-order accurate in space and time, and has excellent dissipative and dispersive properties that are valuable in unsteady LES and DNS. Boundary conditions are treated with the Navier-Stokes Characteristic Boundary Conditions (NSCBC) method [277].

Second- and fourth-order artificial viscosity is introduced to damp spurious waves created by dispersion errors of the numerical scheme. These oscillations may not be smoothed out by physical diffusion, potentially leading to unphysical values for all thermochemical quantities. In the DNS computations of this manuscript, only fourth-order artificial viscosity is added to the density, species, and energy residuals to prevent node-to-node oscillations. In the LES computations, both second- and fourth-order artificial viscosities are used, and second-order artificial viscosity is only applied when stiff numerical gradients are detected.

Turbulent viscosity models In LES, the role of the turbulent viscosity μ_t (Equation 2.68) is to reproduce the energy transfer from the resolved scales to the SGS scales. Two models for μ_t will be used in the LES simulations carried out in this work:

- The WALE model [248] improves on the Smagorinsky model [194] by accounting for the effect of the rotation rate of resolved turbulent eddies, and recovering the correct

near-wall scaling:

$$\mu_t = \bar{\rho}(C_w\Delta)^2 \frac{(\underline{\underline{\boldsymbol{\mathcal{S}}}}^{\boldsymbol{d}} : \underline{\underline{\boldsymbol{\mathcal{S}}}}^{\boldsymbol{d}})^{3/2}}{(\underline{\underline{\bar{\boldsymbol{S}}}} : \underline{\underline{\bar{\boldsymbol{S}}}})^{5/2} + (\underline{\underline{\boldsymbol{\mathcal{S}}}}^{\boldsymbol{d}} : \underline{\underline{\boldsymbol{\mathcal{S}}}}^{\boldsymbol{d}})^{5/4}}\,, \tag{2.108}$$

where $C_w = 0.49$ is a model constant, $\underline{\underline{\bar{\boldsymbol{S}}}}$ is the strain rate tensor of the resolved velocities, and

$$\underline{\underline{\boldsymbol{\mathcal{S}}}}^{\boldsymbol{d}} = \frac{1}{2}\left[\nabla\bar{\boldsymbol{u}}\nabla\bar{\boldsymbol{u}} + (\nabla\bar{\boldsymbol{u}}\nabla\bar{\boldsymbol{u}})^T\right] - \frac{1}{3}(\nabla\bar{\boldsymbol{u}} : \nabla\bar{\boldsymbol{u}})\underline{\underline{\boldsymbol{I}}} \tag{2.109}$$

is the traceless symmetric part of the square of the velocity gradient tensor $\nabla\bar{\boldsymbol{u}}$.

- The Sigma model [249] builds upon the WALE model, and additionally suppresses the turbulent viscosity for solid rotation and when resolved scales are in pure axisymmetric or isotropic expansion:

$$\mu_t = \bar{\rho}(C_\sigma\Delta)^2 \frac{\sigma_3(\sigma_1-\sigma_2)(\sigma_2-\sigma_3)}{\sigma_1^2}\,, \tag{2.110}$$

where $C_\sigma = 1.35$ is a model constant, and $\sigma_1 \geq \sigma_2 \geq \sigma_3$ are the singular values of the velocity gradient tensor.

Dynamic viscosity The dynamic viscosity μ is assumed to be close to that of air, and is modeled via a power-law dependency on the temperature T:

$$\mu = \mu_{ref}\left(\frac{T}{T_{ref}}\right)^b\,, \tag{2.111}$$

where $\mu_{ref} = 1.8\times10^5\,\mathrm{kg\,m^{-1}\,s^{-1}}$, $T_{ref} = 300\,\mathrm{K}$, and $b = 0.68$.

TFLES implementation The TFLES thickening operations (Equations 2.81 to 2.84) are only applied in the flame front thanks to a flame sensor $\mathcal{S}$ [187]. This allows mixing to be correctly computed outside of the flame zone, where the thickening operation is not needed. The thickening factor simply writes:

$$F = 1 + (F_{max} - 1)\mathcal{S}\,, \tag{2.112}$$

where F_{max} is the user-defined value set in the flame zone. The flame sensor is computed as:

$$\mathcal{S} = \tanh\left(\beta'\frac{\Omega}{\Omega_0}\right)\,, \tag{2.113}$$

with $\beta' = 50$. Ω is based on the fuel oxidation reaction:

$$\Omega = Y_F^{\nu_F}Y_O^{\nu_O}\exp\left(-\frac{E_a}{RT}\right)\,, \tag{2.114}$$

and Ω_0 is its maximum value in a 1D laminar flame.

Moreover, when thickening is applied in the flame region, the structure of the flame is considered to be fully resolved, and SGS energy and species diffusive fluxes are discarded.

These terms are therefore transformed like so:

$$\boldsymbol{q}^t \to (1 - \mathcal{S})\boldsymbol{q}^t \,, \tag{2.115}$$

$$\boldsymbol{J}_k^t \to (1 - \mathcal{S})\boldsymbol{J}_k^t \,. \tag{2.116}$$

To improve the computational efficiency of the CDSM model, the test-filtering and average filtering operations are implemented via a second-order truncation of the Taylor series expansion of the application of a Gaussian filter [236]. This allows for a simple expression of the filtered quantity using second- and fourth-order derivatives which can easily be computed on an unstructured mesh. For similar efficiency reasons, the β parameter is updated at a user-defined frequency instead of every iteration [377], since the flame time is typically orders of magnitude larger than the convective timestep imposed by the Courant–Friedrichs–Lewy (CFL) number.

Chapter 3

Deep learning

> The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin.
>
> Richard Sutton, *The Bitter Lesson*[1]

Contents

[1] http://www.incompleteideas.net/IncIdeas/BitterLesson.html

Deep learning (DL) is a branch of machine learning (ML) where deep neural networks are used as high-capacity predictive models. This chapter limits its coverage to the notions that are necessary to understand the DL developments of this thesis, gradually narrowing its focus to the specific tools that are used in this work. First, a brief introduction to machine learning is offered. The chapter then turns towards deep neural networks, explaining their mathematical foundations and how they are trained in practice. A presentation of convolutional neural networks and why they are useful follows. Finally, the U-Net architecture is detailed, as it is the backbone of the DL models trained in the rest of this work. To delve deeper into the fascinating world of deep learning, the reader is referred to the classic books of Bishop [29], Hastie et al. [133], Chollet [58] and Goodfellow et al. [115].

3.1 Introduction to machine learning

3.1.1 What is machine learning?

In machine learning, a predictive model is trained to perform a learning task from exposure to training data. The model takes **features** of the data as input, and forms a **prediction** on a variable of interest. It must be able to **generalize**, that is, retain its predictive ability on data that was not seen during training. Clearly, as the amount of training data grows, the empirical training data distribution $p(x_{train})$ will become more representative of the true data distribution $p(x_{true})$. Machine learning therefore benefits from an abundance of data as well as sufficiently large computing power to process it. ML models contain very few or no handcrafted rules, and are especially useful for problems where an automated solution is needed and rule-based algorithms fail. In particular, machine learning excels in predictive tasks involving high-dimensional data.

3.1.2 Machine learning paradigms

Depending on the nature of the training data available to the model, machine learning approaches fall under three broad categories:

- **Supervised learning**: for each **training sample** x, there is a **label** y that corresponds to the output of the learning task. Labels guide the learning process by providing clear **ground truth** targets for the model.

- **Unsupervised learning**: no labels are available. When some supervision is often inferred from the structure of the data, the term **self-supervised learning** is used. Self-supervised learning can be understood under the framework of energy-based models [186], where the model learns to map a pair of inputs x_1 to x_2 to an energy function. The energy function is small if x_1 and x_2 are compatible with regards to the learning task, and large if they are not. For example, in image recognition, two distorted versions of the same image may be encouraged to be similar as long as the distortion preserves the semantic nature of the content. Difficulties emerge when attempting to construct incompatible pairs while avoiding a collapse of the model to a trivial solution. Self-supervised learning is an active research domain where contrastive and non-contrastive

approaches compete [41, 53, 54, 120, 136, 365, 407]. There is no hard boundary between supervised and unsupervised learning, and it is often useful to combine both frameworks, as is done for example in knowledge distillation [139, 361].

- **Reinforcement learning** [355]: an agent observes the state of an environment and takes actions that affect these states. It is trained to maximize a reward function that can correspond to a predefined goal, or learn to discover intrinsic rewards [358]. Reinforcement learning agents have produced outstanding results in robotic manipulation [10, 121], video games [230, 374], and perfect information strategy games like Go, Shogi, and chess [338, 339].

3.1.3 Learning tasks

Learning tasks are too diverse to enumerate exhaustively. Three common examples are:

- **Regression**, where the goal is to predict continuous numerical values. These target values can be scalars (for instance predicting house prices based on their specifications) or vectors (finding the coordinates of the bounding box of objects in an image [240]).
- **Classification**, which aims to predict discrete, predetermined classes. Image classification is often used to benchmark computer vision models on datasets such as ImageNet [151] or CIFAR-10 [175]. The model predicts a single class label for each image. For a finer understanding of the content of an image, semantic segmentation is the task of classifying each pixel in an image [204]. The output is then a map of class labels with the same dimensions as the input image.
- **Generative modeling**, which learns to sample a data distribution. Among generative models, likelihood-based models directly learn the distribution's PDF by maximum likelihood optimization. This family includes variational autoencoders (VAEs) [165, 309], normalizing flows [308], and auto-regressive models [367]. Generative adversarial networks (GANs) [116] replicate the sampling process of the distribution without explicitly learning its PDF by adversarially learning to transform samples drawn from a noise distribution. Diffusion models [347] slowly perturb training data with increasing random noise, and learn to reverse this process by estimating the score (gradient of the log-PDF) of the perturbed distribution with a neural network. Generative models must find a compromise between high sample quality, sample diversity, and sampling efficiency [398].

3.1.4 Bias-variance tradeoff

Consider a supervised regression task with inputs x and labels $y(x)$. For any training dataset $\mathcal{D}$, an ML model can be trained. Its prediction on an unseen test sample x is noted $\hat{y}(x;\mathcal{D})$. The average prediction error on x is the expected value over all training datasets $\mathcal{D}$ of the least squares error of the model:

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\left[(\hat{y}(x;\mathcal{D})-y(x))^2\right] &= \left(\mathbb{E}_{\mathcal{D}}[\hat{y}(x;\mathcal{D})]-y(x)\right)^2+\mathbb{E}_{\mathcal{D}}\left[\left(y(x)-\mathbb{E}_{\mathcal{D}}[\hat{y}(x;\mathcal{D})]\right)^2\right]\\ &= \text{Bias}^2+\text{Variance}\,.\end{aligned} \tag{3.1}$$

The bias $\mathbb{E}_{\mathcal{D}}[\hat{y}(x;\mathcal{D})] - y(x)$ measures the deviation of the average prediction from the true label $y(x)$. The variance $\mathbb{E}_{\mathcal{D}}\left[(y(x) - \mathbb{E}_{\mathcal{D}}[\hat{y}(x;\mathcal{D})])^2\right]$ is interpreted as the sensitivity of the model performance to the choice of the dataset $\mathcal{D}$.

Typically, bias and variance depend on the **capacity** of the model, which is its ability to represent complex relationships in the data. Increasing the number of parameters, adding input features, or switching to a more powerful class of models are modeling choices that would increase model capacity. Models with high capacity are able to learn fine patterns in the training data, and thus achieve low training errors and low bias. However, this also means that they have high variance since they tend to learn patterns that are specific to the training set and are not relevant to the full distribution. This phenomenon is called **overfitting**. On the other hand, low capacity models are prone to **underfitting**, meaning they fail to struggle to even fit the training set. They tend to have high bias but low variance.

Figure 3.1 illustrates how the training and test errors of a model evolve with its capacity, as presented in classical machine learning theory [29, 133]. As capacity grows, the model is able to learn increasingly complex patterns and its training and test errors decrease. However, if the capacity becomes too large, the model will start to overfit on the training data. Although the training error continues to improve, the test error starts to plateau or even increase. The difference between the test and training error is called the **generalization gap**. The best-performing model achieves the lowest test error, which corresponds to a capacity that is neither too low nor too high. Equation 3.1 underlines that this corresponds to a tradeoff between bias and variance.

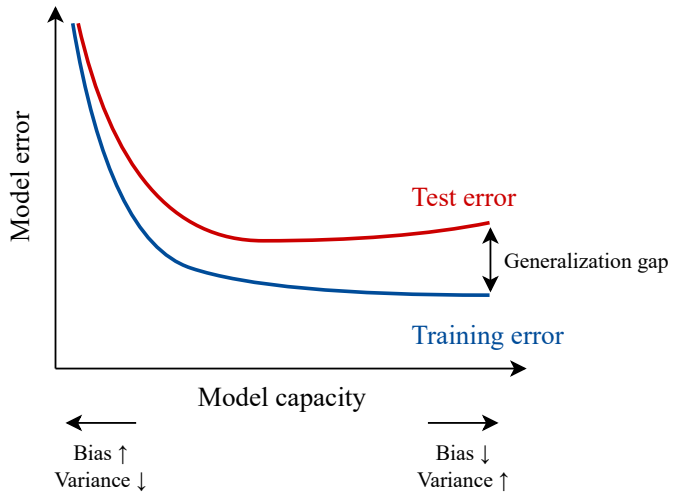


Figure 3.1: Bias-variance tradeoff.

To find the right tradeoff, there is no set formula, and a balance between the complexity of the learning task and the capacity of the model must be found. For large datasets with high-dimensional inputs and outputs, it is important to choose a model with large capacity, and model selection plays a key role in successfully building a predictive model. Techniques to mitigate overfitting include stopping training when the validation loss starts to plateau, penalizing the magnitude of the weights (weight decay), or using data augmentation, *i.e.*

increasing the diversity of training data by applying small perturbations that do not alter the training objective.

The zoology of machine learning models is vast [133], and includes linear models (ordinary least squares, logistic regression), support vector machines, decision trees (random forests, gradient boosted trees), and neural networks. This last category will be at the center of this work since it has seen an extensive revival over the past decade with the emergence of deep learning.

3.1.5 Deep learning

Neural networks were born in the middle of the 20$^{\text{th}}$ century [223, 316] from the idea of building digital analogs to the circuitry of animal brains. They rose to prominence in the 21$^{\text{st}}$ century thanks to the exponential growth of computing power and data availability, which eventually reached the critical mass needed to routinely train deep neural network architectures. The era of deep learning was famously ushered in by a breakthrough success in the 2012 ImageNet Large Scale Visual Recognition Challenge, a yearly competition that evaluates algorithms on visual recognition tasks, with images taken from the ImageNet annotated image database [151]. A deep convolutional neural network won the competition with a classification error of 16%, a significant improvement the previous mark of 25%, beating machine learning and rule-based image recognition algorithms [176]. Since then, progress in hardware efficiency, especially for powerful general purpose Graphics Processing Units (GPUs), and the development of open-source libraries such as Tensorflow [2] and PyTorch [261] have enabled immense advances in the scale of neural networks that can be trained and the accessibility of deep learning to newcomers. Deep neural networks are at the core of major developments in computer vision [31, 155, 299], natural language processing [38, 72], recommender systems [49], and even scientific discovery [154].

3.2 Deep neural networks

3.2.1 Dense layers

Neural networks are composed of a succession of **layers** that perform simple operations on their input **feature map**, and pass on the resulting output feature map to the next layer. The most simple neural network architecture is the multilayer perceptron (MLP) which only contains **dense** layers. For an MLP with n layers $\mathcal{L}_1, ..., \mathcal{L}_n$, the dense layer $\mathcal{L}_i$ transforms the feature map $\boldsymbol{x_i} \in \mathbb{R}^{d_i}$ as:

$$\boxed{\boldsymbol{x_{i+1}} = \mathcal{L}_i(\boldsymbol{x_i}) = g_i(\boldsymbol{W_i}^T \boldsymbol{x_i} + \boldsymbol{b_i})\,.} \tag{3.2}$$

$\boldsymbol{W_i} \in \mathbb{R}^{d_i \times d_{i+1}}$, $\boldsymbol{b_i} \in \mathbb{R}^{d_{i+1}}$, g_i are the weight matrix, bias vector, and activation function of layer i. Weights and biases form the parameters $\boldsymbol{\theta_i} = (\boldsymbol{W_i}, \boldsymbol{b_i})$ that are adjusted when training the network. If the activation function is linear, then $\mathcal{L}_i$ is simply a linear transformation. Instead, using a non-linear activation function allows for greater model expressivity (*i.e.* higher capacity) as the model will be able to capture non-linear relationships. The

rectified linear unit (ReLU) [244], defined as an elementwise maximum operator

$$\sigma(\boldsymbol{x}) = \max(0, \cdot) \odot \boldsymbol{x}\,, \tag{3.3}$$

is one of the most common activation functions. An MLP that only contains ReLU activations is therefore a piecewise linear function that transforms input features $\boldsymbol{x}$ as:

$$\hat{\boldsymbol{y}}_{\boldsymbol{\theta}} = (\mathcal{L}_n \circ ... \circ \mathcal{L}_1)(\boldsymbol{x}) \tag{3.4}$$

and is parametrized by $\boldsymbol{\theta} = (\boldsymbol{\theta_1}, ..., \boldsymbol{\theta_n})$. n is the **depth** of the network. Quantities that characterize the network but that are not directly optimized during training such as the number of layers or their dimensionality are called **hyperparameters**.

3.2.2 Stochastic gradient descent

Neural networks are trained by iterative gradient descent, an optimization procedure that seeks the parameters $\boldsymbol{\theta}$ that minimize a loss function L over the training dataset. L is chosen in accordance with the learning task. For the regression problems tackled in this work, the loss function is a mean squared error between the network predictions and the ground truth:

$$\boxed{L(\boldsymbol{x}; \boldsymbol{\theta}) = (\hat{\boldsymbol{y}}_{\boldsymbol{\theta}} - \boldsymbol{y})^2\,.} \tag{3.5}$$

In practice, when training on large datasets, gradient descent over the entire training set is prohibitively expensive because it requires a gradient evaluation at every training data point. The preferred method is **stochastic gradient descent** (SGD), where the training set is randomly split into small batches and gradient descent is done on one batch at a time. This amounts to approximating the real gradient with its value over one batch. A parameter update on one batch is called an **iteration**, and a pass through the entire training set is called an **epoch**.

During a pass over a batch $\{\boldsymbol{x^{(1)}}, ..., \boldsymbol{x^{(m)}}\}$, the mean gradient of L with respect to $\boldsymbol{\theta}$ is accumulated by automatic differentiation. $\boldsymbol{\theta}$ is then updated in the direction of steepest descent:

$$\boxed{\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon \nabla_{\boldsymbol{\theta}} \left(\frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\boldsymbol{x}^{(i)}; \boldsymbol{\theta}) \right)\,,} \tag{3.6}$$

where ϵ is a hyperparameter called the learning rate. This is the backpropagation step. Algorithm 1 shows pseudo-code for the base SGD algorithm.

Building from the concept of stochastic gradient optimization, many evolutions of the base SGD algorithm have been proposed in the DL literature. One of the most popular variants is the Adam optimizer [164] which adaptively decays the learning rates of individual parameters. It performs consistently well on a wide set of learning tasks [325] and will be used in all DL training runs of this work.

Algorithm 1: Stochastic gradient descent

Data: Number of iterations N, learning rate schedule ϵ_k, batch size m
Result: Trained parameters $\boldsymbol{\theta}$

for $k \leftarrow 1$ **to** N **do**
  Randomly draw a batch $\{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(m)}\}$ from the training set
  Compute the gradient approximation: $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{\theta}} \left(\frac{1}{m} \sum_{i=1}^{m} L(\boldsymbol{x}^{(i)}, \boldsymbol{\theta}) \right)$
  Update the parameters: $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \epsilon_k \boldsymbol{g}$

3.2.3 Training and validation datasets

To detect underfitting and overfitting, the data available during training is split into a training set and a validation set. Parameters are optimized only on the training set, and the loss function on the validation set is monitored throughout the training. As training progresses, the training loss will consistently decrease, while the validation loss may start to plateau or increase after enough iterations have passed: this is a sign that the model has started to overfit. Typically, the model with the lowest validation loss is selected. Hyperparameters may be adjusted to improve the lowest validation loss, at the cost of needing to retrain the model for each set of hyperparameters. Underfitting is solved by adding more parameters or layers to the network to increase its capacity. Overfitting is addressed by reducing the model's capacity, either by directly removing parameters and layers, or indirectly through regularization methods like weight decay [177] and dropout [350]. The final performance of the model should be assessed on a test set that is not used for parameter or hyperparameter tuning.

3.2.4 Why deep neural networks are able to generalize

Despite the widespread usage of deep learning, how and why deep neural networks generalize is still not well understood. Modern deep neural networks can contain trillions of parameters [91] and are typically overparametrized with respect to their training dataset. The fact that overparametrized neural networks are able to fit random labels is at odds with all traditional machine learning explanations for generalization [411]. It implies that DL models learn to memorize training labels, seemingly contradicting their ability to extrapolate to unseen test data. This is corroborated by an empirical phenomenon known as double descent. Unlike the monotonous test error curve of Figure 3.1, the test error of deep neural networks starts to decrease again when model capacity is sufficiently large [25, 247]. Double descent is consistent with the ability of modern overparametrized neural networks to reach near zero training error while avoiding overfitting, even without explicit regularization. To reconcile memorization and generalization, Feldman et al. [92, 93] argue that since natural data distributions are long-tailed, memorization of training labels is necessary to achieve low training and generalization errors. Overparametrization may also be crucial to explain why deep neural networks can converge to arbitrarily low training errors despite the non-smoothness of their optimization landscape [8].

3.2.5 Selecting the right architecture

A key element of the success of deep neural networks is the flexibility of their architecture, as increasing their depth or the dimensionality (width) of the inner layers allows them to scale to very high capacity and thus tackle complex learning tasks. In this regard, MLPs suffer from an explosion of the number of parameters when working on high-dimensional data such as long time series, images, videos, or multidimensional grids. The size of the weight matrix is the product of the dimension of its inputs and of its outputs, making it an inefficient way to parametrize the network when these dimensions are large.

3.3 Convolutional neural networks

3.3.1 Convolutional layers

For high-dimensional data that is naturally structured on a grid, replacing matrix multiplications with convolutions leads to a much more efficient use of the parameters of a neural network.

A convolutional layer $\mathcal{C}_i$ applies a convolution filter $\boldsymbol{G_i}$ to its input:

$$\boldsymbol{x_{i+1}} = \mathcal{C}_i(\boldsymbol{x_i}) = g_i(\boldsymbol{G_i} * \boldsymbol{x_i} + \boldsymbol{b_i})\,. \tag{3.7}$$

The convolution weights of $\boldsymbol{G_i}$ are the trainable parameters of the layer. Convolutional neural networks (CNNs) are neural networks that contain at least one convolutional layer.

The inner workings of CNNs are detailed using the example of a 2D convolutional layer which operates on 2D features. Feature maps in CNNs usually have multiple **channels** to enhance the expressiveness of the network. For instance, color images can be decomposed into red, green, and blue channels that each contain specific information about the image. Channels are stacked along a third dimension. The input $\boldsymbol{x_i}$ of the 2D convolutional layer has a shape $H_i \times W_i \times C_i$ where H_i and W_i are the height and width of each channel, and C_i is the number of channels. $\boldsymbol{G_i}$ is composed of C_{i+1} convolution kernels of shape $h_i \times w_i \times C_i$. The convolution operation is illustrated in Figure 3.2. Each channel $j, 1 \leq j \leq C_{i+1}$ of the output $\boldsymbol{x_{i+1}}$ is obtained by convolving the j-th kernel and $\boldsymbol{x_i}$, and applying the bias and activation function. A 3D convolutional layer works in the same manner, only with an additional depth dimension. In practice, a batch dimension is also added to process all the batch elements at the same time.

The number of parameters of the convolutional layer is $h_i \times w_i \times C_i \times C_{i+1}$. Crucially, it does not depend on the dimension of $\boldsymbol{x_i}$. The size of the convolution kernels and the number of channels are hyperparameters that are defined by the architecture of the network. Generally, $h_i = w_i$ for all kernels.

3.3.2 Padding

Figure 3.2 shows that convolutions reduce the size of the output feature maps ($H_{i+1} < H_i$, $W_{i+1} < W_i$). To preserve the size of the feature map, padding can be applied prior to the convolution operation. This allows the sizes of the output feature map and of the

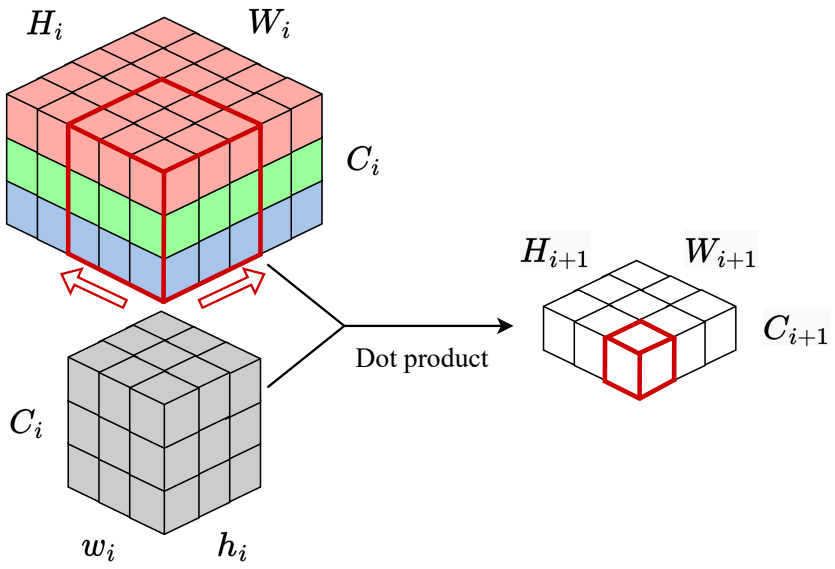


Figure 3.2: 2D convolutional layer. A filter of size $h_i \times w_i \times C_i = 3 \times 3 \times 3$ is convolved with the input feature map ($H_i = W_i = 5$, $C_i = 3$). Each value in the output feature map is computed as the dot product of the filter with a $h_i \times w_i \times C_i$ patch of the input (example outlined in dark red). The output has a shape $H_{i+1} \times W_{i+1} \times C_{i+1} = 3 \times 3 \times 1$.

convolutional kernels to be controlled independently. Most commonly, spatial dimensions of the input are padded with zeros at the boundaries (zero-padding).

3.3.3 Pooling

Learning from high-dimensional data requires some form of compression to find data representations that efficiently encode information that is useful to the learning task, while discarding the rest. To this end, it is beneficial to reduce the size of deep feature maps to encourage them to learn more abstract data representations. In parallel, the number of channels is usually increased to increase the diversity of feature maps. In practice, pooling layers are used to reduce the dimension of feature maps. For example, $k \times k$ max pooling applies a maximum operation over a $k \times k$ neighborhood with a stride k, effectively dividing each dimension by a factor k. Pooling also improves the computational efficiency of the network by reducing the number of parameters needed in subsequent layers. Finally, pooling has a large impact on the receptive field of the network.

3.3.4 Receptive field

The receptive field of a CNN is the region of its input that can influence the prediction at a given output location [115]. From Figure 3.2, it is clear that the receptive field of the output of a convolutional layer is proportional to the size of the convolutional kernels. A pooling layer with a size k multiplies the receptive field by k. The receptive field of a CNN containing a sequence of convolutional and pooling layers can therefore be very large.

In other words, predictions at a given location take into account a wide spatial context. Further investigations into the receptive field are done in Section 5.8.1.

3.3.5 Batch normalization

Normalization layers have long been known to increases convergence speed and stability of deep neural networks. For instance, AlexNet use a local response normalization layer [176]. In modern deep learning practice, batch normalization (BN) [148] has emerged as the main normalization technique used in CNNs. BN layers are placed after dense or convolutional layers to stabilize the mean and variance of the feature map. During training, the mean μ_i and standard deviation σ_i of the input feature map $\boldsymbol{x_i}$ are aggregated along the batch and spatial dimensions. For each channel, the BN layer rescales the feature map:

$$\boldsymbol{x_{i+1}} = \mathcal{B}_i(\boldsymbol{x_i}) = \gamma_i \frac{\boldsymbol{x_i} - \mu_i}{\sigma_i} + \beta_i \,. \tag{3.8}$$

γ_i and β_i are a scale and a shift parameter that are trainable. In inference mode (when the model is used on the validation or test set), values of μ_i and σ_i saved from the training set are used to prevent data leakage.

One of the explanations for the success of BN is that it helps the optimization process by smoothing the loss landscape [322]. This also explains why it empirically solves gradient explosion or vanishing. In addition, BN has a regularization effect which tends to limit the capacity of the model and prevent overfitting [212].

3.3.6 Why CNNs are useful

Supervised training of neural networks is a form of inductive learning, for which generalization depends on the inductive biases of the model [119]. These are the factors outside of the observed data that intrinsically steer the model towards learning a specific representation. Generalization is largely driven by how well the model's inductive biases fit the properties of the data representation it is trained to learn. The inductive biases of neural networks are heavily influenced by their architecture. MLPs have weak inductive biases, whereas CNNs have strong locality and translation equivariance inductive biases [23] which explains their success in generalization of computer vision tasks [410].

Locality is enforced by the limited extent of convolutional kernels. Unlike mechanisms such as attention [368] that enable long-range dependencies, convolutions enforce local relationships, *i.e.* limited to a restricted spatial neighborhood. This property is desirable when building an LES SGS combustion model, as the evolution of the flame should be fully determined by local properties. In contrast, applications of CNNs to the non-local problem of solving the elliptic Poisson differential equation showed that in this case, very large receptive fields are needed to combat the locality bias of convolutional layers [56].

In a convolutional layer, the same set of weights is applied to every input location, instead of having to learn location-dependent weights. This property is called parameter sharing and makes CNNs very efficient to deal with high-dimensional features where absolute positions are not relevant to the learning task. In particular, this work applies DL models to volumetric

data discretized on 3D grids where convolutions are much more efficient than dense layers in terms of computation and memory. Parameter sharing also makes convolutional layers equivariant to translation. This means that a convolutional layer $\mathcal{C}$ and a spatial translation t commute:

$$(\mathcal{C} \circ t)(\boldsymbol{x}) = (t \circ \mathcal{C})(\boldsymbol{x})\,. \tag{3.9}$$

In an SGS model working on the resolved flow, if the flow is translated, the corresponding SGS field should follow the same translation. Translation equivariance is therefore a useful property for this task. In general, generalization of data-driven models is greatly improved by leveraging invariance or equivariance to physical transformations. Intuitively, physical invariance reduces the complexity of the modeling task and leads to perfect generalization to invariant transformations of the data.

CNNs benefit greatly from deep architectures which stack many convolutional layers. In practice, shallow convolutional layers have been observed to learn Gabor filters, which naturally occur in the visual cortex of mammals and are often chosen to extract image features in hand-made image classifiers [115]. Moving deeper into the network, convolution filters learn increasingly complex concepts. Feature visualization is a technique that highlights what specific filter, channels, or layers learn to detect. Figure 3.3 is taken from Olah et al. [255] where it is applied at the level of a channel. Gradient ascent in the input feature space is performed to find the input that maximizes the activation of a given channel in a CNN [85]. In this case, the network is a GoogleNet architecture [356] trained for object classification on the ImageNet dataset [151]. Shallow layers are sensitive to textures, while deeper layers respond more to increasingly abstract and generic concepts (dog eyes, a house, balls).

(**a**) Layer 3a (**b**) Layer 4a (**c**) Layer 4c (**d**) Layer 5a

Figure 3.3: Feature visualization of channels of GoogleNet [356] trained on the ImageNet dataset [151]. Going from left to right, layer depth increases. For each layer, one channel is selected, and the input that maximizes the activation of this channel is shown. Figures from Olah et al. [255] licensed under CC BY 4.0.

3.4 U-Nets

U-Nets are convolutional neural network architectures that are designed for field-to-field visual tasks, and are commonly used in computer vision for 2D and 3D image segmentation [64, 315, 416]. They follow the structure of an autoencoder, with an encoder part that progressively reduces the dimension of the feature maps and increases the number of chan-

nels, and a symmetrical decoder part that maps to an output with the same dimension as the input. The network is split into stages separated by pooling operations that downsample (in the encoder) or upsample (in the decoder) the feature maps. The specificity of U-Nets lies in the skip connections that concatenate the end of each encoder stage to the start of the matching decoder stage. The combination of upsampling and concatenation encourages each decoder stage to aggregate information from two separate spatial scales. U-Nets are therefore multiscale CNNs that combine information on a continuum of scales extending up to their receptive field. Skip connections also allow the network to combine low-level features learned by the shallow layers of the encoder with the more complex, abstract features learned by the decoder, and accelerate training convergence [416].

The architecture of the U-Net used throughout this work is shown in Figure 3.4. Each stage is composed of two successive combinations of:

- a 3D convolution with a $3 \times 3 \times 3$ kernel,
- a batch normalization layer,
- a ReLU activation function,

followed by $2 \times 2 \times 2$ pooling operations. In the encoder, maxpooling operations decrease the spatial dimensions of the feature maps by a factor of 2. The shape of the input field is then recovered by upsampling pooling operations in the decoder. All the convolutional layers include zero-padding to preserve the spatial dimensions of their input feature map. Padding could be adapted to specific boundary conditions if they were known in advance [7], for instance for periodic boundaries. However, one of the strengths of this architecture is that it is fully convolutional, meaning there are no constraints on the shape of the inputs it can process. This allows training to be performed on random crops for which no specific boundary conditions exist. Previous studies working on CNNs trained for flow field predictions have found no substantial impact of zero padding compared to padding strategies that are consistent with their boundary conditions [234].

This architecture follows conventional best practices for deep CNNs. One of them consists in stacking consecutive small $3 \times 3 \times 3$ kernels instead of using a single equivalent larger kernel. Using several small kernels has the advantage of requiring fewer parameters to reach a given receptive field size, which can accelerate and regularize the training of deep CNNs. This is a key concept in the design of the Inception module [356], the VGG architecture [341], and ResNets [137]. Batch normalization was found to be mandatory for training to converge. Without BN, training stalls as soon as the first epoch. Whether BN should be placed before or after the ReLU non-linearity is a matter of debate, but it has not been observed to significantly affect performance here or in the literature. Although it is arguable that BN $\rightarrow$ ReLU makes sense to ensure that the following convolutional layer sees normalized data, here ReLU $\rightarrow$ BN was chosen in accordance with the original paper [148]. This network contains a total of 1.4 million trainable parameters.

In the applications presented in this manuscript, U-Nets are trained to map a resolved LES field to an SGS field of interest. This is a 3D field-to-field regression task for which the U-Net architecture is well-adapted. Multiscale CNN architectures were successfully used for CFD modeling by Lapeyre et al. [183] to predict SGS flame wrinkling and Ajuria et al. [4]

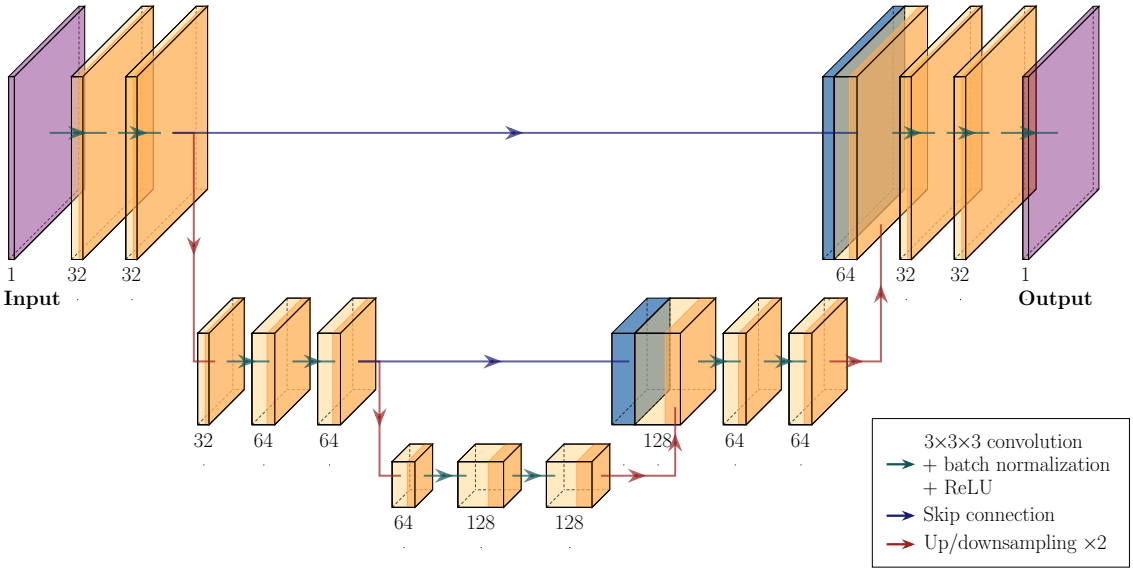


Figure 3.4: U-Net architecture. Each feature map is represented with 2 out of its 3 spatial dimensions as height and depth, and the channel dimension as width. The number of channels is noted below each feature map. The hidden layers connecting the feature maps are represented by the arrows. The input and output fields are colored in pink, and the feature maps concatenated by the skip connections are colored in blue.

to solve the Poisson equation in incompressible flows. In the following, the U-Net model will simply be called the CNN. For SGS modeling, this multiscale approach can be thought of as an extension of the dynamic procedure which extracts information at a test-filtered scale to inform local predictions. Its ability to predict the entire output field at once is also attractive in terms of computational efficiency, as a single inference of the network is needed for the whole domain. This is an appealing approach for combustion SGS modeling since the CNN is able to make local predictions based on the knowledge of the full spatial structure of the flame front.

Chapter 4

Deep learning for numerical combustion

Contents

In a few short years, deep learning has become one of the hottest topics in scientific computing and is now widely applied to enhance numerical fluid simulations. This chapter closes the introductory part of this thesis with a survey of the existing literature to place this work in the appropriate context, and expands on the methodology and modeling framework used in the rest of the thesis. First, a short review on the state of the art of deep learning applications in CFD is established, followed by a more specific examination of advances in numerical combustion. A methodology to train and evaluate DL SGS models is detailed. Finally, Part I ends with a description of the modeling framework used in the three DL models developed in the thesis.

4.1 Deep learning for CFD

Like many scientific fields, CFD has widely adopted deep learning as a powerful modeling tool that opens the door to fascinating new applications. In this section, some recent advances in the field are present, and the reader is referred to the many existing reviews for more details [39, 80, 81, 373].

4.1.1 Learning useful flow representations

The large capacity of deep neural networks enables them to learn flexible representations of fluid flows. One task where this is useful is super-resolution, where coarse turbulent fields are upsampled to high-resolution versions. Deep neural networks are trained in a supervised [102, 103, 199] or unsupervised [107, 162] fashion to generate realistic fully resolved turbulent flows. While super-resolution increases the dimensionality of a flow representation, reduced-order modeling seeks a compressed, low-order representation of the flow that can be used whenever high-fidelity flows are unwieldy. Existing techniques include proper orthogonal decomposition [210] and dynamic mode decomposition [324]. With deep learning, convolutional autoencoders are trained to encode a compressed representation of fluid flows into a low-dimensional latent vector, corresponding to the feature map at the bottleneck of their architecture. This latent vector is then used to advance temporal dynamics at a low computational cost [84, 219] or visualize flow fields, effectively performing a type of nonlinear mode decomposition [243]. Some works are dedicated to assessing the uncertainty of reduced-order models involving neural networks [104, 218].

4.1.2 Predicting spatio-temporal dynamics

Learning a model for the evolution of turbulent flows could disrupt the field of numerical simulation by replacing traditional CFD solvers which are bounded by slow convergence or iterative temporal iterations. For steady-state flows, several methods learn the converged flow state by encoding the geometry of the domain using signed distance functions [127] or point cloud representations [157]. Spatio-temporal modeling of unsteady flows is a topic of active research, with extensive works aiming to predict the temporal evolution of turbulent flows [232, 245, 304, 349, 351, 387] with applications like turbulence inflow generation [105, 163]. All these methods displace the computational burden from the inference phase (rolling out temporal iterations) to the training phase of a deep neural network.

Wandel et al. [384] propose an unsupervised training framework where fluid data is cycled through a CNN until a physics-constrained loss function converges. This is analogous to many recent methods that have been proposed to solve partial differential equations (PDEs) in a data-free manner, *i.e.* without needing paired input-output data that are generated from costly high-fidelity simulations. This is achieved by sampling random solution states and minimizing a physics-constrained loss function that includes the residual of the PDE as well as Dirichlet or Neumann boundary conditions [179, 417]. Along with finite difference schemes to approximate PDE derivatives, this method was used to train deep learning PDE solvers based on MLPs [342] or auto-regressive convolutional architectures [110]. Physics-informed neural networks [295] use automatic differentiation instead of finite difference schemes to

analytically compute derivatives in the physics-constrained loss function. They are a mesh-independent class of DL PDE solvers that have been applied to inverse problems [128], modeling vortex-induced vibrations [296] and turbulent scalar mixing [294], and solving the Navier-Stokes equations [153]. In practice, they can face limitations due to difficulties in learning unsteady dynamics, high training costs, the need to retrain them when changing parameters or boundary conditions, imperfect numerical accuracy, and sometimes failure to train at all [389].

More generally, the intersection of deep learning and differential equations is a promising research area. There are fundamental analogies between residual neural networks [137] and ordinary differential equations (ODEs). Skip connections in a residual network can be interpreted as an Euler scheme to discretize first-order derivatives [206], enabling generative models known as continuous normalizing flows that leverage ODE solvers [52] and are connected to diffusion models [144]. Fourier neural operators [193] are DL PDE solvers that learn the solution operator of a PDE as a succession of kernel integral operators containing learnable transformations in Fourier space. Like physics-informed neural networks, they are mesh- and resolution-independent, but have the additional ability to learn operators for a class of parametrized PDEs, and not only a single PDE. Two of their drawbacks are their reliance on paired input-output training data, and their large memory usage which still prevents them from scaling to large 3D applications.

4.1.3 Accelerating CFD solvers

Instead of outright replacing CFD solvers, some works focus on using deep learning to accelerate existing solvers. Tompson et al. [360] train a CNN to approximate the solution of the Poisson equation in incompressible flows, which is commonly solved using iterative solvers. Ajuria et al. [4] extend this method by enforcing stricter accuracy requirements. They develop a hybrid approach in which an iterative Jacobi solver is initialized with approximate solutions given by a multi-scale CNN. This significantly reduces the number of Jacobi iterations needed for convergence, thus accelerating the Poisson solver. Applications of multi-scale CNNs to predict acoustic wave propagation also show that replacing traditional solvers with CNNs could remove CFL restrictions on the solver timestep, enabling faster simulations [6]. Bar-Sinai et al. [20] propose to learn the coefficients of the stencils that discretize PDE differential operators. For coarse spatial and temporal resolutions, the resulting data-driven discretizations can improve the numerical accuracy of CFD solvers and relax mesh resolution requirements, thus accelerating computations [169]. Promising research avenues are opened by the development of fully differentiable solvers such as SU2 [83] and *phiflow* [142] that enable end-to-end optimization of ML models developed for PDE solvers [24].

4.1.4 Turbulence modeling

A plethora of approaches have been proposed to train deep neural networks for SGS turbulence modeling in LES, including regression of an SGS model trained by supervised learning [122, 222, 259, 283, 390, 399, 415], model classification to select and optimally blend existing models [220], deconvolution [132, 221, 406], and reinforcement learning [253].

4.1.5 A transverse theme: embedding physical knowledge

Across all the applications of deep learning to CFD, a common theme is the integration of physical knowledge in a data-driven approach. Some methods build physical inductive biases in the design of the model. This applies to physics-informed neural networks that explicitly embed PDEs in the network architecture, or models that use convolutional layers to encourage translation equivariance and locality. Alternatively, physical constraints can be imposed by judiciously selecting the inputs and outputs of the model. In Ling et al. [197], an MLP learns to predict the decomposition of the Reynolds stress anisotropy tensor on an invariant tensor basis, thus ensuring that the resulting tensor is Galilean invariant by construction. Similarly, Prakash et al. [284] express model inputs in a the strain-rate eigenframe to achieve rotation and reflection invariance in a model for SGS stresses. Pre-processing transformations and data augmentation are used by Frezat et al. [101] to encourage invariance to translation, rotation, linearity, and Galilean transformations for predictions of SGS scalar fluxes. For coarse-graining of incompressible turbulence, Mohan et al. [233] predict a vector potential, to ensure that the associated velocity field is divergence-free. Vollant et al. [375] find that MLPs trained to predict the SGS scalar flux divergence generalize better if they are trained to predict coefficients in a physically-derived functional form, rather than the SGS quantity directly. Finally, soft constraints on physical relations can simply be imposed using physics-constrained loss functions [179, 295, 387] or data augmentation [101], but do not ensure that these relations will be consistently satisfied. Hard constraints are more difficult to enforce and may be imposed by neural network verification [289] or corrections applied to the predictions of a trained network [6].

4.2 Deep learning for numerical combustion

Narrowing down the field of CFD to numerical combustion, this section presents a short, non-exhaustive overview of machine learning applications in combustion simulations. See Ihme et al. [146] for a recent in-depth review of this topic.

Some of the first applications of machine learning to combustion targeted the reduction of complex mechanisms. Neural networks in particular are often proposed to replace look-up tables in tabulated chemistry for significant improvements in memory efficiency and speed [9, 30, 47, 48, 61, 74, 99, 131, 147, 257, 331, 382, 412]. Some chemistry modeling approaches transform the thermochemical state into a low-dimensional embedding to reduce the number of variables to be transported, especially for complex mechanisms involving many species. While principal component analysis has long been the method of choice to project and reconstruct the full thermochemical variables, neural networks have sometimes been used as reduced-order models to replace it [82, 229, 264]. Reconstruction of high-fidelity reactive flows is made possible by convolutional neural networks trained to perform super-resolution [32] or deconvolution [203, 250]. Finally, using neural networks for turbulent combustion closures has seen a recent surge of interest with the emergence of deep learning.

More specifically, deep learning for SGS models in LES is the subject of this thesis and is covered more extensively here. Table 4.1 reviews some recent works that involve neural networks to model unclosed terms in the LES equations of turbulent reactive flows. Although

many works use MLP architectures that are typically only a few layers deep, they are still included under the term *deep learning* given the relevance of their approach. All of these works involve supervised training of a neural network trained to regress input-output pairs generated from a filtered DNS, and consistently beat existing algebraic models. From the choice of model targets, three distinct families emerge: FSD models [183, 305, 336], PDF models [55, 158, 357, 402, 403], and models that directly predict the filtered reaction rate, sometimes with unclosed dissipation terms [152, 203, 246, 329, 337]. Indirect models that rely on the existing FSD and PDF frameworks thereby inject some physical knowledge in the resulting model for the filtered reaction rate, although whether this knowledge should be subsumed in an end-to-end direct model is debatable. Determining which input features to use is typically not done in a formal way. Some approaches choose to rely on a very restricted set of inputs [183, 203, 305, 329], while others opt to use many more features. Several authors perform feature importance analyses to understand which features are more relevant for the model: Yellapantula et al. [403] use integrated gradients [354], while Shin et al. and Kasten et al. [158, 337] use Shapley additive explanations [211] and maximal information coefficients [307]. Finally, despite the beneficial inductive biases and parameter efficiency of CNNs outlined in Section 3.3, CNNs are rarely used, and when so, often with receptive field restrained to 2D or neighbors of a reduced box stencil.

Table 4.1 underlines the rarity of three key assessments of the performance of a DL model:

1. evaluation on a high Reynolds configuration instead of an academic flame,
2. evaluation on a configuration that is different from the training case,
3. *a posteriori* evaluation.

These three elements are key to ensuring the applicability of DL models to industrial LES computations. They are therefore the core problems that are addressed throughout this thesis

The first point is apparent from the list of testing configurations shown in the table. Many works evaluate their model on statistically planar flames or low-Reynolds jet flames. This is understandable, as high Reynolds DNS datasets are costly to generate and not systematically shared with the combustion community. To remedy this issue, Chung et al. [62] have recently launched an open repository for DNS combustion datasets. Models in this thesis will be evaluated on high Reynolds filtered DNS or LES simulations.

Regarding the second point, the limits to generalization of SGS neural network models are still not well understood. Generalization is usually assessed by evaluating the model on the training distribution sampled at different spatial [357] or temporal [32, 55] locations, or through minor parametric variations [55, 183, 250, 402, 403]. For FSD models specifically, Ren et al. [305] study highly turbulent statistically stationary planar flames at $Ka = 38$ (case L), 390 (case M), and 1710 (case H). Cases M and H are located in the broken reaction zone regime, where the flamelet assumption may not hold. Snapshots show a highly fragmented reaction front and the authors point out that the resolved and total FSD fields have large discrepancies for these cases. After training on case H, the model performs well on case M and at larger filter sizes, beating a selection of static wrinkling models. It is interesting to note that it performs relatively poorly on case L which belongs to the thin reaction zone regime

and features an intact reaction zone. This result highlights the model's sensitivity to changes in the turbulent combustion regime. Attili et al. [15] draw similar conclusions after training the U-Net from Lapeyre et al. [183] on four DNS of jet flames with increasing Reynolds numbers [209]. Their results show that generalization to unseen turbulent levels works better between high Reynolds number flames, which they suggest is due to the asymptotic behavior of high Reynolds turbulence. In addition, models trained on a specific region of the flame (flame base, fully turbulent region, or flame tip) perform noticeably worse when tested on a different region, thus highlighting the spatial variations of the wrinkling distribution in a given flame. Generalization to new configurations is addressed in Chapters 6 and 8

Missing literature for the third point can partly be explained by the difficulty of coupling deep neural networks with high-performance LES solvers. *A posteriori* evaluation of DL models is covered in Chapters 7 and 8. The difference between *a priori* and *a posteriori* testing is recalled in the following section.

Reference	Inputs	Targets	Input size	Architecture	Training case / *Test case if different*	*A posteriori* evaluation
Lapeyre et al. [183]	$\bar{c}$	$\overline{\Sigma}$	Full 3D field	CNN (U-Net)	Jet flame (low Re)	–
Seltz et al. [329]	$\dot{\omega}(\tilde{c}), \nabla\cdot(\bar{\rho}D\nabla\tilde{c})$	$\overline{\dot{\omega}}$, SGS scalar flux	3^3 box	CNN	Jet flame (high Re)	–
de Frahan et al. [357]	$\widetilde{Z}, \widetilde{Z''^2}, \tilde{c}, \widetilde{c''^2}$	$p(Z,c\,\vert\,\widetilde{Z}, \widetilde{Z''^2}, \tilde{c}, \widetilde{c''^2})$	Single point	MLP, VAE	Low-swirl burner	–
Yao et al. [402]	$\widetilde{Z}, \tilde{\nu}_t, \vert\vert\widetilde{\underline{\underline{S}}}\vert\vert, \widetilde{D}, \bar{\rho}, \widetilde{J}_m, \widetilde{U}_d, \widetilde{C}$	$p(Z), \langle N\,\vert\,Z\rangle, \widetilde{N}$	Single point	MLP	Spray flame	–
Yellapantula et al. [403]	$\tilde{c}, \widetilde{c''^2}, \widetilde{D}_c, 2\widetilde{D}_c\vert\nabla\tilde{c}\vert^2$, principal rates of strain	$\tilde{\chi}_{c,SGS}$	Single point	MLP	Statistically planar turbulent flame	–
Ren et al. [305]	$\vert\nabla\tilde{c}\vert, u'_\Delta$	$\overline{\Sigma}$	3^2 box	CNN	Statistically planar turbulent flame	–
Chen et al. [55]	$\widetilde{Z}, \widetilde{Z''^2}, \tilde{c}, \widetilde{c''^2}, \widetilde{Z''c''}$	$p(Z,c\,\vert\,\widetilde{Z}, \widetilde{Z''^2}, \tilde{c}, \widetilde{c''^2})$	Single point	MLP	MILD flame	–
Shin et al. [336]	$\bar{c}, \vert\nabla\bar{c}\vert, \vert\nabla^2\bar{c}\vert$	$\overline{\Sigma}$	Single point	MLP (ResNet + MDN)	Jet flame (low Re)	✓
Liu et al. [203]	$\bar{c}$	$\overline{\dot{\omega}}$, SGS scalar flux	2D slices	CNN (autoencoder)	Statistically planar turbulent flame	–
Nakazawa et al. [246]	$\widetilde{Y}_k, \widetilde{T}, \tilde{k}, \tilde{\epsilon}, \bar{\rho}$	$\overline{\dot{\omega}}$	Single point	MLP	Statistically planar turbulent flame / *V-flame*	–
Jigjid et al. [152]	$\overline{\Phi}, \tilde{c}, \vert\nabla\tilde{c}\vert, \exp\left(-\widetilde{Z}/Z_{st}\right)$, $\vert\nabla\widetilde{Z}\vert, \Delta$	$\overline{\dot{\omega}}$	Single point	MLP	MILD flame	–
Shin et al. [337]	$\tilde{c}, \vert\nabla\tilde{c}\vert, \vert\nabla^2\tilde{c}\vert, u'_\Delta, \vert\tilde{\boldsymbol{u}}\vert, \vert\vert\nabla\tilde{\boldsymbol{u}}\vert\vert$, $\vert\vert\widetilde{\underline{\underline{S}}}\vert\vert, \vert\vert\nabla\times\tilde{\boldsymbol{u}}\vert\vert, \tilde{\kappa}, \tilde{a}_t, \Delta$	$\overline{\dot{\omega}}$	Single point	MLP (ResNet)	Statistically planar turbulent flame / *Jet flame (low Re)*	–
Kasten et al. [158]	$\tilde{c}, Da, Ka, \Delta/\delta_L, \tau$	$\tilde{\chi}_{c,SGS}$	Single point	MLP (ResNet)	Statistically planar turbulent flame	–

Table 4.1: List of recent works using neural networks for turbulent combustion closure. Table-specific nomenclature: Z: mixture fraction, $\tilde{\chi}_{c,SGS}$: SGS dissipation rate of c, J_m: evaporation rate, U_d: droplet relative velocity, C: droplet number density, N: scalar dissipation rate, MILD: Moderate or Intense Low-oxygen Dilution, MDN: mixture density network, ϕ: combustion mode, κ: flame curvature, a_t: tangential strain rate, $\tau = (T_{adiab} - T_u)/T_u$.

4.3 Methods for training and evaluating a deep learning SGS model

SGS machine learning models trained via supervised learning require the exact subgrid terms during the training process. By definition, these terms are not available in an LES since they need to be modeled. A solution is to produce resolved and subgrid variables from DNS solutions. This process usually involves applying a spatial filter to the DNS, and downsampling it to a coarse mesh. It is grounded on the analogy between the LES governing equations and filtered DNS equations. The resulting filtered coarse DNS is simply called the **filtered DNS**. The resolved filtered variables can then be used as inputs to the DL model, while the subgrid terms to be modeled are available through the DNS solutions.

There are two main ways of evaluating LES subgrid-scale models [319]:

- *A priori* evaluation requires fully resolved variables from a DNS. The DNS is spatially filtered and downsampled on a coarse mesh to generate coarsely resolved variables on which the model can be applied. The exact subgrid terms can be computed on the coarse mesh based on the DNS variables. They serve as a reference against the predictions of the SGS model.

- *A posteriori* evaluation assesses how the model performs when used in a real LES. This validates the model in the true simulation context in which it is intended to be used, including model propagation errors and LES numerical errors. However, it is not usually possible to validate the model from a one-to-one comparison with experimental or DNS data. Instead, validation relies on statistical comparisons of integral metrics that are relevant to the configuration. For DL models, *a posteriori* evaluation requires the LES solver to be able to query the neural network on-the-fly, which can be technically challenging.

For DL models, *a priori* evaluation on a held-out set of filtered DNS data is straightforward. However, it is important that the filtered DNS is similar to real LES data for *a priori* performance to be indicative of *a posteriori* performance. This applies to any SGS model, but it is especially important for deep neural networks which are high-variance models that could overfit the specific distribution of the training filtered DNS.

4.3.1 Non-equivalence of filtered DNS and LES

Unfortunately, there is no rigorous, straightforward connection between the distributions of filtered DNS and LES variables. First, in implicitly filtered LES solvers like AVBP, filtering is caused by a combination of the effects of the numerical scheme, the SGS model, and discretization by the computational grid. There is no closed-form expression for the resulting implicit filter, so no way to replicate it when explicitly filtering DNS variables for *a priori* testing. Section 4.3.3 will discuss how explicit DNS filtering is done in practice.

Second, even assuming the existence of a perfect explicit filter, filtered DNS and LES will still differ in practice due to numerical and modeling errors. This is formalized by Ghosal's analysis of LES numerical errors [111] which is briefly presented here.

Let ψ denote DNS variables whose time evolution is governed by an exact Navier-Stokes operator $\mathcal{N}$:

$$\frac{\partial \psi}{\partial t} = \mathcal{N}(\psi)\,. \tag{4.1}$$

Filtering and downsampling ψ to a coarse mesh is formalized as the application of a projection operator P mapping DNS solutions to the space of solutions discretized on the coarse mesh[1]:

$$\frac{\partial \mathcal{P}(\psi)}{\partial t} = (\mathcal{P} \circ \mathcal{N})(\psi)\,. \tag{4.2}$$

Let $\mathcal{N}_{\text{LES}}$ be the LES Navier-Stokes operator, including the LES numerical schemes and SGS models. It operates on coarse variables ψ_{LES}. The difference between filtered DNS variables and LES variables $e = \mathcal{P}(\psi) - \psi_{\text{LES}}$ follows a forced Navier-Stokes equation:

$$\frac{\partial e}{\partial t} - \mathcal{N}_{\text{LES}}(e) = (\mathcal{P} \circ \mathcal{N} - \mathcal{N}_{\text{LES}} \circ \mathcal{P})(\psi)\,. \tag{4.3}$$

The forcing term $(\mathcal{P} \circ \mathcal{N} - \mathcal{N}_{\text{LES}} \circ \mathcal{P})(\psi)$ aggregates LES numerical and modeling errors. Even for an ideal SGS model, numerical errors stemming from discrete approximations of derivatives and non-linear terms will remain and prevent e from staying arbitrarily small. In other words, a filtered DNS and an LES of the same flow are not statistically equivalent. This implies some degree of discrepancy between the distributions of realizable filtered DNS and LES variables. In machine learning terms, training a model on a filtered DNS and evaluating it *a posteriori* in an LES is a form of out-of-distribution generalization.

With these theoretical caveats in mind, it remains necessary to transform DNS data in a way that best approximates LES, even if it is not perfect. The main difference between DNS and LES being the mesh resolution, downsampling the DNS to a coarser mesh is the core part of this transformation. However, doing so without pre-filtering the DNS would lead to undesirable **aliasing**, as explained in the following.

4.3.2 Aliasing

Aliasing occurs when a signal of maximum frequency f_{max} is sampled at a frequency f_s such that $f_s/2 \leq f_{max}$, violating the Nyquist-Shannon sampling theorem. $f_s/2$ is the Nyquist frequency, which is the maximum frequency that can be resolved by the sampler. Frequencies in the $[f_s/2, f_{max}]$ range that cannot be resolved are aliased to lower frequencies mirrored across the Nyquist frequency, a phenomenon called **frequency folding**. Figure 4.1 shows an example of frequency folding for a signal characterized by its energy spectrum $E(f)$ in the frequency domain. The original signal is sampled at a frequency f_s, leading to the aliasing of all the frequencies $f \geq f_s/2$ to frequencies lower than the Nyquist frequency. The spectrum of the downsampled signal is the sum of the bandlimited original spectrum and the aliased spectrum, leading to spurious high values near the Nyquist frequency. Aliasing can be mitigated by bandlimiting a signal prior to downsampling. This is achieved by using a low-pass filter that dissipates high-frequency fluctuations while having a limited effect on

[1]Filtering is not the focus of this argument, so it can be assumed that the filter has ideal properties including idempotency and optimal preservation of DNS statistics.

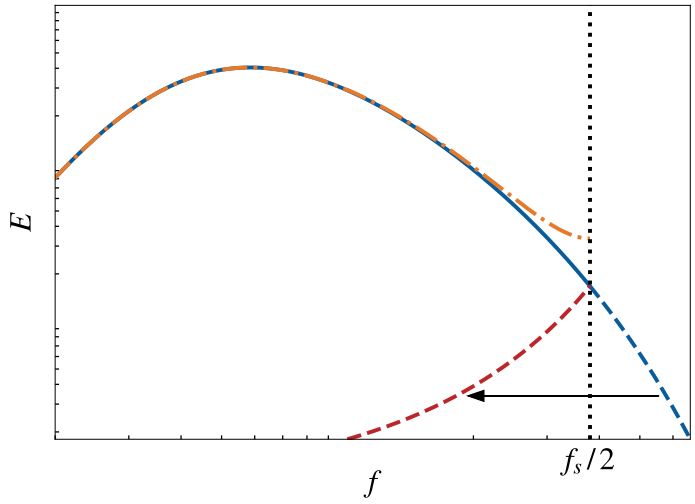


Figure 4.1: Frequency folding due to aliasing. In the original spectrum (blue), frequencies above $f_s/2$ (dashed blue) are aliased to lower frequencies (dashed red), leading to a distorted downsampled spectrum (orange).

low-frequency signal components. The cutoff frequency of the filter marks the transition between its passing and stopping behavior.

Aliasing and the effect of low-pass filtering is illustrated in Figure 4.2 on an RGB image from the Kodak open image dataset[2]. Downsampling the original image by a factor 8 without pre-filtering leads to an aliased representation with a large amount of high-frequency noise. Filtering the original image with a Gaussian kernel before downsampling produces a low-resolution reconstruction that preserves most of the information at the pixel scale.

The effect of pre-filtering is also visible on the energy spectrum of the four images plotted in Figure 4.3. For the downsampled image without pre-filtering, while the energy of low frequencies is well preserved, high frequencies are more energetic than the original image. This is consistent with the large pixel-to-pixel fluctuations observed in Figure 4.2. The filtered image has reduced energy across all but the largest scales. But downsampling the filtered image leads to a spectrum that is more consistent with the original image near the Nyquist frequency.

Like in the RGB images, downsampled 3D DNS data can also suffer from aliasing effects. Figure 4.4 illustrates how aliasing can occur when working with turbulent flame fronts. A DNS flame front with fine-grained wrinkling is downsampled by a ratio $R = 10$ without pre-filtering. The resulting coarse flame front contains high-frequency noise which typically characterizes aliasing. Since this work focuses on training CNNs on coarse turbulent flame fronts, and given the sensitivity of CNNs to high-frequency information such as texture [109, 115, 138], eliminating this noise through filtering is mandatory.

[2]Available at http://www.cs.albany.edu/~xypan/research/snr/Kodak.html

Figure 4.2: Effect of pre-filtering on aliasing of an RGB image. Top left: original image, top right: aliased image, bottom left: filtered image, bottom right: filtered and downsampled image.

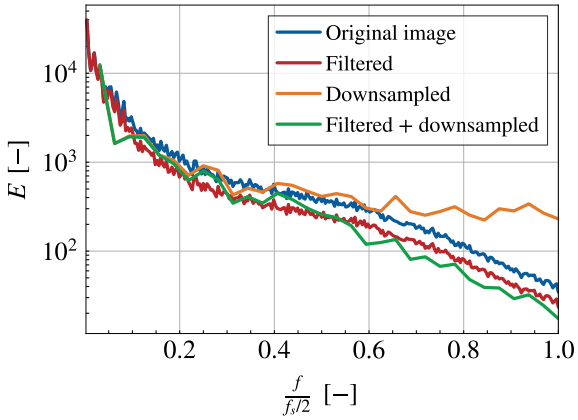


Figure 4.3: Energy spectrum of the four images of Figure 4.2.

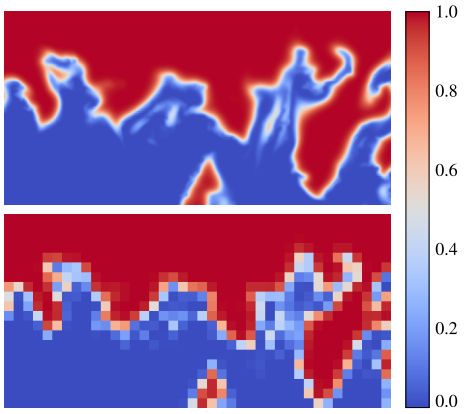


Figure 4.4: Aliasing on a premixed turbulent flame front represented by the progress variable field. Top: DNS flame front, bottom: downsampled aliased flame front.

4.3.3 Filtering

Filtering a signal S is mathematically expressed as a spatial convolution with a kernel G

$$\bar{S}(x) = (G * S)(x) = \int_{-\infty}^{+\infty} G(x - x')S(x')\,\mathrm{d}x'\,. \tag{4.4}$$

In practice, G has finite support, so the integration bounds are finite. $G(x)$ is called the impulse response of the low-pass filter associated with this convolution. It characterizes the effect of the filter in physical space. It satisfies the normalization condition of Equation 2.58.

In the frequency domain parametrized by wavenumbers k, the filtered spectrum $\bar{\mathcal{S}}$ is the product of the kernel transfer function $\mathcal{G}$ and the original spectrum $\mathcal{S}$

$$\bar{\mathcal{S}}(k) = \mathcal{G}(k)\mathcal{S}(k)\,. \tag{4.5}$$

Consider a DNS field resolved on a mesh of size h that must be downsampled by a factor R to a coarse mesh of size Rh. The smallest wavelengths that can be resolved on the coarse mesh have a size $\Delta = 2Rh$. An ideal low-pass filter avoiding any aliasing would suppress spatial frequencies above the cutoff wavenumber $k_c = 2\pi/\Delta$ and preserve any frequency below that. This ideal filter can be expressed as a step function in wavenumber space

$$\mathcal{G}(k) = \begin{cases} 1, & \text{if } |k| \leq k_c, \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

Its impulse response is a sine cardinal (sinc) with infinite support

$$G(x) = \mathrm{sinc}(k_c x) = \frac{\sin(k_c x)}{k_c x}, \quad x \in [-\infty, +\infty]\,. \tag{4.7}$$

The counterpart of the sinc filter is the box filter, which has a step function impulse response

$$G(x) = \begin{cases} \frac{1}{\Delta}, & \text{if } x \in [-\Delta/2, \Delta/2], \\ 0, & \text{otherwise,} \end{cases} \tag{4.8}$$

and a sine cardinal transfer function

$$\mathcal{G}(k) = \mathrm{sinc}(k\Delta/2)\,. \tag{4.9}$$

The Gaussian filter is another common alternative. Its impulse response is a Gaussian

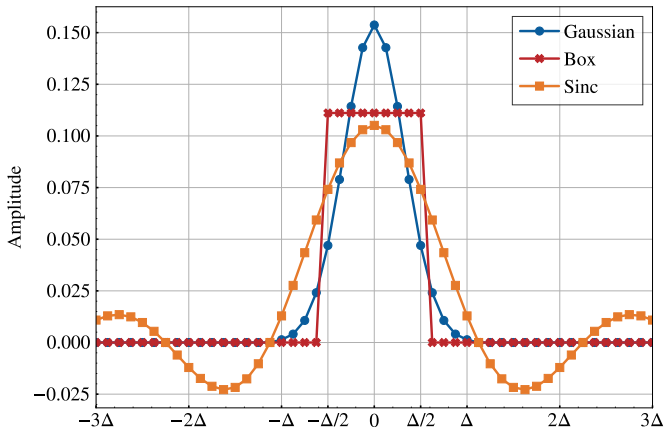
$$G(x) = \left(\frac{6}{\pi\Delta^2}\right)^{1/2} \exp\left[\frac{-6x^2}{\Delta^2}\right] \tag{4.10}$$

with the same standard deviation $\sigma = \Delta/\sqrt{12}$ as a box filter of size Δ. Its frequency response is also Gaussian

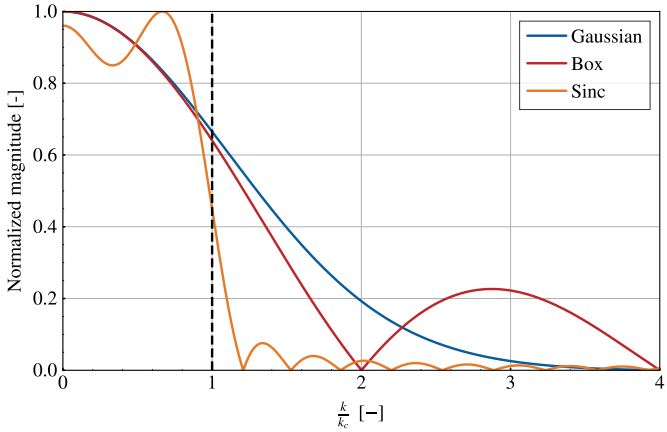
$$\mathcal{G}(k) = \exp\left[\frac{-\Delta^2 k^2}{24}\right]\,. \tag{4.11}$$

The impulse responses of the Gaussian and sinc filters have infinite support, so they must be truncated for practical use. $\sqrt{3}\Delta$ is a good estimate of a truncation width for the Gaussian kernel that preserves most of its non-trivial values. Since the sinc kernel decays slowly to zero, truncating it will result in an imperfect approximation. A longer truncation length will mitigate this imperfection at the cost of increasing the computational cost of the filter due to the increased kernel size.

Figure 4.5 presents the impulse and frequency responses of box, Gaussian and sinc filters with the same cutoff wavenumber k_c. The absolute magnitude of the wavenumber response of the filters is shown in Figure 4.5b. Due to the truncation, the sinc filter loses its ideal low-pass properties. A ringing phenomenon called ripple appears, creating oscillating magnitudes around the Nyquist frequency. The box filter features a large side lobe in the $[2k_c, 4k_c]$ range which could lead to aliasing for these wavenumbers. The Gaussian filter is the only one to impose a smooth, albeit slow, transition around the cutoff wavenumber.



(**a**) Impulse response.



(**b**) Wavenumber response. The dashed vertical line marks the cutoff wavenumber $k_c = \pi/\Delta$.

Figure 4.5: Characteristics of equivalent box, Gaussian and sinc filters.

In the rest of this work, filtering will always be performed using a Gaussian filter unless specified otherwise. Although the box and sinc filters have been used in historical developments of LES [95], the majority of modern LES literature employs Gaussian filtering [188, 313] due to its compact nature and smooth decay at high wavenumbers. Gaussian kernels also have useful mathematical properties. The composition of two Gaussian kernels of widths Δ_1, Δ_2 is also a Gaussian kernel of width $\sqrt{\Delta_1^2 + \Delta_2^2}$. An efficient way to perform multidimensional Gaussian filtering is to convolve the signal with a 1D Gaussian kernel in each dimension. This produces the same result as convolving it with a multidimensional Gaussian kernel while being faster.

4.3.4 Filtering and thickening

Filtering and thickening are two operations that remove high-frequency structures in a flame front. To pre-process filtered DNS data in order to best approach a thickened flame front, the size of the explicit Gaussian filter must be carefully chosen.

TFLES implicitly thickens the flame front by modifying the species conservation equations. This operation acts as an additional filter on the LES progress variable field, but has no explicit closed form. Its cutoff length is equal to the thickened flame thickness $F\delta_L = N_c h$, where N_c, the number of mesh points on which the thickened front is resolved, is typically greater than 5. Thickening therefore supersedes the implicit LES filtering of size $\Delta_{\text{LES}} \sim h$. In a TFLES, the progress variable field is thus not strictly speaking a Favre-averaged filtered representation $\widetilde{c}$ of c, but rather a thickened representation $\check{c}$. This notation will be used whenever it is relevant to distinguish between filtered and thickened progress variable fields.

The effect of thickening on a 1D laminar flame front can be approached by explicitly filtering the flame front with a filter size Δ chosen with the following criterion:

Proposition 4.1 (Filter size of a thickened flame)**.** *When matching thickening at a factor F with a filtering operation of size Δ, Δ is chosen so that a filtered 1D flame has the same maximum progress variable gradient as a thickened 1D flame*

$$\bar{\delta}_{L,c} = \check{\delta}_{L,c}$$

Note that Proposition 4.1 relies on a consistent definition of the progress variable. Figure 4.6 plots the progress variable profile of a thickened 1D flame and of several filtered 1D flames. The original flame is a stoichiometric propane-air premixed 1D laminar flame computed using a single-step mechanism. The progress variable is defined from the fuel mass fraction: $c = 1 - Y_F/Y_F^u$. The laminar flame thickness is $\delta_{L,c} = 352\,\mu\text{m}$. The thickened profile is generated with a thickening factor $F = 5$, leading to a thickened flame thickness $\check{\delta}_{L,c} = 1760\,\mu\text{m}$. Gaussian filtering, Favre-averaged Gaussian filtering, and box filtering are applied to the flame front, with filter sizes tuned according to Proposition 4.1. Despite this constraint, the profiles have noticeable discrepancies near the trailing and leading edges. Differences between the filtered and the thickened profiles are plotted on Figure 4.7. Box filtering leads to flatter leading and trailing edges than in the thickened front. Reynolds- and Favre-averaged Gaussian filtering lead to similar flame profiles that closely match the thickened profile, but with opposite sign differences. These results encourage choosing the

Gaussian filter over the box filter, and suggest that Gaussian filtering with and without Favre-averaging can be suitable approximations of thickening.

Non-Favre-averaged Gaussian filtering will be used throughout this work to generate filtered progress variable fields used as CNN inputs. Since resolved variables in a compressible LES are Favre-averaged, Favre-averaged Gaussian filtering may seem like a more apt choice in the context of *a posteriori* testing. Nevertheless, this choice is not expected to noticeably deteriorate the performance of the model, as the differences between non-Favre-averaged and Favre-averaged flame profiles have been shown above to be small, and LES flame profiles are already very different from Favre-averaged profiles due to the unknown nature of the LES filter and limited flame front resolution. A dedicated *a priori* investigation of the effect of the nature of the filter on the performance of the CNN is performed in Section 5.6.3.1.

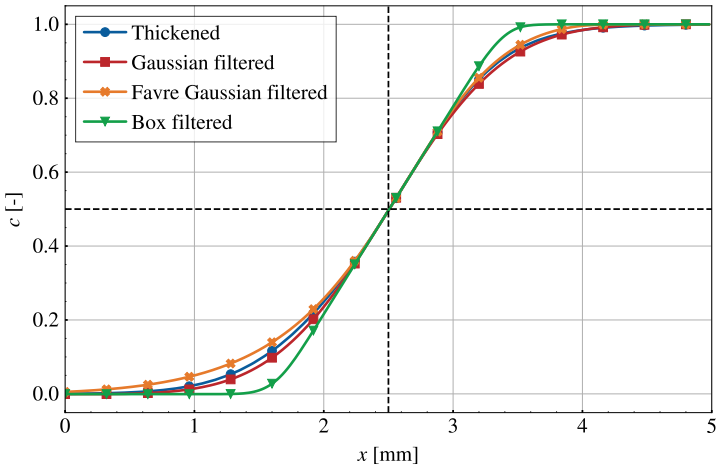


Figure 4.6: Progress variable profiles of thickened and filtered 1D flames.

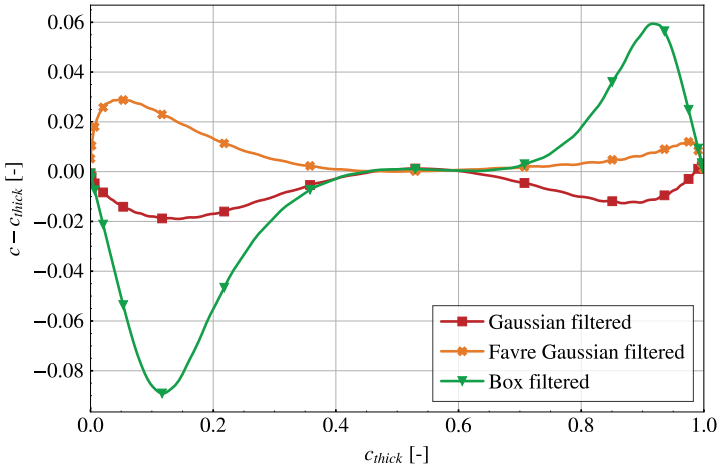


Figure 4.7: Deviation of filtered progress variables to the thickened progress variable.

Proposition 4.2 induces a relationship between the appropriate filter size Δ and the thickening factor F of the thickened flame. This relationship may depend on the progress variable profile induced by the chemistry and on the nature of the filter. To investigate this dependency, 1D flames of a single-step methane-air mechanism [270] and a detailed hydrogen-air mechanism [1] are thickened at various F values and filtered according to Proposition 4.2 for each F. Figure 4.8 then plots the ratio of filter size to thickened flame thickness $\alpha = \Delta/(F\delta_{L,c})$ with respect to F for these two flames. It appears that for both fuels, $\alpha \approx 1.4$ for $F > 5$ with a Gaussian filter, which is consistent with results from the literature on the dynamic Charlette model [287, 377]. For a box filter however, α approaches a value of 1. These observations provide two insights:

- the relationship between Δ and F seems to depend on the nature of the filter, and not on the chemistry,
- this relationship seems to be linear for highly filtered/thickened flames.

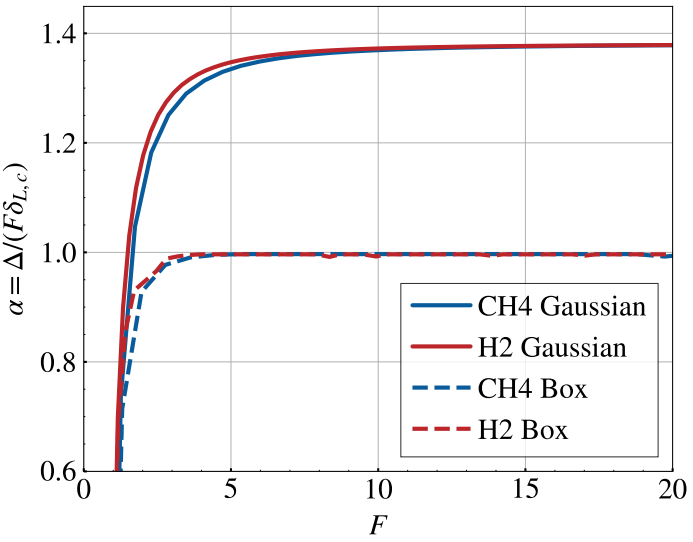


Figure 4.8: Ratio of filter size to filtered flame thickness on 1D flames. Solid: Gaussian filter, dashed: Box filter. Blue: methane-air flame, red: hydrogen-air flame.

This discussion on filtering and thickening exclusively dealt with laminar 1D flame fronts, and it is more delicate to extend to realistic turbulent 3D flames, where additional factors can lead to differences between a thickened and a filtered flame front. On the one hand, thickening is fundamentally an *a posteriori* operation that affects the flame front along its local normal direction. On the other hand, *a priori* Gaussian filtering is isotropic, and therefore averages out wrinkling in the tangential direction. Filtering will also create interferences when a flame front interacts with another flame front or a domain boundary at the scale of the filter size [239]. Despite these limitations, it must be stressed that *a priori* filtering is to date the only way to generate bandlimited LES-like SGS quantities for supervised learning of a machine learning model.

4.3.5 Choosing the filtering parameters

When preparing the training dataset, two variables must be determined: Δ, the filter size which controls the intensity of the filter, and h, the size of the coarse mesh onto which the filtered fields are downsampled. Δ controls the absolute size of the spatial structures in the data. When filtering turbulent flame fronts, larger filter sizes create thicker, less wrinkled flame fronts. This is especially relevant for CNNs, as they learn from the geometrical structures and spatial gradients of their inputs. Normalizing by the progress variable thickness to allow a fair comparison between premixed flames with different laminar thicknesses leads to the ratio $\Delta/\delta_{L,c}$. $\Delta/\delta_{L,c}$ pilots the amount of SGS information that is lost by the filtering process: filtering a thin flame leads to more SGS combustion than filtering a thick flame. Note that the progress variable thickness $\delta_{L,c}$ is used instead of the thermal thickness δ_L. As will be detailed in Section4.4, c is the input of the model, so $\delta_{L,c}$ is truly representative of the flame front thickness seen by the model. This is important in cases where c is not defined as a normalized temperature, and $\delta_{L,c}$ may be significantly different from δ_L. To ensure the consistency of the target SGS quantity predicted by the model, the following criterion is proposed:

Proposition 4.2 (Filter size choice)**.** *The filter size of the training dataset is chosen so that $\Delta/\delta_{L,c}$ is conserved between the training and test dataset.*

A second ratio $\bar{\delta}_{L,c}/h$ determines the resolution of the filtered flame seen by the CNN. Since CNNs work on voxel grids with no innate distance information, it is important that the resolution of the structures they see in the training and evaluation sets is similar [362]. This allows the CNN to generalize from one configuration to another, even if they have different flame thicknesses and DNS mesh sizes. Hence a second criterion:

Proposition 4.3 (Coarse mesh size choice)**.** *The coarse mesh size of the training dataset is chosen so that $\bar{\delta}_{L,c}/h$ is conserved between the training and test dataset.*

If the test dataset is a TFLES simulation, $\bar{\delta}_{L,c}$ is replaced by the thickened flame thickness $\check{\delta}_{L,c}$. Proposition 4.3 enforces consistency of the CNN inputs to ensure that the flame fronts seen by the CNN are resolved on the same number of mesh points.

Propositions 4.2 and 4.3 are combined to find the Δ and h values used to filter the training dataset, even if the training and generalization flames have different laminar thicknesses. In Chapters 5 and 8, the CNN is evaluated on a TFLES flame with a constant thickening factor. In these cases, Algorithm 2 outlines the procedure leading to the filtering parameters. In Chapter 6, the generalization case is a filtered DNS, so Algorithm 2 does not apply and Propositions 4.2 and 4.3 are simply applied (details in Section 6.3).

4.4 Modeling framework of this thesis

In the vast taxonomy of machine learning frameworks presented in Section 3.1, this work falls under supervised regression of 3D fields. It aims to predict the full 3D field of an SGS quantity of interest from 3D fields that are resolved in an LES computation. This SGS quantity is

Algorithm 2: Determination of the filtering parameters

Data: Generalization flame: mesh size h^G, thickness $\delta_{L,c}^G$, thickening factor F
Data: Training flame: thickness $\delta_{L,c}^T$
Result: Filtering parameters for the training flame: Δ^T, h^T

1 $\check{\delta}_{L,c}^G \leftarrow F\delta_{L,c}^G$
2 With the generalization flame chemistry, find the size Δ^G of the Gaussian filter that generates a filtered flame with the thickness $\check{\delta}_{L,c}^G$ (Proposition 4.1)
3 $\Delta^T \leftarrow \Delta^G \delta_{L,c}^T / \delta_{L,c}^G$ (Proposition 4.2)
4 Compute the filtered flame thickness $\bar{\delta}_{L,c}^T$
5 $h^T \leftarrow h^G \bar{\delta}_{L,c}^T / \check{\delta}_{L,c}^G$ (Proposition 4.3)

Paradigm	Supervised learning
Learning task	3D field-to-field regression
Model architecture	U-Net (CNN)
Training data	Filtered DNS
Input	Filtered resolved quantities
Output	SGS quantity
Optimization process	SGD (Adam optimizer)
Loss function	Mean squared error (Equation 3.5)
Evaluation context	*A priori* and *a posteriori*

Table 4.2: Definition of the DL modeling framework.

then used in a physics-based turbulent combustion model to close the filtered reaction source term. Table 4.2 and Figure 4.9 present the main elements of this methodology.

Following this methodology, the characteristics of the three models developed in this thesis are summarized in Table 4.3. Note that the filtered reaction rate is not predicted directly, but deduced from a physical closure. Accordingly, the models do not depend on specific mixture compositions or chemical mechanisms, and could be applied in theory to any premixed flame satisfying the flamelet assumption.

	y	f
Chapter 5	$\overline{\Sigma}^+$	TFLES
Chapter 6	$\overline{c'^2}$	PB-CNN
Chapter 8	Ξ	TFLES

Table 4.3: Overview of the three DL models. Details are found in the corresponding chapters.

This species- and chemistry-agnostic philosophy also governs the choice of input features, which are kept to a minimum. In nearly all the DL models that are investigated, the filtered progress variable $\bar{c}$ is the only input variable. c is a universal parameter that fully parametrizes the thermochemical state of a fully premixed flame. This framework assumes

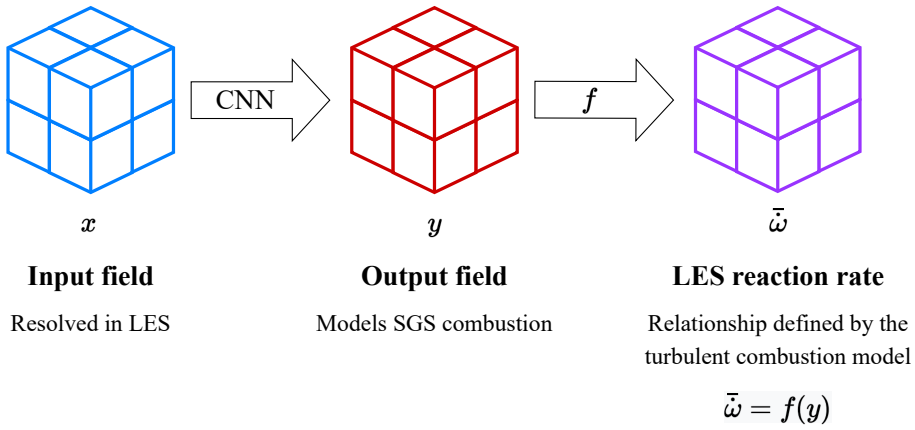


Figure 4.9: Summary of the DL modeling methodology.

that the sole knowledge of c is sufficient to deduce the SGS output field y. This strong assumption is discussed in each chapter individually. In particular, the filter size Δ is a key variable in many traditional SGS models like the CDSM model. It could also be integrated to the CNN, for instance by including Δ as an input feature. However, this creates a risk for improper generalization for test-time Δ values outside of the range seen during training. Instead, the approach presented in Section 4.3.5 is used to set the filter size in the preparation of the training dataset. The downside to this approach is a lack of flexibility regarding to the filter size and mesh resolution of the generalization dataset, which must be known in advance. This is not an issue in the applications presented in this manuscript, but may arise for instance with meshes containing very disparate cell sizes. Chapter 8 experiments with additional input variables, for instance ones that inform on the turbulence intensity. Apart from the choice of a multiscale convolutional architecture, the models are fully data-driven and physics-agnostic. Physical knowledge is embedded in the downstream turbulent combustion model and in the choice of the training dataset.

Part II

A priori evaluation

Chapter 5

High Reynolds testing of a total FSD model

Contents

This chapter builds upon the CNN model proposed in Lapeyre et al. [183] to model the total FSD. The total FSD represents the resolved and SGS flame surface area per unit volume and can be related to a wrinkling factor and a TFLES efficiency function, as explained in Section 2.7.3. The problem formulation of the original paper and its main results are first recalled. This model is then applied to a challenging high Reynolds jet flame configuration, where it is trained to generalize to larger scales and higher Reynolds numbers. The R2 and R3 flames investigated in this chapter are presented. The entire model training pipeline is then detailed, from the preparation of the data and visualizations of comparative statistics to the training of the CNN. *A priori* results on the R2 and R3 datasets are presented, including the effects of variations in the data preparation. An LES of the R3 configuration is performed, and the *a priori* performance of the CNN on snapshots of this LES is compared

to the Charlette model. Finally, interpretations regarding what the CNN sees and learns are sought.

5.1 A deep learning model for the total FSD

This chapter extends the approach of Lapeyre et al. [183] to a new high Reynolds number configuration, and additionally investigates how hard the learning problem is, how the model would perform in *a posteriori* conditions, and what the model actually learns. The problem formulation of Lapeyre et al. is therefore summarized here. They propose a deep learning model for the total flame surface density which is trained by supervised field-to-field regression. Filtered DNS snapshots of a methane-air Bunsen burner are used to generate a training set of matching $\overline{c}$ and $\overline{\Sigma} = \overline{|\nabla c|}$ fields. $\overline{\Sigma}$ is normalized by σ, its maximum value on a 1D laminar flame:

$$\overline{\Sigma}^{+} = \frac{\overline{\Sigma}}{\max(\overline{|\nabla c|}_{1\mathrm{D}})} = \frac{\overline{\Sigma}}{\sigma}\,. \tag{5.1}$$

A deep convolutional neural network with a U-Net architecture is trained to map $\overline{c}$ to $\overline{\Sigma}^{+}$. It is then evaluated *a priori* on a test set drawn from a similar Bunsen simulation where the inlet bulk velocity was modulated to induce large-scale variations in the flame shape over a transient phase. Figure 5.1 shows a slice of progress variable.

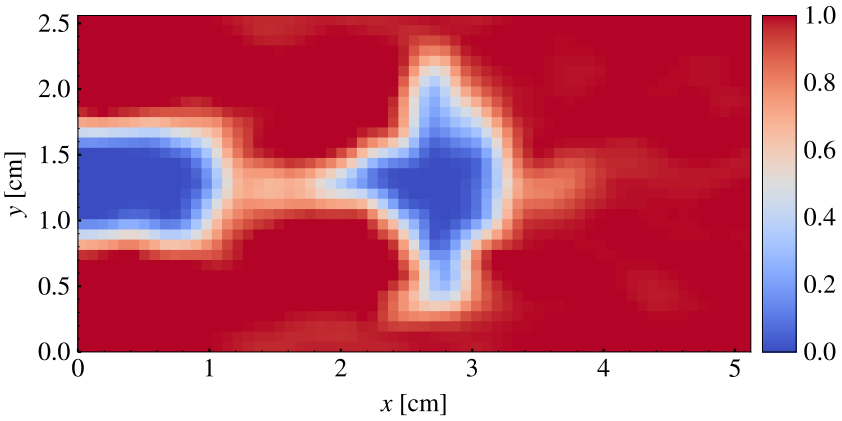
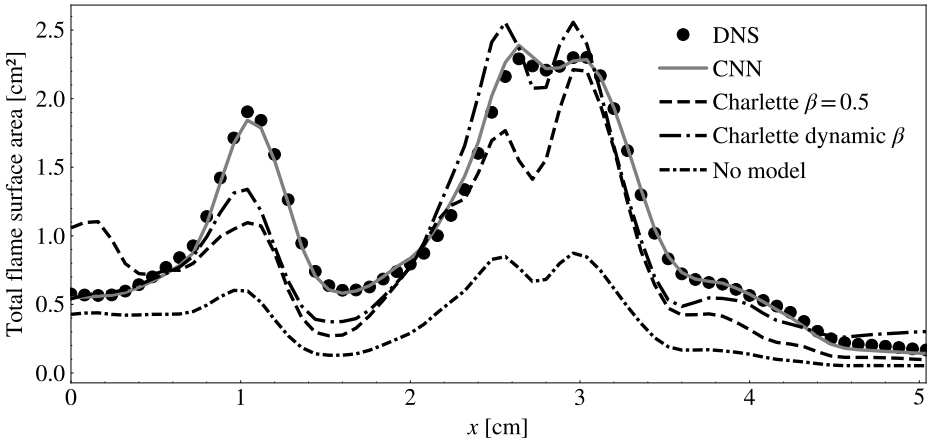


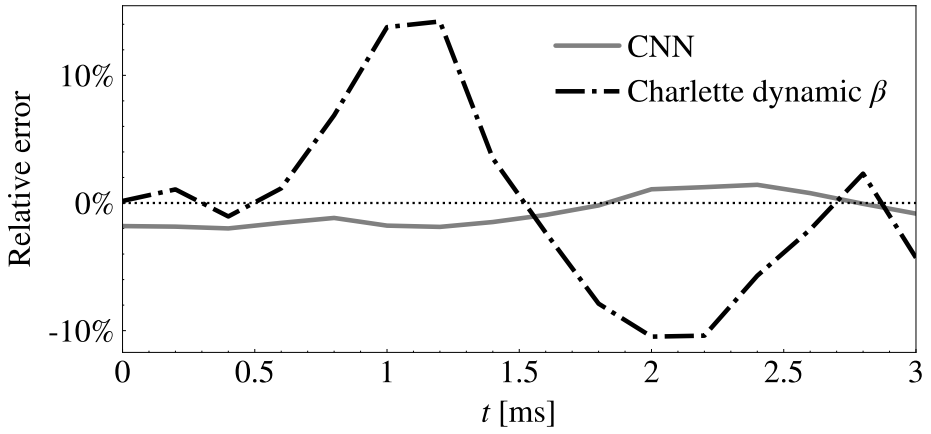
Figure 5.1: Progress variable slice from a filtered DNS in the test set of Lapeyre et al. [183] at $t = 0.8\,\mathrm{ms}$.

Key results on the test set are shown in Figure 5.2. Figure 5.2a plots the streamwise evolution of the total flame surface area predicted on the test set snapshot with the largest DNS total flame surface by the CNN, the static Charlette model ($\beta = 0.5$) and the dynamic Charlette model. These values are computed by integrating the total FSD on transverse slices of the width of a coarse cell. For reference, DNS ground truth total flame surfaces are included, as well as *No model* values obtained from the resolved FSD only. Three distinct regions emerge: a weakly turbulent flame base attached to the inlet ($x \approx 0 - 1.5\,\mathrm{cm}$),

followed by a detached pocket of unburnt gases ($x \approx 1.5 - 4.5\,\text{cm}$) and a postflame region of combustion products with no flame front.



(**a**) Evolution of the total flame surface area along the streamwise x direction at $t = 0.8\,\text{ms}$.



(**b**) Time evolution of the relative error on the domain-integrated total flame surface area.

Figure 5.2: *A priori* evaluation of the Lapeyre et al. CNN.

The static Charlette model with constant $\beta = 0.5$ finds the correct trend but consistently fails to accurately match the DNS flame surface values. The dynamic Charlette model with local β ($\hat{\Delta} = 1.5\Delta$, $\Delta_m = 2\hat{\Delta}$) including corrections from Wang et al. [385] and Mouriaux et al. [239] performs very well in the detached pocket and close to the inlet, but struggles near the tip of the attached flame which features prominent flame front interactions. Finally, the CNN agrees nearly perfectly with the target values in all regions of the domain. Figure 5.2b shows that this behavior is consistent throughout the whole duration of the transient phase, whereas the error made by the Charlette dynamic model fluctuates in time.

This paper demonstrates the viability of deep CNNs for SGS combustion modeling, shows that they can reach excellent *a priori* modeling accuracy for the total FSD, and explores a weak form of generalization through parametric variations on the training configuration.

Building upon this seminal work, this chapter investigates the *a priori* performance of U-Nets on more challenging turbulent cases and assesses their generalization to test sets with greater Reynolds numbers and physical scales than the training set. This marks a first step towards extending the range of applications of these models to large-scale industrial configurations where there is no DNS data on which to train or fine-tune the model.

5.2 Presentation of the R2/R3 flames

The flames that will now be investigated are the R2 and R3 methane-air premixed slot jet flames [207–209]. DNS snapshots from these simulations were kindly provided by Dr. Antonio Attili.

A fully premixed unburnt methane–air mixture at equivalence ratio $\phi = 0.7$ and bulk velocity $U_{bulk} = 100\,\mathrm{m\,s^{-1}}$ is injected through a rectangular slot of width $H = 1.2\,\mathrm{mm}$ for R2 and $H = 2.4\,\mathrm{mm}$ for R3. It is surrounded by a slower coflow of burnt gases at $U_{coflow} = 15\,\mathrm{m\,s^{-1}}$, which leads to the formation of two shear-driven turbulent flame fronts. The pressure and unburnt temperature are set to $P = 4\,\mathrm{bar}$, $T_u = 800\,\mathrm{K}$ to mimic typical engine conditions. Chemistry is described by a skeletal chemical mechanism containing 16 species and 72 reactions [207]. The laminar flame speed corresponding to these operating conditions is $s_L = 1\,\mathrm{m\,s^{-1}}$ and the thermal thickness is $\delta_L = 110\,\mu\mathrm{m}$. Turbulent fluctuations at the inlet for the bulk flow velocity are extracted from pre-computed turbulent channel flow simulations.

Both R2 and R3 are discretized on a uniform Cartesian mesh with a resolution $h = 20\,\mu\mathrm{m}$. The domain dimensions of R2 are 28.8 mm, 19.2 mm, and 5.16 mm in the streamwise (x), crosswise (y), and spanwise (z) directions, respectively. For R3, all three domain dimensions of the domain are doubled and all other parameters are kept constant. The Reynolds number $Re = U_{bulk}H/\nu$ is therefore doubled.

The Kolmogorov lengthscale η is computed by the authors from the averaged kinematic viscosity $\bar{\nu}$ and Favre-averaged dissipation $\widetilde{\epsilon}$ as $\eta = (\bar{\nu}^3/\widetilde{\epsilon})^{1/4}$. It satisfies the criterion $\eta \geq 2h$ which ensures that the finest turbulent structures are well resolved.

Table 5.1 summarizes all the relevant simulation parameters. In this chapter, unless otherwise specified, a temperature-based progress variable will be used. A slice of c progress variable field is shown in Figure 5.3 for sample snapshots of R2 and R3. Qualitatively, the same flame structures can be found in both flames. Starting from the slot and moving in the downstream direction, the flame fronts are increasingly wrinkled, eventually featuring rib-like oblique structures. At the end of the main body, instead of a well-defined flame tip, pockets of unburnt gases detach from the main body of the flame and finish burning as they are convected downstream.

	R2	R3
Fuel	CH_4	
ϕ	0.7	
T_u	800 K	
P	4 bar	
s_L	$1\,\mathrm{m\,s^{-1}}$	
δ_L	110 µm	
U_{bulk}	$100\,\mathrm{m\,s^{-1}}$	
h	20 µm	
H	1.2 mm	2.4 mm
Re	5600	11200
u'/s_L	10	9.8
l_t/δ_L	4.8	5.9
η	23 µm	25 µm
Ka	23	21

Table 5.1: R2 and R3 simulation parameters [209].

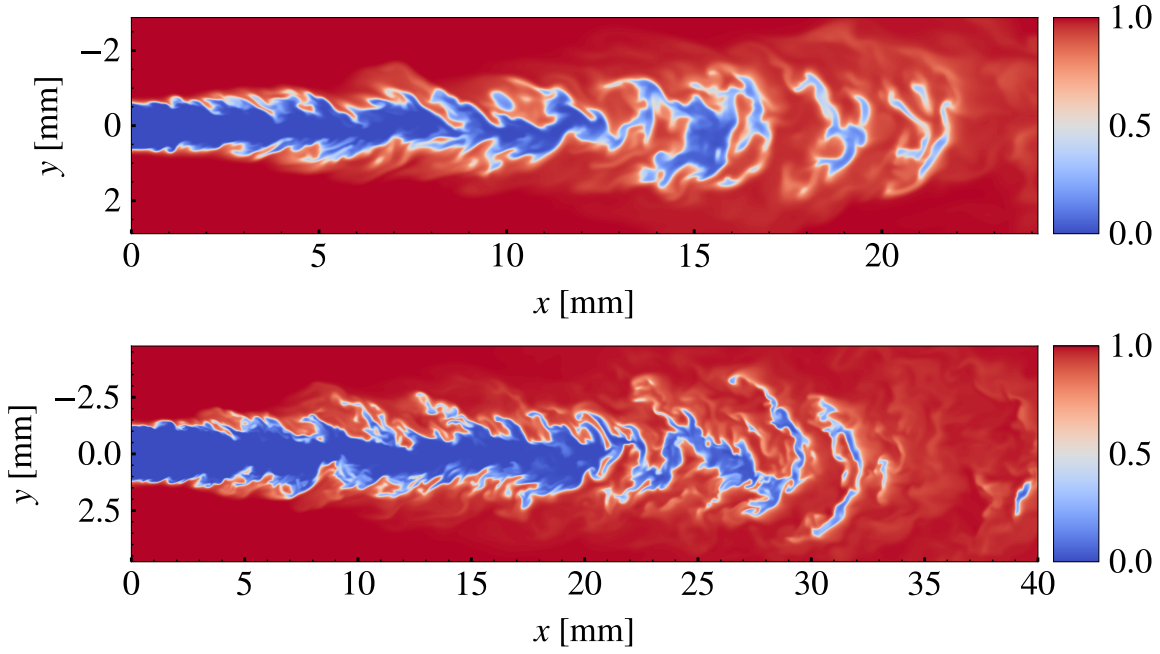


Figure 5.3: Progress variable slices of R2 (top) and R3 (bottom). Zones of the domain far from the flame in the spanwise direction were cropped for the visualization.

5.3 Data preparation

To generate the training and test datasets, DNS variables discretized on the uniform cartesian grid of R2 and R3 can be easily manipulated as structured arrays to facilitate data processing.

c and $|\nabla c|$ are filtered at a filter size $\Delta/\delta_L = 4.9$ ($\Delta/h = 27$) and downsampled by a factor $R = 4$. These values are obtained via Algorithm 2 and the parameters of the TFLES used to validate the model in Section 5.7. The output field of the CNN $\overline{\Sigma}^+$ is computed as in Equation 5.1 using the maximum FSD on the corresponding 1D flame for normalization.

Figure 5.4 shows c, $\bar{c}$ and $\overline{\Sigma}^+$ on a given slice. On the $\bar{c}$ slice, it is apparent that small-scale wrinkling has been lost through filtering. This information is contained in the $\overline{\Sigma}^+$ field that must be modeled by the CNN.

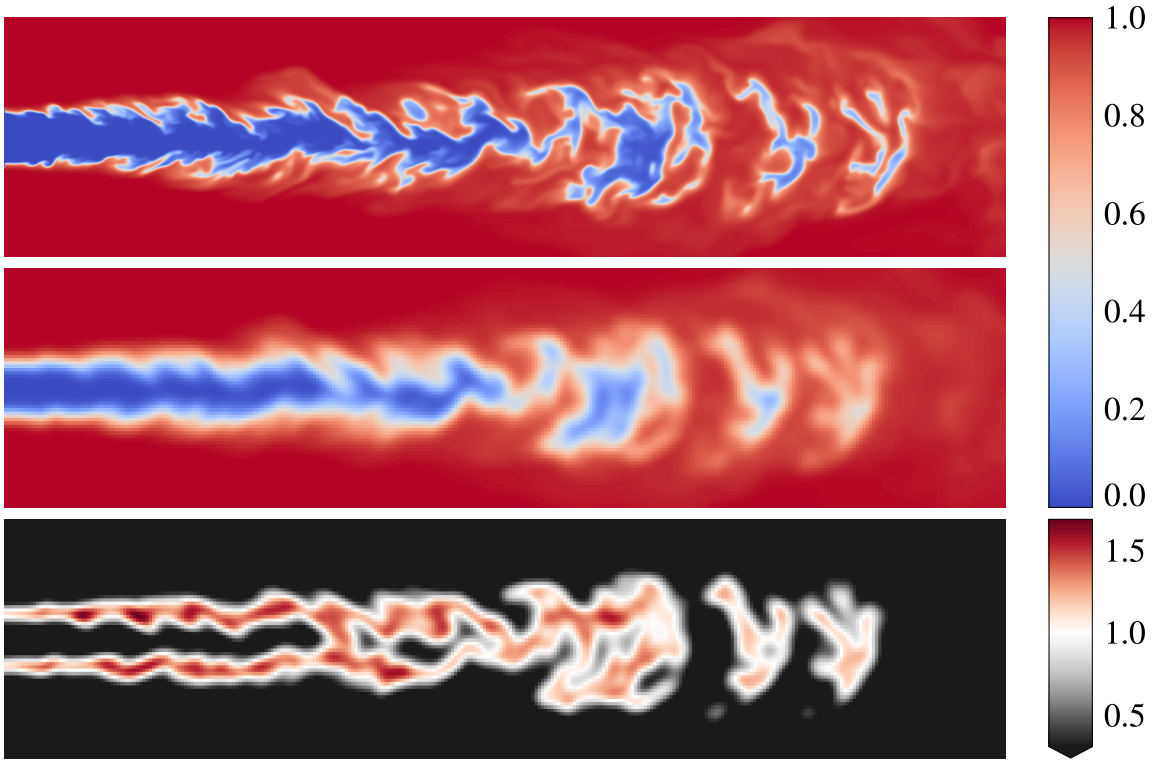


Figure 5.4: Result of filtering and downsampling an R2 snapshot. Top: DNS c slice. Middle: Filtered and downsampled $\bar{c}$ slice. Bottom: Filtered and downsampled $\overline{\Sigma}^+$ slice.

Training is performed on 90 snapshots of R2 sampled every 0.01 ms from the steady-state regime of the DNS ($t = 5.5$ ms to $t = 6.4$ ms). The first 80 constitute the training dataset, and the last 10 are kept as a validation set to monitor the training process. One steady-state R3 snapshot ($t = 1.8$ ms) will be used as a test set to assess how well the CNN can generalize to the R3 configuration.

Training	**Validation**	**Test**
First 80 snapshots of R2	Last 10 snapshots of R2	1 snapshot of R3
$80 \times 64 \times 80 \times 348$	$10 \times 64 \times 80 \times 348$	$1 \times 128 \times 160 \times 696$

Table 5.2: Composition and dimensionality of the datasets used for training and *a priori* validation.

5.4 Comparative statistics

Before building the model, a statistical analysis of the R2 and R3 flames is performed to understand the difficulty of learning a mapping $\bar{c} \mapsto \overline{\Sigma}^+$ that generalizes from R2 to R3. First, the shift in the relationship between the input $\bar{c}$ and the output $\overline{\Sigma}^+$ is investigated through their JPDF. It is visualized via kernel density estimation [133] which produces a continuous PDF $\hat{p}$ that approximately fits a PDF p given observed samples $(x_1, ..., x_n) \sim p$. At a new location x_0, the kernel density estimate (KDE) $\hat{p}$ is evaluated as

$$\hat{p}(x_0) = \frac{1}{n\lambda} \sum_{i=1}^{n} K_\lambda(x_0, x_i)\,, \tag{5.2}$$

where K_λ is a Gaussian kernel with a standard deviation λ. K_λ provides a smooth estimate of the number of samples x_i that are close to x_0.

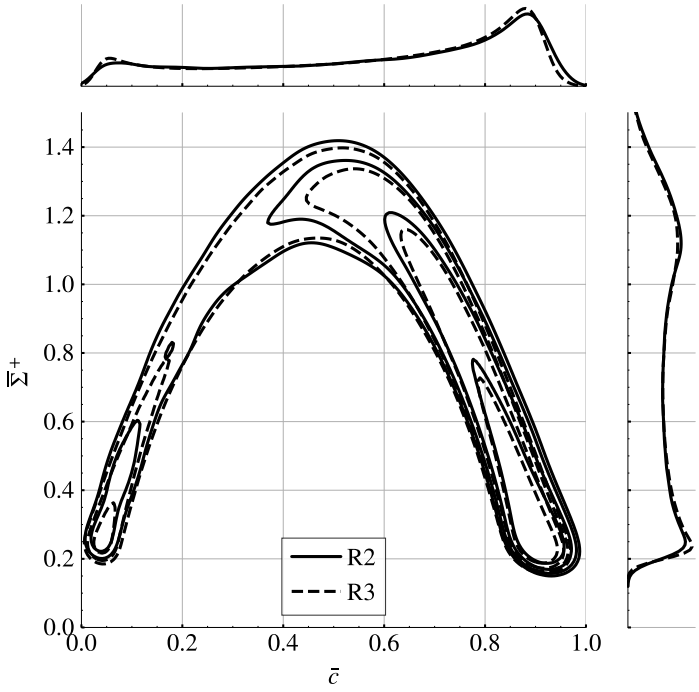


Figure 5.5: KDE plots of the joint PDF and marginal PDFs of $\bar{c}$ and $\overline{\Sigma}^+$ on R2 and R3.

Figure 5.5 shows kernel density estimates of the JPDF of $\bar{c}$ and $\overline{\Sigma}^+$ for R2 and R3. KDEs are visualized by five iso-density levels for clarity. The KDEs of the marginal PDFs are also plotted above (for $\bar{c}$) and on the right (for $\overline{\Sigma}^+$). Compared to visualization methods like histograms that strictly rely on sample binning, KDEs allow clear comparisons of univariate and bivariate data distributions. These plots only include locations where the condition $\overline{\Sigma}^+ > 0.2$ is met. This makes the PDFs easier to read by filtering out the large number of points at the tails of the PDFs that do not contain a meaningful amount of flame surface, typically in the fully unburnt or burnt mixtures, and which are therefore less critical to predict accurately.

The joint PDFs for R2 and R3 are found to be very similar despite the different Reynolds number. This could indicate that a model trained on R2 would be able to generalize to R3. This result can be surprising in light of the differences found by Attili et al. [15] in terms of total wrinkling and structure of the flame fronts. They find that the turbulent flame surface area normalized by a reference unwrinkled surface was found to grow from about 3.1 in R2 to 4.3 in R3. This reveals that the R3 flame is globally more wrinkled than R2. As a result, the total flame surface density contained in any box of size Δ should be higher for R3. The fact that the distribution of $\overline{\Sigma}^{+}$ does not reach higher values for R3 is therefore somewhat unexpected. One reason for this behavior could be that the filter size Δ is not large enough ($\Delta/\delta_L = 4.9$) compared to the typical size of the wrinkles. Larger filter sizes should highlight the differences between R2 and R3, up to the limit where averaging is performed on the whole domain where the area ratios of Attili et al. should be recovered. Another possible contributing factor is the thickening of the reaction layer with increasing Reynolds number which drives down the values of $|\nabla c|$ on the R3 DNS.

The joint PDF gives insights on the pointwise relationship between the input and output of the model. Since the CNN is sensitive to the geometry of the flame front, a complementary analysis is performed on how the geometry of the input varies between the two flames. This is quantified through the flame front curvature κ:

$$\kappa = \nabla \cdot \left(-\frac{\nabla c}{|\nabla c|} \right) . \tag{5.3}$$

The PDF of the curvature of an iso-surface $c = 0.7$ (corresponding to the temperature of maximum heat release rate in a laminar flame) normalized by the laminar flame thickness is plotted in Figure 5.6. It shows that the curvatures of the flame fronts are similarly distributed, which once again suggests that the task of generalizing from R2 to R3 is reasonably achievable.

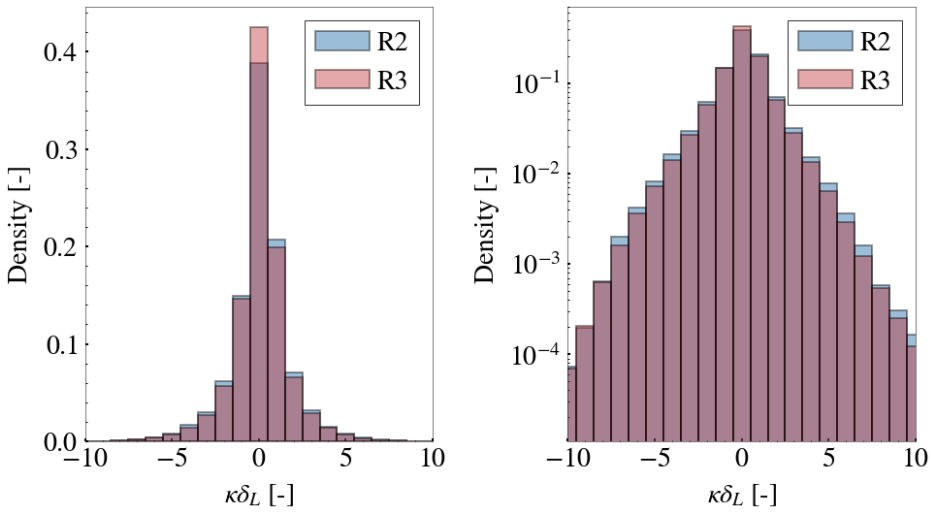


Figure 5.6: PDF of normalized curvature on the $c = 0.7$ isosurface for R2 and R3 in linear (left) and logarithmic scale (right).

5.5 Model training

5.5.1 Model architecture

The DL model is a U-Net with the architecture presented in Section 3.4. Variations on the activation layer, number of encoder–decoder stages, number of convolutional layers in each stage have been tested, with no significant improvement over this baseline. Since this architecture already reaches excellent performance (see Section 5.6), a thorough neural architecture search or a hyperparameter search have not been deemed necessary. Architectural improvements could be explored for a bigger, more diverse dataset that would require very high model capacity.

5.5.2 Data pipeline

Training the model is done through SGD, as presented in Section 3.1. SGD performs gradient descent steps on batches sampled from the training dataset. The construction of these batches is now detailed. The current dataset is composed of a small number of samples with a large number of degrees of freedom, which distinguishes it from many computer vision datasets. For instance, the ImageNet dataset [151] contains 14 million RGB images, which are commonly cropped to a $256 \times 256 \times 3$ size when training computer vision models. GPU memory constraints impose a tradeoff between the batch size and the number of degrees of freedom of batch elements. The low number of degrees of freedom of 2D, 3-channel images allows for large batch sizes and major training speedups [333, 405]. Conversely, for the R2 dataset, the input field of a single training snapshot contains approximately 1.8 million single-precision floating point values, and the entire dataset contains 79 training snapshots. Large batches built from full snapshots are therefore not achievable. Instead, a batch is formed by randomly selecting $n_s = 4$ training snapshots and extracting $n_c = 8$ random crops of size $d_c^3 = 40^3$ on each snapshot. This leads to a batch of $n_c \times n_s$ samples, as illustrated in Figure 5.7. As the random crops cover a small fraction of each snapshot, this strategy generates batches with high variability, which could be beneficial to help the convergence of SGD through noisier gradient updates.

Any point in the domain can be at the center of the random crop, except those located at a distance $d_c/2$ or smaller to the boundary. The filtered DNS can be padded to allow for random crops centered near the boundaries, as long as the padding is consistent with the boundary condition. This is helpful to slightly increase the amount of training data that can be sampled. In periodic directions, padding consists in mirroring information at the opposite boundary. Periodic padding of size $d_c/2$ is therefore performed in the spanwise direction.

Spatial data augmentation is applied to the random crops with random 90° rotations and reflections with respect to the 3 coordinate axes, adding variability to the orientation of the flame fronts in the dataset.

The validation set is neither random cropped nor spatially augmented. This is to ensure that the validation loss is consistently computed on the same data, as its value is used to select the best performing model over the training process.

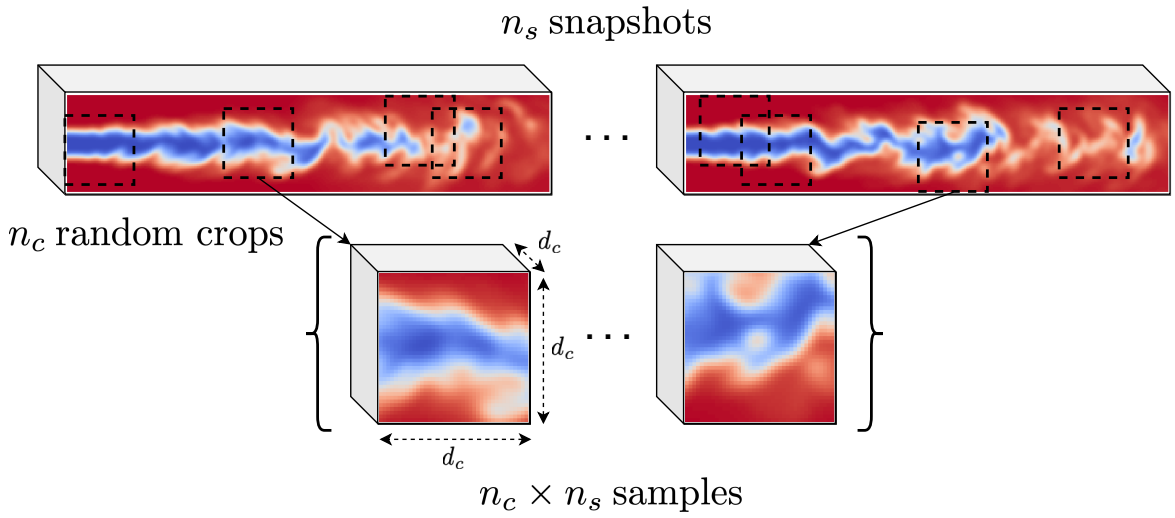


Figure 5.7: Building a training batch from randomly cropped samples among the training set.

Training is performed by stochastic gradient descent using the Adam optimizer [164] with an initial learning rate $\epsilon = 0.01$. The loss function is a voxelwise mean squared error (Section 3.1). The biases of all the convolutional filters are initialized to 0, and the filter weights are initialized according a uniform Glorot distribution [114]:

$$w \sim \mathcal{U}\left[-\sqrt{\frac{6}{n_{in}+n_{out}}}, \sqrt{\frac{6}{n_{in}+n_{out}}}\right], \tag{5.4}$$

where $\mathcal{U}$ is the uniform distribution, n_{in} and n_{out} are the number of input and output channels of the layer. Table 5.3 lists the hyperparameters used during training.

Figure 5.8 shows training and validation loss curves for a full training run, plotted against the cumulative number of training samples seen since the start. Training samples denote individual random crops of size d_c^3 which are grouped in batches of size $n_c \times n_s = 32$. One training epoch is arbitrarily defined as a set of 80 batches, or 2560 samples. After each epoch, the training loss is computed from the average loss value over the epoch batches, and the model is evaluated over the entire validation set to compute a validation loss value. The validation loss is consistently higher than the training loss, with no visible overfitting which would be denoted by a noticeable increase of their difference. Due to the stochastic nature of the optimization and sampling processes, the losses do not decrease monotonically. In particular, after about 1.9 million training samples, a sharp peak in the training loss is observed. This could be due one or many successive batch instances containing a list of random crops that are not representative of the total target distribution to be modeled. This risk was alleviated by purposefully cropping out domain areas containing only burnt gases above and below the flame in the spanwise direction. Although it may slow down convergence, this peak is not necessarily harmful for the training process, and the losses recover their original levels shortly after. In fact, temporary increases in the loss function are typical when using an SGD training technique called warm restarts [205]. Warm restarts are done by episodically increasing the value of learning rate and letting it decay using an

annealing schedule. They are used to escape suboptimal local minima that can stall training early on, and can often lead to better final performance. In the case of Figure 5.8, the training and validation losses both recover shortly after the peak.

n_c	n_s	d_c	ϵ	Batches per epoch	Epochs
8	4	40	0.01	80	1000

Table 5.3: Training hyperparameters.

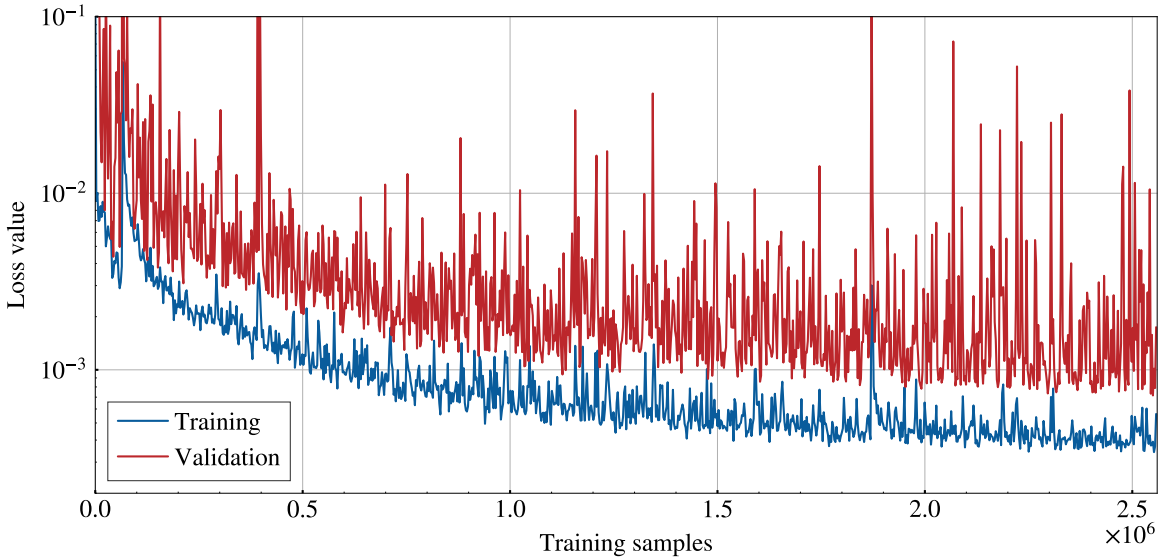


Figure 5.8: Training and validation loss curves.

Training is arbitrarily stopped after $\sim 2.6 \times 10^6$ samples, when adequate performance has been reached and training is starting to plateau. Hardware details and training performance metrics for this run are recorded in Table 5.4. GPU memory consumption is dictated by the size of the validation set which is fed into a single batch. Its value can be reduced by batching the validation set if limiting GPU memory is a concern. For the training steps only, memory consumption is 10 GB.

Hardware	NVIDIA A100 40 GB GPU
ML framework	Tensorflow 2.4.0
Wall clock time	4 h17 min
GPU memory consumption	34 GB

Table 5.4: Technical details for the training run of Figure 5.8.

5.6 *A priori* results on filtered DNS

5.6.1 Performance on the validation set

The best performing model is built from the model weights saved at the lowest validation loss attained during training, after around 2.55 million training samples. This CNN is first evaluated on the validation set. Since the entire dataset only includes snapshots from the steady-state regime of the R2 flame, validation snapshots are drawn from the same distribution as the training set. These results ensure that the model has learnt to adequately predict $\overline{\Sigma}^+$ on this distribution.

Figure 5.9 shows slices of the input field $\overline{c}$, target outputs $\overline{\Sigma}^+_{\text{true}}$, predicted outputs $\overline{\Sigma}^+_{\text{CNN}}$ and its deviation from the ground truth $\overline{\Sigma}^+_{\text{CNN}} - \overline{\Sigma}^+_{\text{true}}$ for one validation snapshot. The CNN qualitatively reproduces all the trends of the ground truth, despite significant variations all across the flame front. Deviations to the ground truth are limited in magnitude and localized in small areas.

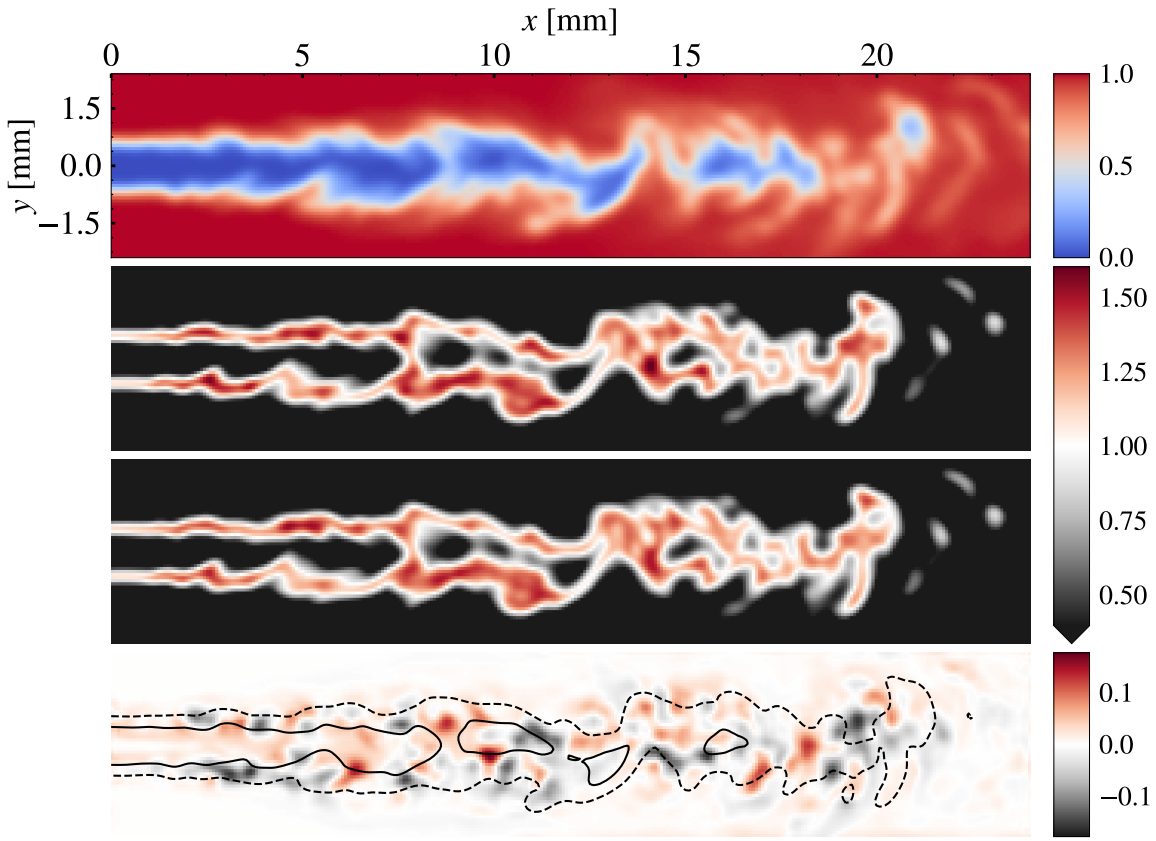


Figure 5.9: Sample slices from a validation snapshot of R2. Starting from the top: inputs $\overline{c}$, ground truth $\overline{\Sigma}^+_{\text{true}}$, CNN predictions $\overline{\Sigma}^+_{\text{CNN}}$, and deviation from the ground truth $\overline{\Sigma}^+_{\text{CNN}} - \overline{\Sigma}^+_{\text{true}}$. On the bottom slice, progress variable isolines (solid line: $\overline{c} = 0.2$, dashed line: $\overline{c} = 0.8$) denote the location of the flame front.

Figure 5.10 provides a quantitative pointwise comparison of model predictions to the ground truth on the entire validation set. The vast majority of the points are concentrated around the ideal model line $y = x$, and the conditional mean of model predictions fits this line almost perfectly. There are no outliers with highly unphysical values. This confirms the excellent accuracy of the model. Integrating the total FSD field over the whole domain produces the total flame surface area of the flame. This value is well predicted by the CNN with a relative error of less than a percent. All in all, this qualitative and quantitative analysis of the CNN predictions on the validation set proves its ability to accurately predict the distribution of $\overline{\Sigma}^+$ on R2.

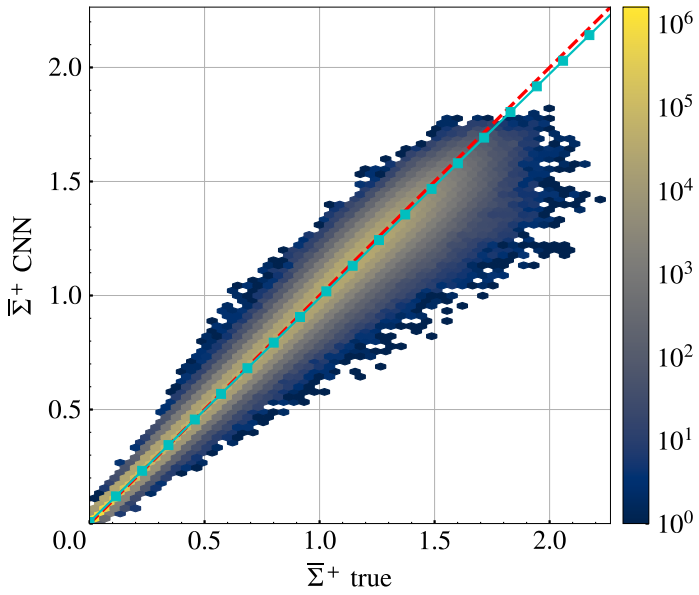


Figure 5.10: Hexbin plots of model predictions against the ground truth on the R2 validation set. Data points are gathered in hexagonal bins colored according to the number of data points they contain (logarithmic color scale). The red line $y = x$ represents a perfect model. The cyan line is the conditional mean of model predictions.

5.6.2 Performance on the generalization set

The CNN is now evaluated on a snapshot of R3. Figure 5.11 shows excellent qualitative results, much like on R2. Looking at the flame front more closely in Figure 5.12, it is evident that the model is able to correctly reproduce fine local variations of total FSD. The green circle highlights an example of a local flame element in the filtered DNS with sub-unity $\overline{\Sigma}^+$ values across the entire flame front. This is explained by a departure from a pure flamelet regime at the present high turbulence conditions (high Ka and Re), and by local turbulence-induced thickening of the flame front [15]. Note that $\overline{\Sigma}^+$ is not the SGS wrinkling factor, and instead quantifies the amount of local total flame surface relative to the peak value in a laminar 1D flame. It is therefore not constrained to only take values higher than unity in the flame front.

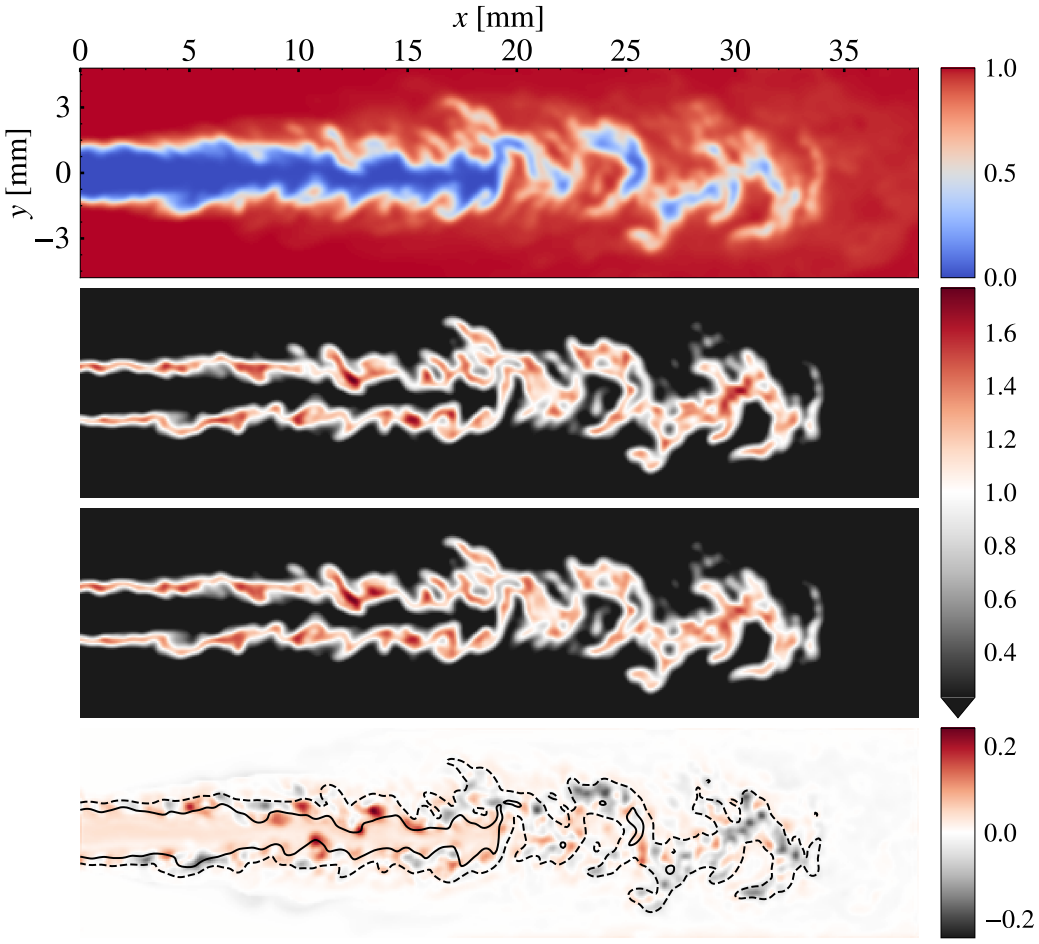


Figure 5.11: Sample slices from a snapshot of R3. Starting from the top: inputs $\overline{c}$, ground truth $\overline{\Sigma}^{+}_{\text{true}}$, CNN predictions $\overline{\Sigma}^{+}_{\text{CNN}}$, and deviation from the ground truth $\overline{\Sigma}^{+}_{\text{CNN}} - \overline{\Sigma}^{+}_{\text{true}}$. On the bottom slice, progress variable isolines (solid line: $\overline{c} = 0.2$, dashed line: $\overline{c} = 0.8$) are plotted.

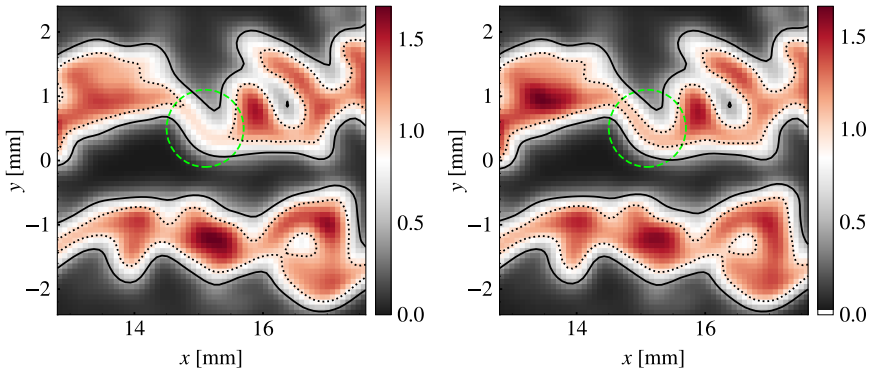


Figure 5.12: Zoom on ground truth (left) and CNN predictions (right) of $\overline{\Sigma}^{+}$, with isolines of $\overline{\Sigma}^{+} = 0.5$ (solid) and $\overline{\Sigma}^{+} = 1$ (dotted). A flame element of the filtered DNS with sub-unity maximum total FSD is circled in green.

Quantitative metrics confirm the excellent accuracy of the model. Pointwise predictions are compared to the ground truth in Figure 5.13, where a near-perfect linear trend with minimal variance is apparent.

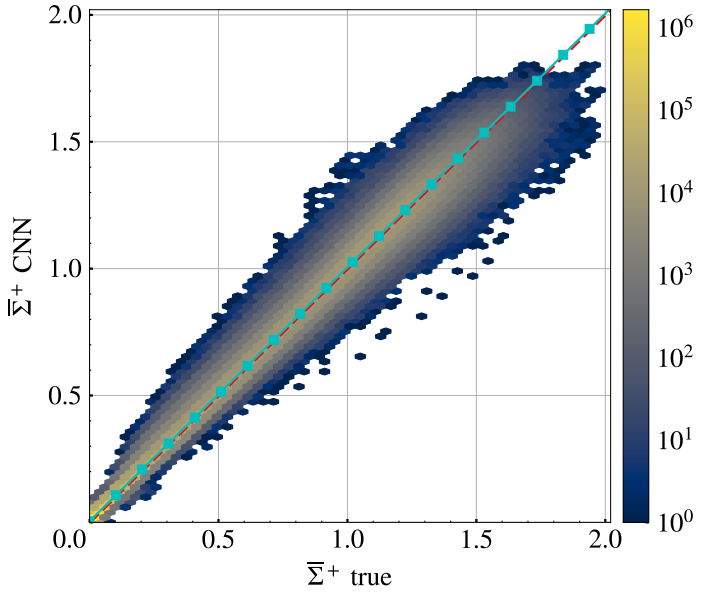


Figure 5.13: Hexbin plots of model predictions against the ground truth on R3.

The streamwise evolution of the flame surface is examined by integrating $\overline{\Sigma}^+$ on cross-section slices of the width of a coarse cell at every streamwise grid location:

$$A(x) = \int_{x^*=x-h/2}^{x+h/2} \sigma \overline{\Sigma}^+(x^*, y, z)\, \mathrm{d}x^*\, \mathrm{d}y\, \mathrm{d}z. \tag{5.5}$$

$A(x)$ is plotted in Figure 5.14 for the true total FSD (labeled *DNS*), the CNN predictions, the CDSM model (Equation 2.107 with $\hat{\Delta} = 1.5\Delta$, $\Delta_m = 2\hat{\Delta}$), and the resolved FSD (labeled *No model*). The CDSM model is implemented *a priori* and used as a reference algebraic model for the total FSD. The resolved FSD shows the outcome of not using any SGS flame surface model. The CNN is able to accurately reproduce the total flame surface area across the entire length of the flame. On the other hand, the CDSM model does not predict enough wrinkling near the inlet, and overpredicts near the flame tip. Close to the inlet, the flame is mostly flat, and small wrinkled structures are lost in the filtered flame front, which would explain the underprediction of the CDSM model. The flame tip region features prominent flame-flame interactions as pockets of unburnt gases are detached, and high curvatures as these pockets are small. These two factors are weaknesses of the dynamic Charlette model which the Mouriaux corrections attempt to fix, but they may not be sufficient in this specific configuration.

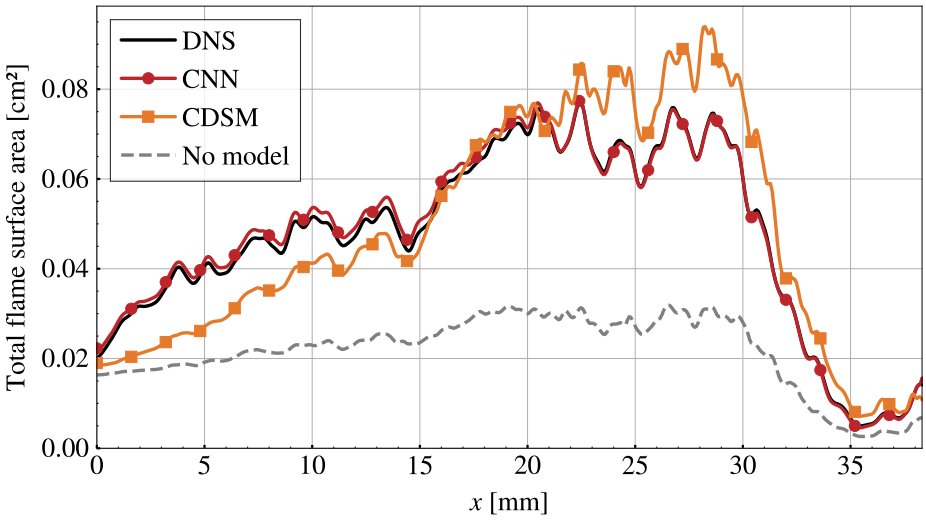


Figure 5.14: Evolution of the total flame surface area along the streamwise direction on the R3 snapshot.

This ability of the CNN to generalize from R2 to R3 is consistent with results from Attili et al. [16]. They train a version of this U-Net on the R1/R2/R3/R4 flames and obtain very good generalization results among the higher Reynolds number flames (R2 and above). They conjecture that this is linked to the asymptotic behavior of high Reynolds number turbulence. In addition, when training the CNN on a specific region of the flame (flame base, fully turbulent region, or flame tip), it performs noticeably worse when tested on a different region. Their results highlight the CNN's sensitivity to the specific wrinkling distribution of its training set.

5.6.3 Robustness to data transformations

Given the propensity of deep neural networks to overfit a narrow training distribution, it is important to understand whether the excellent *a priori* performance of the CNN is contingent on specific processing choices in the generation of the training and test datasets. Variations on the progress variable formulation and on the nature of the filter are introduced to evaluate the CNN's robustness to small data transformations that preserve the physical formulation of the problem. These transformations slightly perturb the joint distribution of the CNN inputs and outputs, and robustness to these transformations would therefore indicate that the model has not completely overfit the specific training distribution.

5.6.3.1 Filter type

First, robustness to the filter type is investigated by generating $\overline{c}$ and $\overline{\Sigma}^+$ fields using Favre-averaged Gaussian filtering and box filtering on an R2 validation snapshot with the same filter size Δ as in the original Gaussian filtered dataset. Changing the nature of the filter affects the relationship between $\overline{c}$ and $\overline{\Sigma}^+$. This is visualized in Figure 5.15 and Figure 5.16 which compare the KDE of the JPDF of $\overline{c}$ and $\overline{\Sigma}^+$ for Gaussian filtered quantities and

Favre-averaged Gaussian filtered or box filtered quantities.

From Figure 5.15, it appears that Favre-averaging shifts the $\overline{c}$ distribution towards lower values of $\overline{c}$, as density is higher near the unburnt mixture. The normalized total FSD is unchanged, as $\overline{\Sigma}^{+}$ does not involve Favre-averaging (Equation 5.1). Figure 5.16 shows that compared to Gaussian filtering, box filtering slightly lowers the top end of the $\overline{\Sigma}^{+}$ distribution. Favre Gaussian filtering alters the input distribution of the model, while box filtering modifies the output distribution. Note that σ depends on the nature of the filter and was adjusted accordingly in each case.

The CNN is then evaluated on Favre-averaged Gaussian filtered and box filtered snapshots of R2. Figure 5.17 compares its predictions to the ground truth (filtered with the appropriate filter). Both scatter plots contain more variance than the one for the Gaussian filter (Figure 5.10). Nonetheless, most predictions are distributed close to the ideal model. In both cases, the streamwise evolution of the flame surface is very well predicted by the CNN (Figure 5.18). These results indicate that a CNN trained on a Gaussian filtered dataset is equally accurate for different filter types, which may extend to thickened flames *a posteriori*.

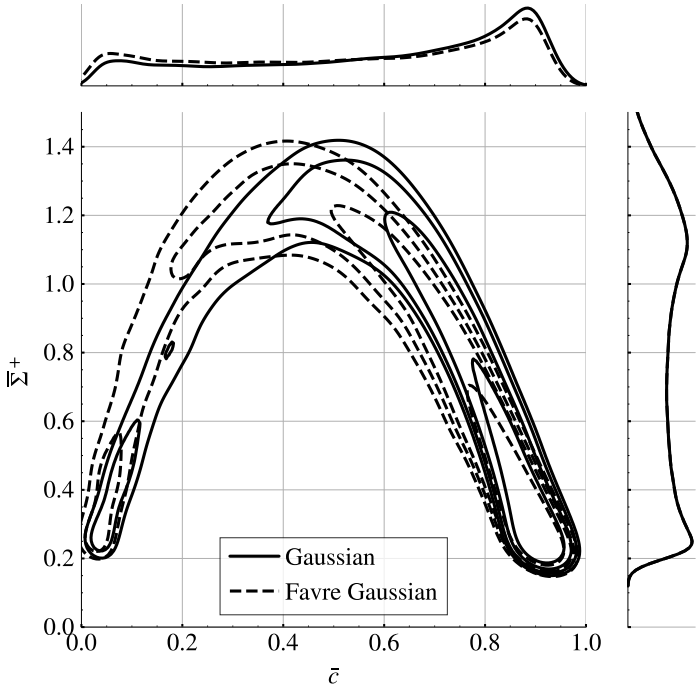


Figure 5.15: KDE plots of the joint PDF and marginal PDFs of Gaussian filtered and Gaussian Favre filtered $\overline{c}$ and $\overline{\Sigma}^{+}$.

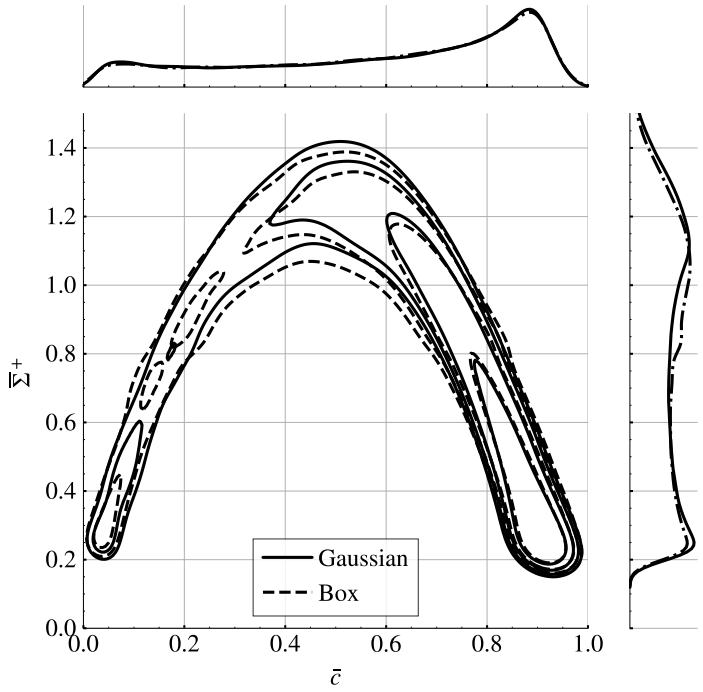


Figure 5.16: KDE plots of the joint PDF and marginal PDFs of Gaussian filtered and box filtered $\bar{c}$ and $\overline{\Sigma}^+$.

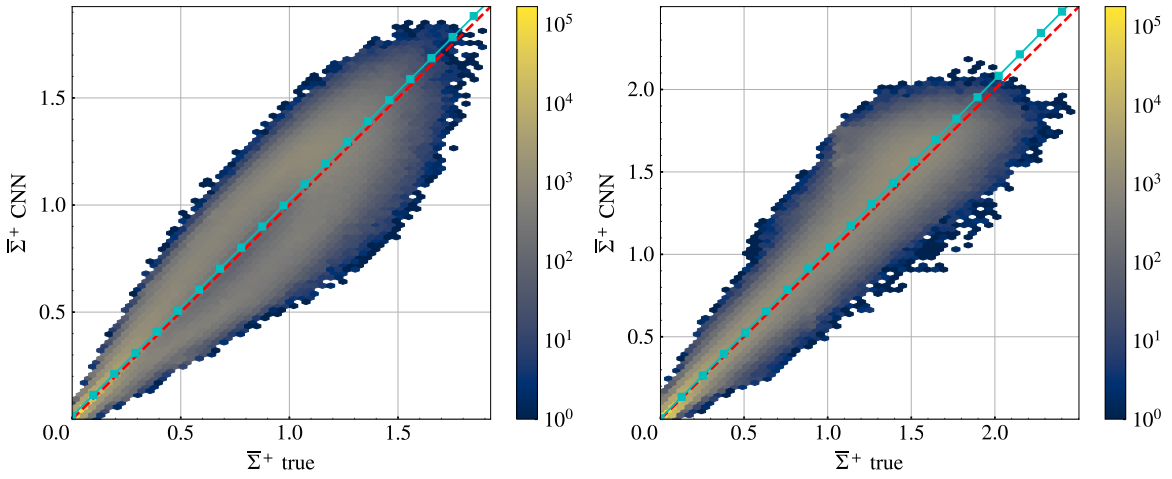


Figure 5.17: Hexbin plots of model predictions on Favre Gaussian filtered (left) and box filtered (right) variables against the ground truth on R2.

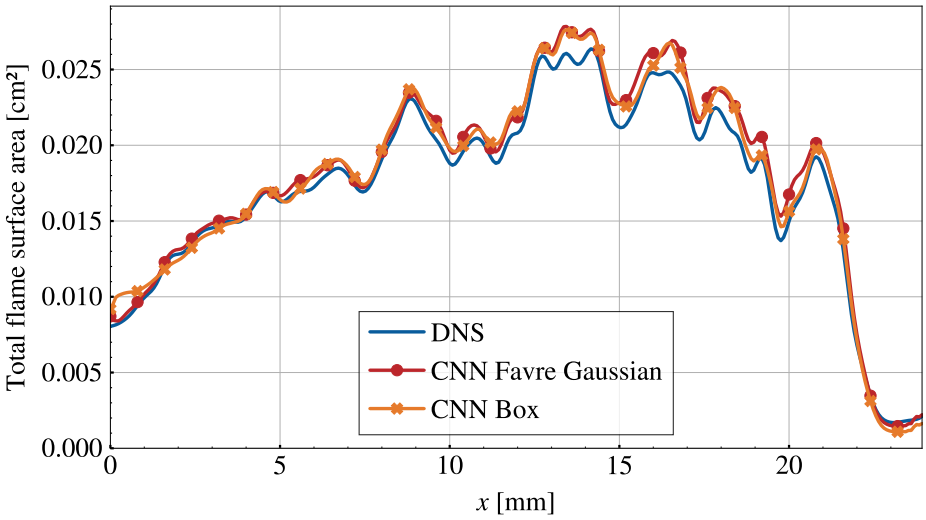


Figure 5.18: Evolution of the total flame surface area along the streamwise direction on a validation snapshot of R2 for different filter types.

5.6.3.2 Progress variable formulation

Next, robustness to the progress variable choice is assessed. A progress variable can be defined from any combination of reactive scalars that varies monotonically from the unburnt to the burnt state. Ideally, it should contain non-trivial gradients in the reaction zone to be relevant for reaction rate closures based on a flame surface formalism.

The progress variable in the original dataset was defined as the normalized temperature, and is noted c_T. Alternatively, a progress variable can be defined from the mass fraction of a reactant like the fuel:

$$c_{\mathrm{CH_4}} = 1 - \frac{Y_{\mathrm{CH_4}}}{Y^u_{\mathrm{CH_4}}} \tag{5.6}$$

or from the mass fraction of a product like O_2:

$$c_{\mathrm{O_2}} = \frac{Y_{\mathrm{O_2}} - Y^u_{\mathrm{O_2}}}{Y^b_{\mathrm{O_2}} - Y^u_{\mathrm{O_2}}}\,. \tag{5.7}$$

Figure 5.19 compares the laminar profiles of the three progress variable formulations in a 1D flame corresponding to the R2/R3 operating conditions. The fuel consumption rate and heat release rate normalized to the $[0, 1]$ range are also shown for reference. The c_T profile is the slowest to reach unity in the burnt mixture and has large gradients far into the postflame zone, which is typical of detailed mechanisms of hydrocarbon fuels. Using c_T could lead to a model predicting high total FSD levels in the burnt mixture, but at worst these values will have no effect on the flame propagation speed since the fuel consumption rate and heat release rate are so low in these regions. $c_{\mathrm{CH_4}}$ reaches unity right after the peak heat release rate, which may discourage its use for flame surface density models using this mechanism. $c_{\mathrm{O_2}}$ exhibits an intermediate behavior.

Figure 5.20 and 5.21 show that the JPDFs based on species mass fractions are shifted

towards higher values of $\bar{c}$ and $\overline{\Sigma}^{+}$. As a result, CNN predictions tend to be slightly lower than the ground truth (Figures 5.22 and 5.23). This can be interpreted as the CNN reproducing the range of $\overline{\Sigma}^{+}$ values in its training dataset. Nonetheless, the deviation from the ground truth remains small, and practical implications are expected to be limited as the formulation of the progress variable can easily be chosen to match how the training dataset was generated.

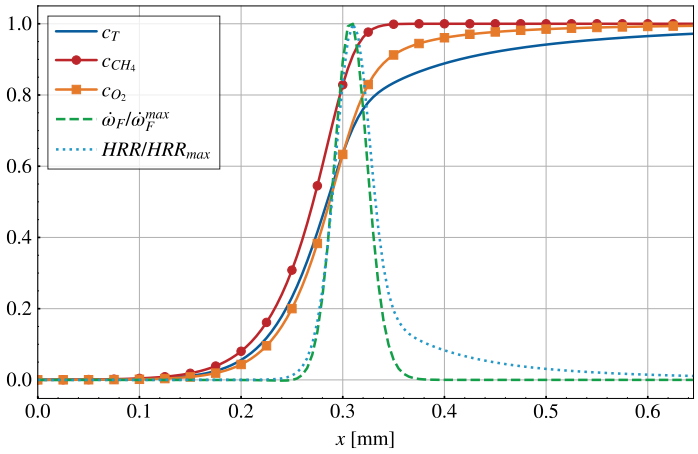


Figure 5.19: 1D laminar flame profile of the R2 flame with the three progress variable formulations, the normalized fuel consumption rate, and the normalized heat release rate.

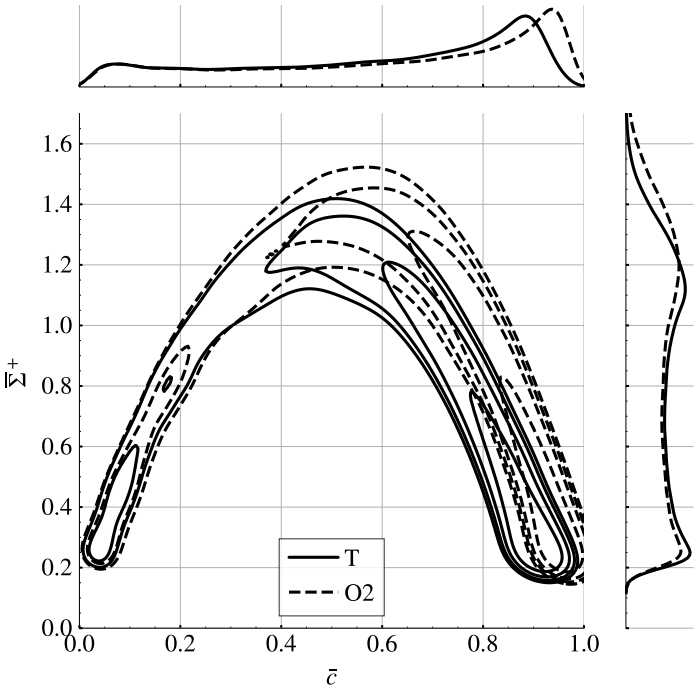


Figure 5.20: KDE plots of the joint PDF and marginal PDFs of $\bar{c}$ and $\overline{\Sigma}^{+}$ based on T and Y_{O_2}.

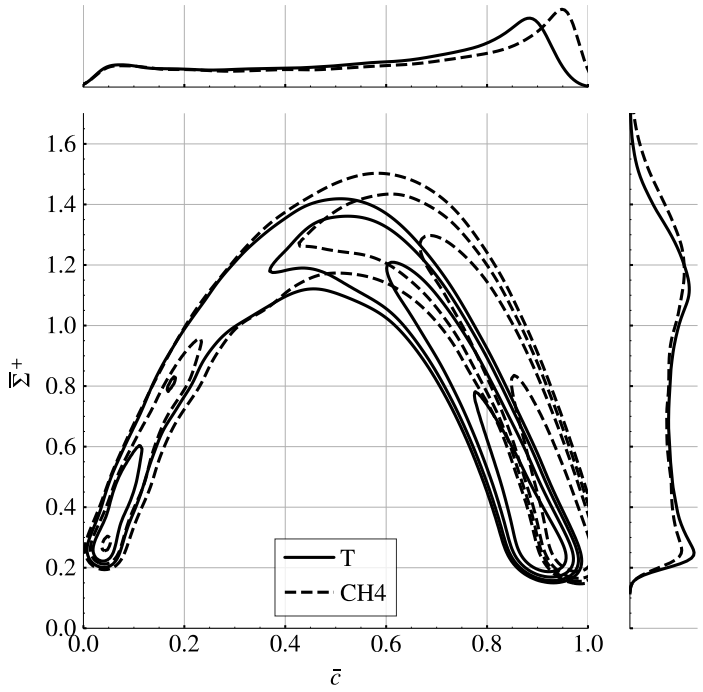


Figure 5.21: KDE plots of the joint PDF and marginal PDFs of $\bar{c}$ and $\overline{\Sigma}^+$ based on T and $Y_{\mathrm{CH_4}}$.

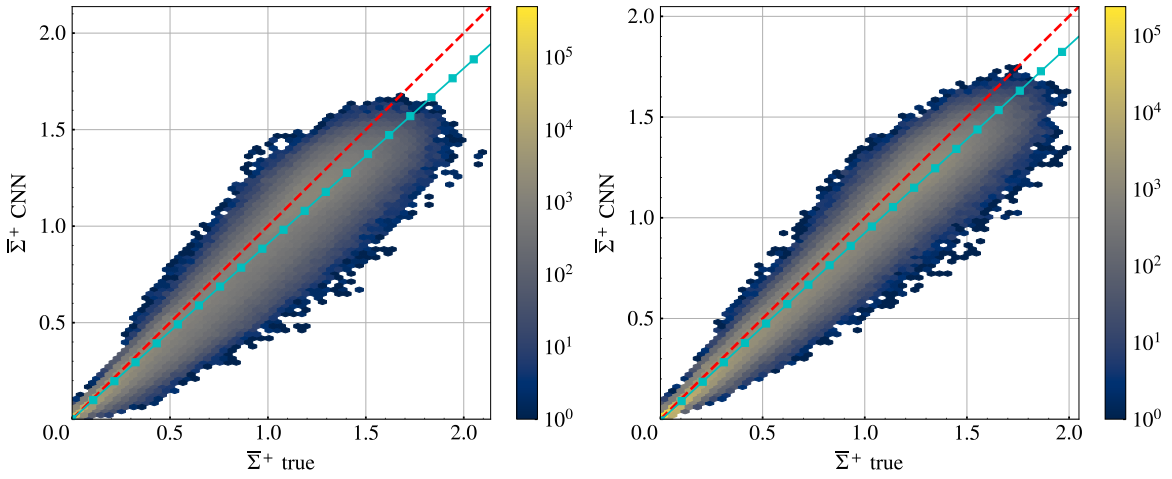


Figure 5.22: Hexbin plots of model predictions based on $c_{\mathrm{CH_4}}$ (left) and $c_{\mathrm{O_2}}$ (right) variables against the ground truth on R2.

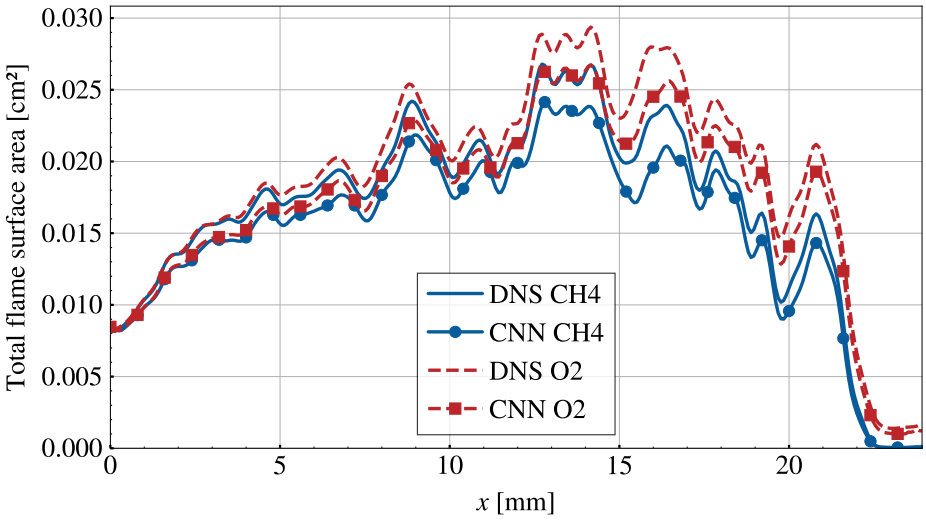


Figure 5.23: Evolution of the total flame surface area along the streamwise direction on a validation snapshot of R2 for different progress variable formulations.

5.7 *A priori* results on LES snapshots

5.7.1 Presentation of the LES

The performance of the CNN is evaluated on snapshots of an LES of R3. This is a hybrid between *a priori* and *a posteriori* testing, as the data comes from a true LES and not a filtered DNS, but model errors are not propagated through time as only isolated snapshots are considered.

A TFLES simulation of R3 is performed under the same physical conditions as the original DNS. The domain is discretized on a fully unstructured mesh of 95 million tetrahedral cells. Figure 5.24 shows the distribution of cell sizes near the inlet lips. The mesh is discretized to the DNS resolution $h_{\mathrm{DNS}} = 20\,\mu\mathrm{m}$ near the inlet walls to fully capture the near-wall turbulence and the initial development of the turbulent mixing layer aft of the lips. The mesh is gradually coarsened in the downstream direction until it reaches a constant mesh size $h = 4h_{\mathrm{DNS}} = 80\,\mu\mathrm{m}$. Based on the location of the flame in the R3 DNS, this constant resolution region extends in the crosswise direction up to a distance $2H$ away from the centerplane $y = 0$. Beyond that, the mesh is coarsened as there are no flame fronts that would need to be well resolved.

The original DNS used an auxiliary turbulent channel flow computation to inject inlet turbulence. In this LES, turbulent velocity fluctuations are instead added to the bulk flow using a synthetic generation method [174]. A turbulent velocity fluctuation field $\boldsymbol{u'}$ is built from a Fourier series decomposition

$$\boldsymbol{u'}(\boldsymbol{r},t) = \sum_{n=1}^{N} \boldsymbol{v}^n(\boldsymbol{\kappa}^n)\cos(\boldsymbol{\kappa}^n{\cdot}\boldsymbol{r} + \omega^n t) + \boldsymbol{w}^n(\boldsymbol{\kappa}^n)\sin(\boldsymbol{\kappa}^n{\cdot}\boldsymbol{r} + \omega^n t)\,, \tag{5.8}$$

with $N = 20$ modes. The Fourier modes $\{\boldsymbol{v}^n, \boldsymbol{w}^n\}$, wavevectors $\boldsymbol{\kappa}^n$, and pulsations ω^n are

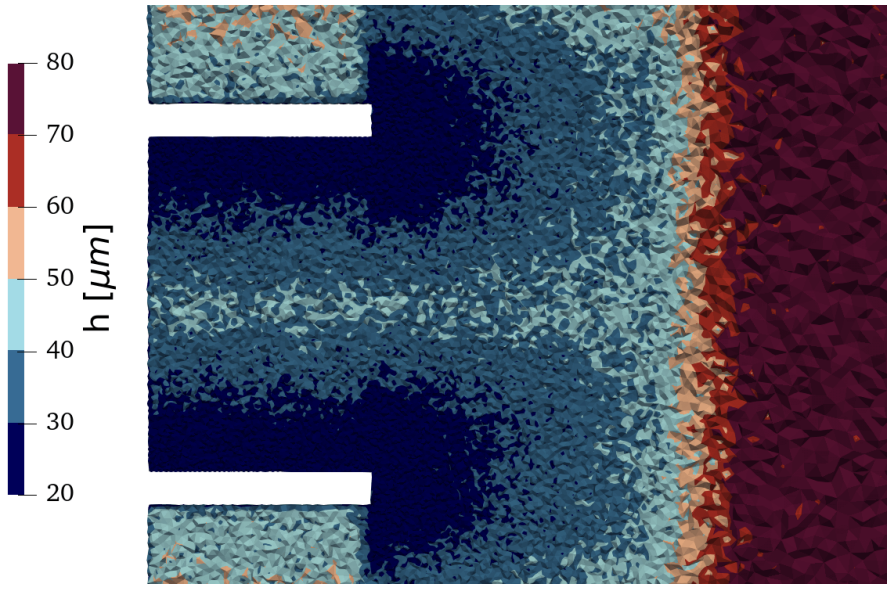


Figure 5.24: Cell size distribution of the LES mesh near the R3 slot.

random variables sampled to obtain a Passot-Pouquet [260] turbulence spectrum

$$\mathcal{E}(\kappa) = 16\frac{u'^2}{\kappa_e}\sqrt{\frac{2}{\pi}}\left(\frac{\kappa}{\kappa_e}\right)^4 \exp\left[-2\left(\frac{\kappa}{\kappa_e}\right)^2\right], \tag{5.9}$$

where $u' = 10\,\mathrm{m\,s^{-1}}$ is the turbulent fluctuation intensity and κ_e is the wavenumber associated with the most energetic lengthscale. κ_e is related to the integral lengthscale l_t of the spectrum through $l_t = \sqrt{2\pi}/k_e$, and its value is chosen to obtain $l_t = 0.6\,\mathrm{mm} = H/4$.

This LES was performed using the AVBP code whose specificities are presented in Section 2.8. NSCBC inlet and outlet boundary conditions [277], adiabatic no-slip walls, and periodic conditions are imposed in the streamwise, crosswise, and spanwise directions, respectively. The same skeletal mechanism as the original DNS is used for the chemistry. SGS turbulent stresses are modeled using the Sigma model (Equation 2.110). SGS combustion modeling is handled by the thickened flame model with a target thickening value $F = 4$, which would resolve the flame front on approximately 5.5 cells in the constant $h = 80\,\mu\mathrm{m}$ region. Thickening is only applied inside a flame sensor that dynamically tracks the position of the flame front [150]. Finally, the efficiency function is computed by the CDSM model updated every 1000 iterations, $\hat{\Delta} = 1.8\Delta$, $\Delta_m = 2.2\Delta$.

The filter size $\Delta/h = 27$ used to generate the training dataset was adapted to satisfy Proposition 4.2 given the thickening factor $F = 4$ used in this LES. Figure 5.25 compares the 1D filtered and thickened progress variable profiles. The profiles are significantly different above $c = 0.8$, as thickening preserves the slow rise to unity which is smoothed out by filtering.

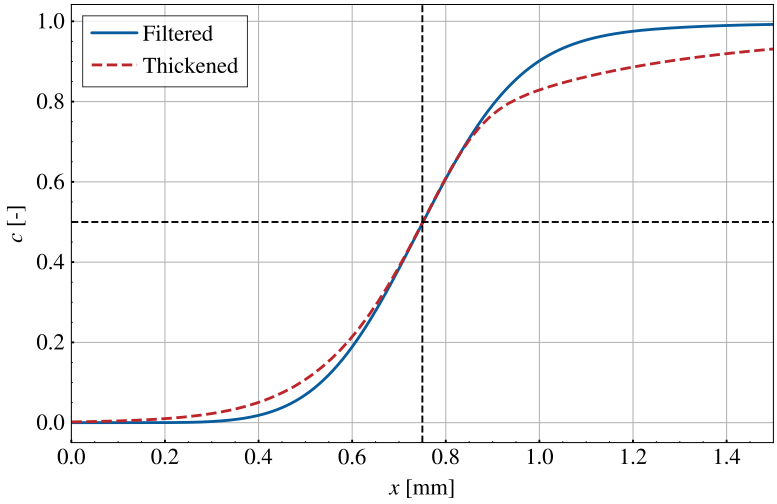


Figure 5.25: Profiles of filtered ($\Delta/h = 27$) and thickened ($F = 4$) c on the R2 1D laminar flame.

5.7.2 CNN results

An LES snapshot from the steady-state regime of R3 is used to compare the CDSM model used in the LES with the CNN. The Charlette model directly provides the wrinkling factor Ξ which is multiplied by $|\nabla\bar{c}|/\sigma$ to obtain $\overline{\Sigma}^+$.

Figure 5.26 shows visible differences in the behavior of the two models. The CNN predicts high values of $\overline{\Sigma}^+$ throughout the flame front. There are many extremely high values for the Charlette model, which seems to be caused by low $|\nabla\bar{c}|$ values in regions where strong turbulent mixing broadens the temperature field. In the KDE plots of Figure 5.27, the CNN is seen to predict $\overline{\Sigma}^+$ with a smaller spread. The CNN predicts relatively more high values, which are distributed towards lower $\bar{c}$ levels.

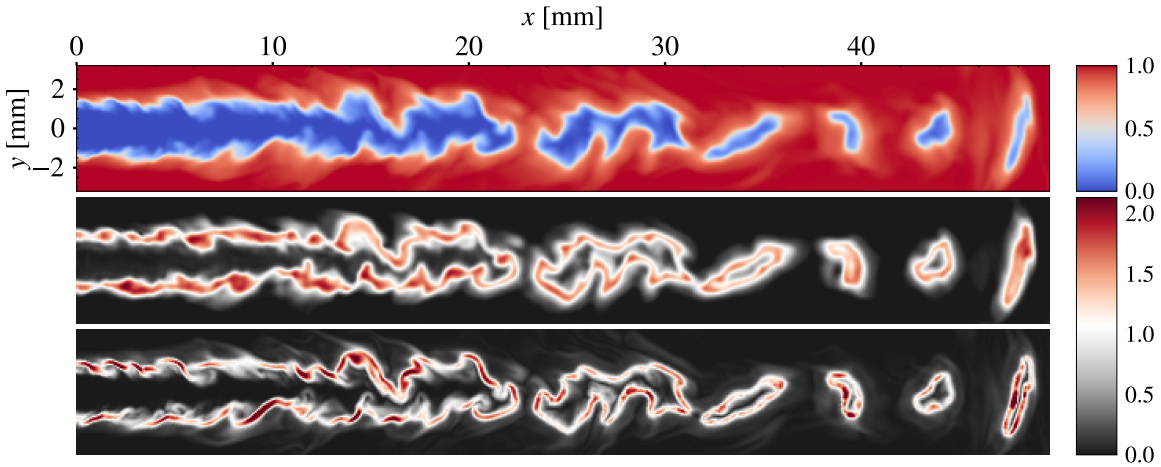


Figure 5.26: Slices of progress variable field (top) and $\overline{\Sigma}^+$ fields predicted by the CNN (middle) and the CDSM model (bottom) on a snapshot of the R3 LES.

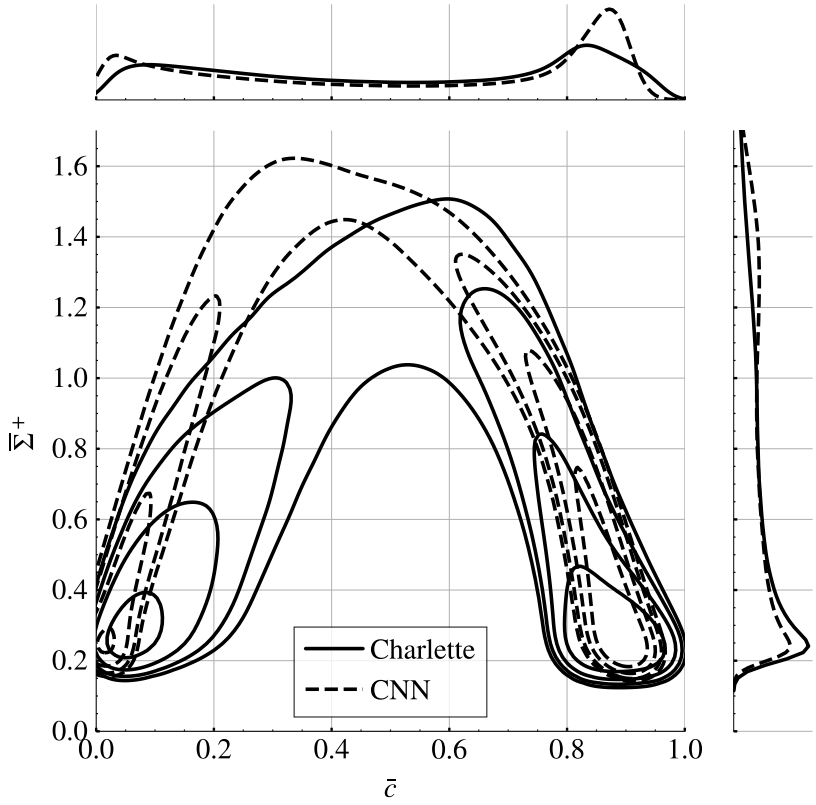


Figure 5.27: KDE plots of the joint PDF and marginal PDFs of $\bar{c}$ and $\overline{\Sigma}^+$ predicted by the CNN and Charlette model.

To assess the pertinence of the LES, the length of the flame is identified as the integral quantity of interest that must be well predicted by this simulation. The LES length is therefore compared to the DNS length. This length is estimated by computing the time- and transverse-averaged progress variable field, then measuring the point where the domain centerline meets the isosurface $c = 0.73$ of this averaged field. $c = 0.73$ corresponds to the progress variable value at peak methane consumption rate on a 1D laminar flame. The length of the LES flame is measured as 3.9 cm, while the DNS length is 2.8 cm, which is 30% lower. This implies that the CDSM model is not able to predict the right flame length due to insufficiently high wrinkling values.

Naturally, one might wonder how the CNN would affect these results. Although *a posteriori* simulations are left to Chapter 8, it is possible to compute the reaction rate field that would have resulted from using the CNN instead of the CDSM model by using the wrinkling fields. This assessment is not as rigorous as a true *a posteriori* evaluation since it is applied to frozen snapshots, but should still indicate what would happen if the CDSM model were replaced with the CNN in the simulation.

The domain is decomposed in 10 slabs that equally divide the length of the domain L along the downstream direction. In each of these slabs, the total fuel consumption rate is integrated for the CDSM and the CNN models and shown in Figure 5.28. It appears that overall, the CNN would tend to correct the overprediction of the flame length by inducing

higher burning rates in the middle and end sections of the flame. The total fuel consumption rate integrated in the whole domain is 8% higher for the CNN compared to the CDSM model. These encouraging results suggest that the CNN could improve the accuracy of the R3 LES if it were used *a posteriori*.

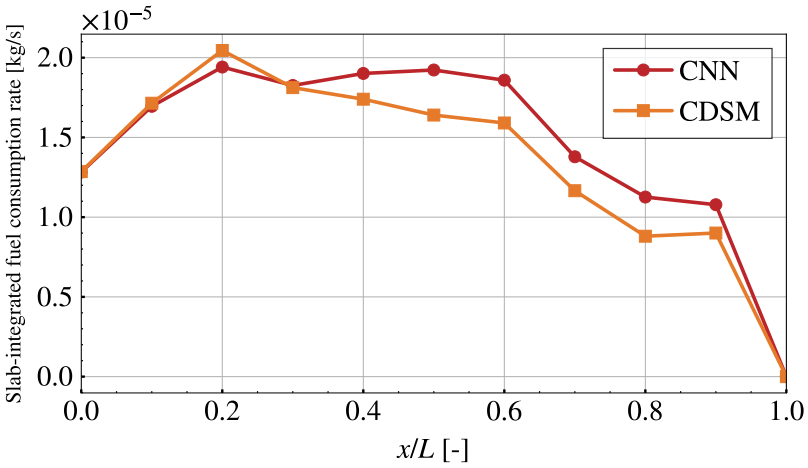


Figure 5.28: Fuel consumption rate integrated on 10 slabs along the streamwise direction in the R3 LES.

5.8 Model interpretability

This section is devoted to visualizing and analyzing how the CNN is trained and how it performs its predictions. Deep neural networks are notoriously difficult to interpret, and a large body of the literature has specifically focused on visualizing the inner workings of CNNs [190, 213, 255, 330, 340, 408]. For classifiers, common techniques include generating saliency maps representing the influence of each pixel of a given input image on its class prediction [340, 408], and feature visualization [255] which seeks inputs that maximize the response of a given neuron, layer, or class logit. Because the U-Net discussed in this chapter performs regression instead of classification, many of these methods are not applicable. This short study will be limited to two simple questions:

- What does the U-Net see? What region of the input can affect the output at a given location?
- What does the U-Net learn? Are there any redundant feature maps? How can we visualize the training process?

5.8.1 What does the U-Net see?

To understand what region of the input matters for the prediction of the U-Net, the extent of its receptive field (Section 3.3) is investigated. The size of its theoretical receptive field could be derived from the number of convolutional and pooling layers that it contains. In practice however, due to the distribution of the hidden layer connections inside the network, points

located at the center of the receptive field contribute more to the prediction than those at the periphery. To account for this effect, the notion of *effective* receptive field (ERF) [213] is introduced. It measures the extent of the receptive field that is actually meaningful to the prediction, and is measured by counting the number of connections originating from each input location. In practice, the ERF is computed from a U-Net where all the biases are initialized to 0, and all the convolution weights are set to 1. A gradient signal $\partial L/\partial y$ is set to 1 at the center of the output layer and 0 everywhere else. Backpropagating this signal through the CNN leads to a $\partial y/\partial x$ input signal representing the contribution of each input location to the center of the output layer. This input signal is approximately Gaussian distributed. The ERF is defined as the group input points with a magnitude greater than $1 - 95.45\% = 4.55\%$ of the center value, *i.e.* less than two standard deviations away. With n_{ERF} the cardinality of this group, the size of the ERF is defined as $\sqrt[d]{n_{ERF}}$ with d the dimensionality of the inputs ($d = 2$ for images, 3 for 3D fields).

Figure 5.29 compares the extent of the ERF of the U-Net with the filtered R2 flame. The size of the ERF is approximately equal to $4\Delta \approx 4\bar{\delta}_L$, which indicates that the CNN can learn from large-scale structures of the flame front. In comparison, the context size of the Charlette dynamic model can be estimated as the averaging filter size which is typically $2 - 6$ times the filtered laminar flame thickness [371, 377]. Both models can thus theoretically incorporate multi-scale information at similar context sizes. However, increasing the context size of the dynamic model may lead to numerical issues caused by flame/boundary and flame/flame interactions [239] and greatly impacts the computational cost of the procedure [377], whereas for CNNs it can simply be achieved by using a deeper network. Moreover, the CNN incorporates information from a continuum of scales ranging from the mesh resolution to the size of the ERF, while the dynamic model involves a set of discrete scales (the resolved, test-filtered, and averaging-filter scales).

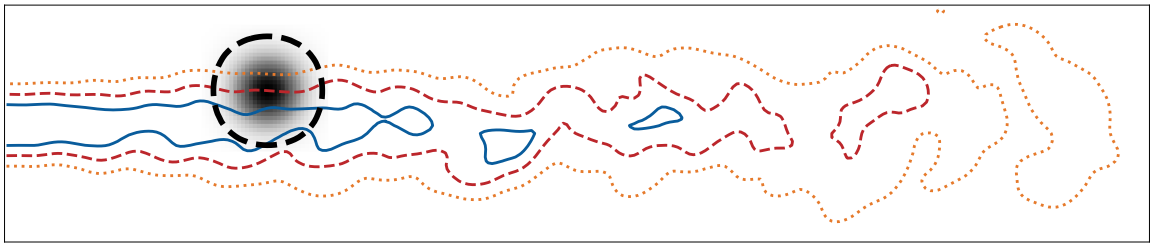


Figure 5.29: ERF superimposed on iso-lines of $\bar{c} = 0.1, 0.5, 0.9$ (solid blue, dashed red, dotted yellow) on a slice of filtered R2. Grayscale intensity in the ERF is proportional to the impact of the input voxel location on the output prediction at the center of the ERF. Dashed circular line: edge of the ERF.

The large size of the U-Net receptive field is a natural consequence of the successive convolution and pooling layers that it contains. This property differentiates it from shallow CNNs which have often been used for SGS modeling in numerical combustion [250, 305, 329, 336, 382]. These CNNs operate on a grid of first-degree neighbors, typically of size 3^3, to include limited spatial context (Table 4.1). Their receptive field is limited to this grid, and they should not contain many convolutional layers nor any pooling layer. These networks

have the benefit of being simple and lightweight at the cost of model capacity, and sacrifice sampling efficiency as they must be applied to each output location separately.

5.8.2 What does the U-Net learn?

Analyzing the evolution of the convolutional filters learnt by the U-Net can provide insights into how training progresses beyond simply monitoring the loss values. The distributions of the biases and kernel weights at specific convolutional layers are recorded every 100 epoch. Results for the first convolutional layer are shown in Figure 5.30 and Figure 5.31 in the form of KDE plots. Biases are initially set to 0 and quickly shift to a bimodal distribution that stabilizes early in the training process. Weights gradually spread over a large range of values centered around zero as training progresses. Interestingly, the biases of all the subsequent layers remain very close to their initial value of 0. Their weights do evolve during the training, as shown in Figure 5.32 which corresponds to the first convolutional layer of the bottom stage of the U-Net, where spatial dimensions are the smallest and the number of channels is maximal (Figure 3.4). The weights of the final convolutional layer seem to take the longest to stabilize (Figure 5.33) as noticeable modal shifts still occur from epoch 500 to 600 and from epoch 700 to 800. From these visualizations, it seems that the weights of early layers (close to the input) converge faster than those in late layers (close to the output). This is also seen in the distribution of the β and γ parameters of the BN layers, which control the mean and the variance of the feature maps that pass through the CNN (Figures 5.34, 5.35, 5.36, 5.37). This observation could be connected to the typical behavior of classifier CNNs trained on image datasets, where early layers learn generic universal geometric kernels, while later layers learn semantically meaningful specialized kernels which can take longer to get established [115, 255]. Additional work delving into the feature maps of the U-Net could clarify this point.

After training the U-Net, examining the correlations between filters in a given convolutional layer can highlight superfluous channels. Highly correlated filters can be probably be pruned to reduce the parameter count of the U-Net. For every pair of channels (i, j), Pearson correlation coefficients $r_{i,j}$ are computed from the corresponding set of filter weights $(w_k^{(i)}, w_k^{(j)})_{k=1,\dots,N}$:

$$r_{i,j} = \frac{\sum_{k=1}^{N}(w_k^{(i)} - \overline{w^{(i)}})(w_k^{(j)} - \overline{w^{(j)}})}{\sqrt{\sum_{k=1}^{N}(w_k^{(i)} - \overline{w^{(i)}})^2}\sqrt{\sum_{k=1}^{N}(w_k^{(j)} - \overline{w^{(j)}})^2}} \tag{5.10}$$

Figure 5.38 and Figure 5.39 contain the cross-correlation matrix $(r_{i,j})$ of the filters of the first and penultimate convolutional layers. They indicate that very few filters are highly correlated with one another. This also implies that each filter is learning a unique, useful transformation of its input feature map. To reduce the size of the network, a promising pruning strategy would be to look for *lottery ticket* subnetworks which perform as well as the full network [100, 404, 414].

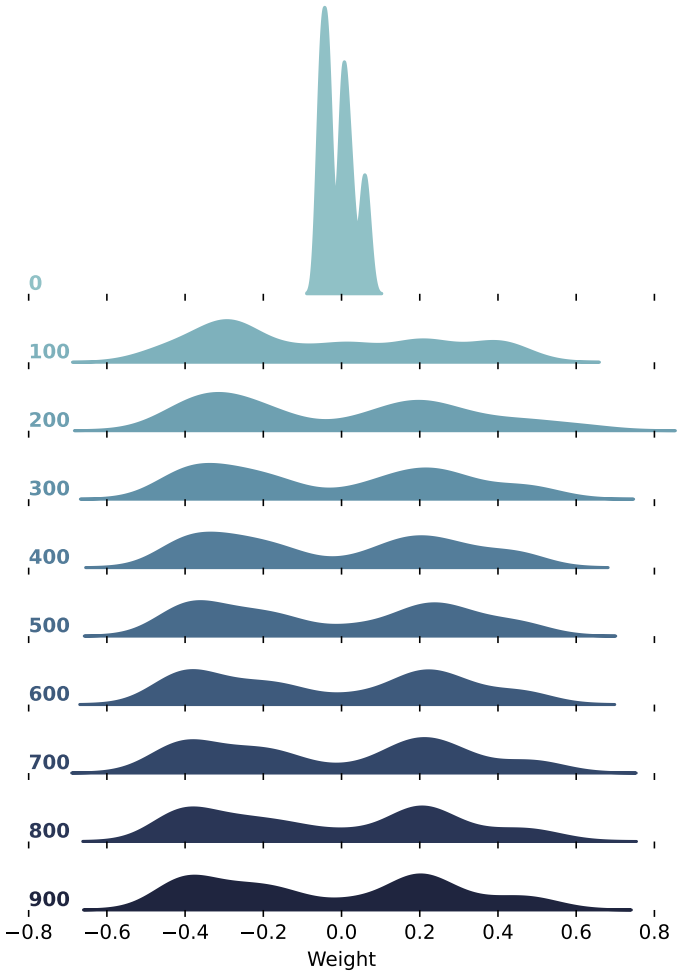


Figure 5.30: KDE plots of the bias distributions at the first convolutional layer, plotted at the end of every 100 training epoch (numbering starts at 0)

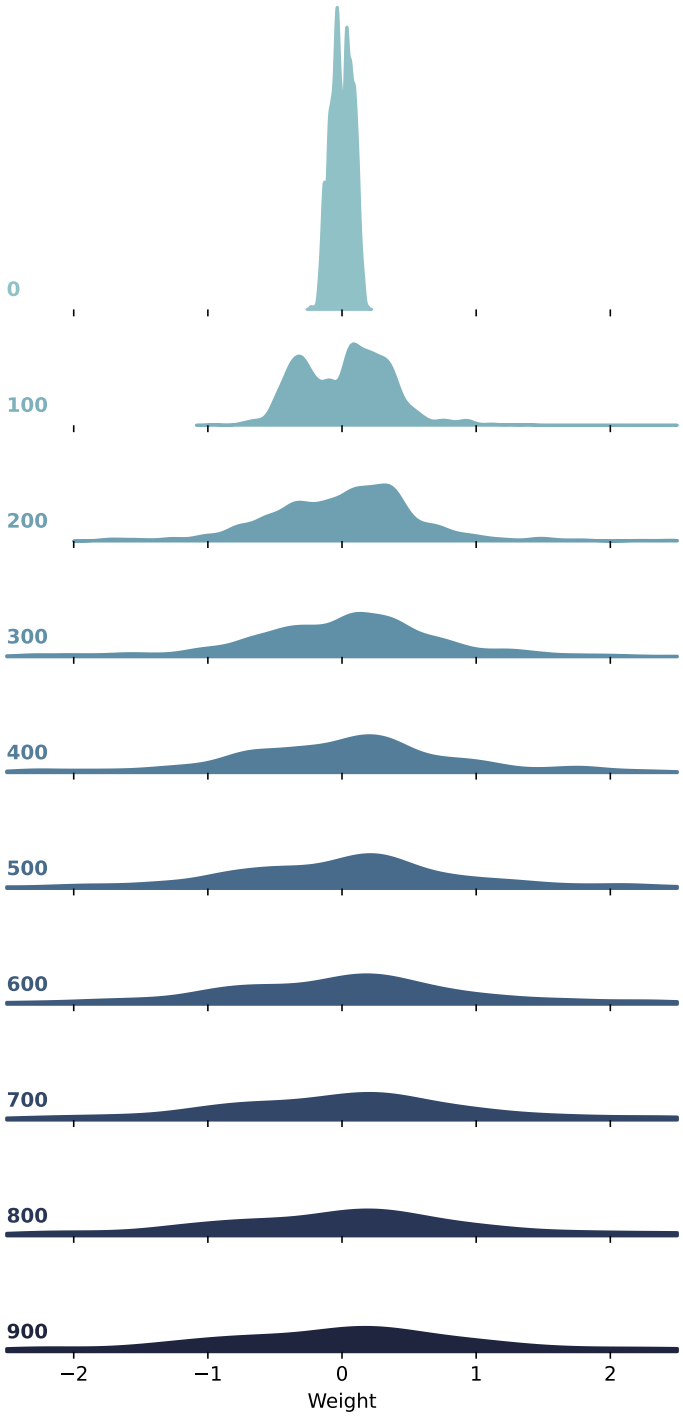


Figure 5.31: KDE plots of the filter weight distributions at the first convolutional layer.

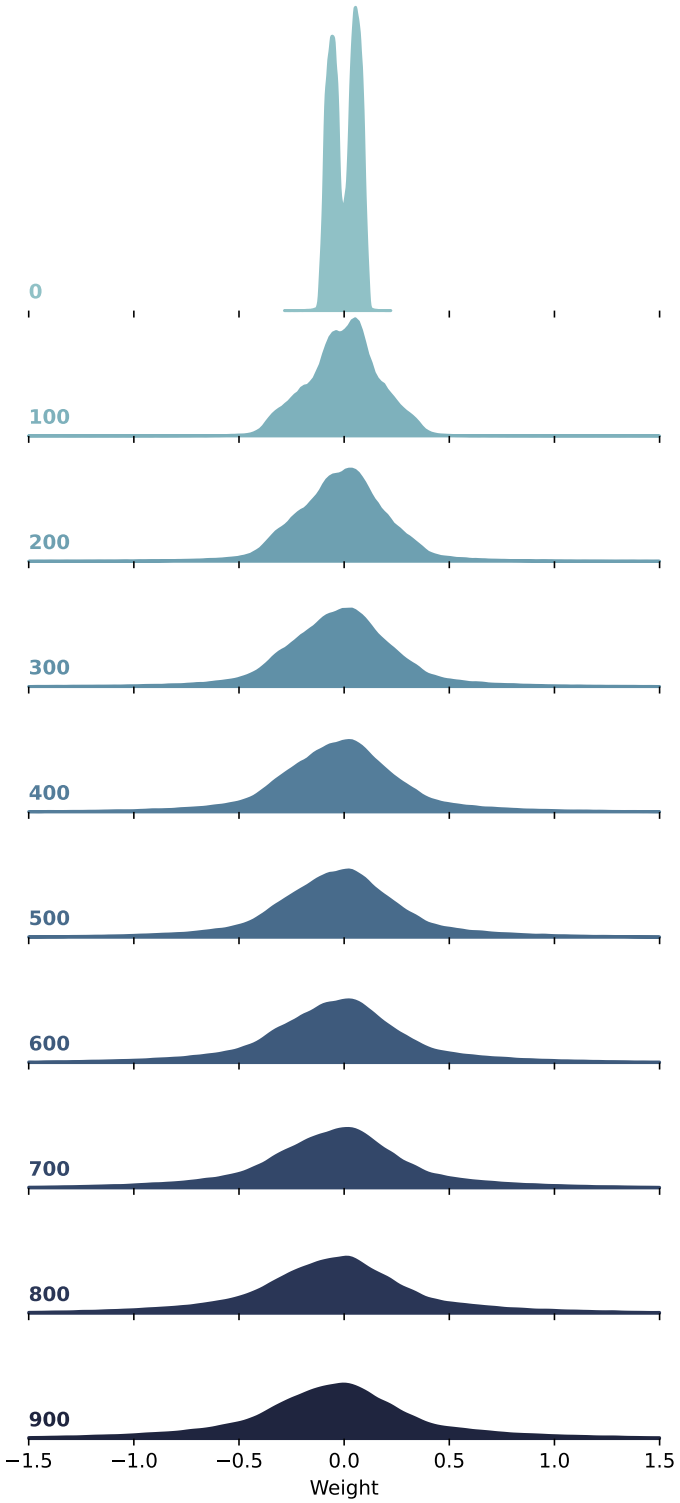


Figure 5.32: KDE plots of the filter weight distributions at the first convolutional layer of the bottom stage.

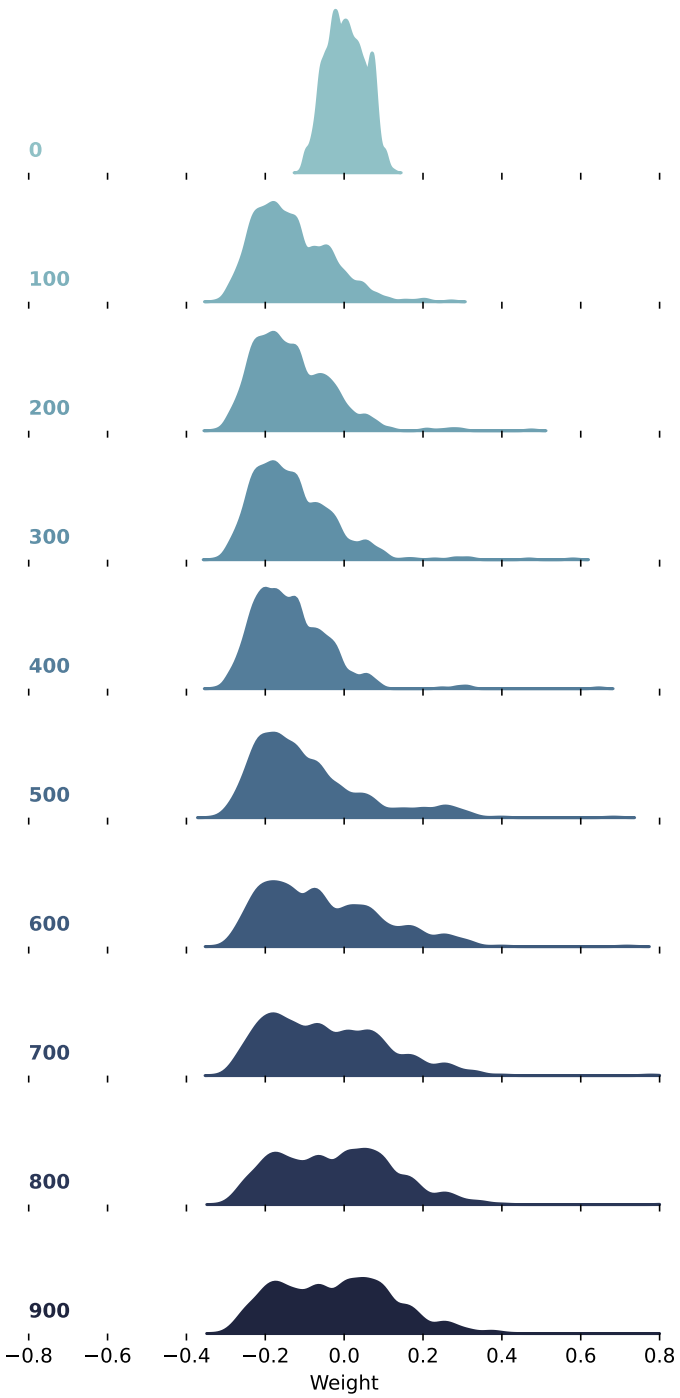


Figure 5.33: KDE plots of the filter weight distributions at the final convolutional layer.

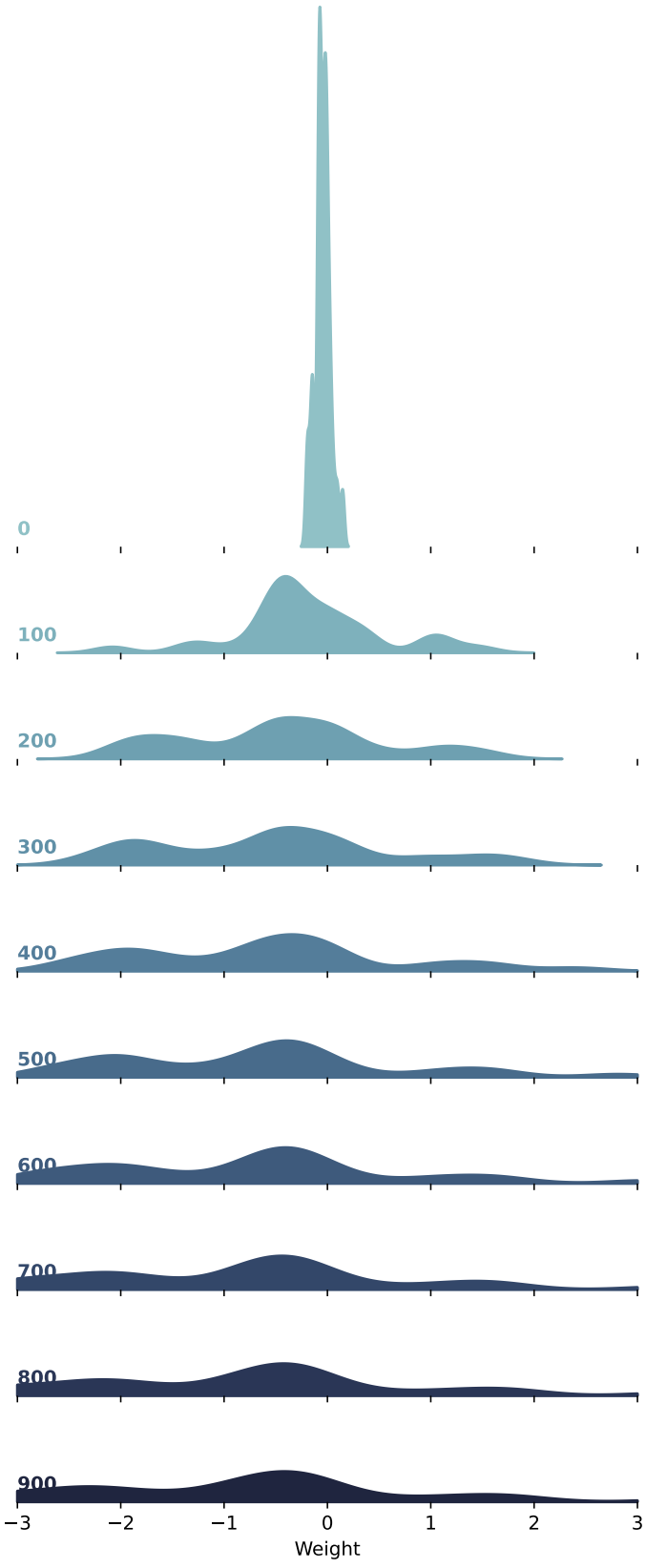


Figure 5.34: KDE plots of the β parameter of the first BN layer.

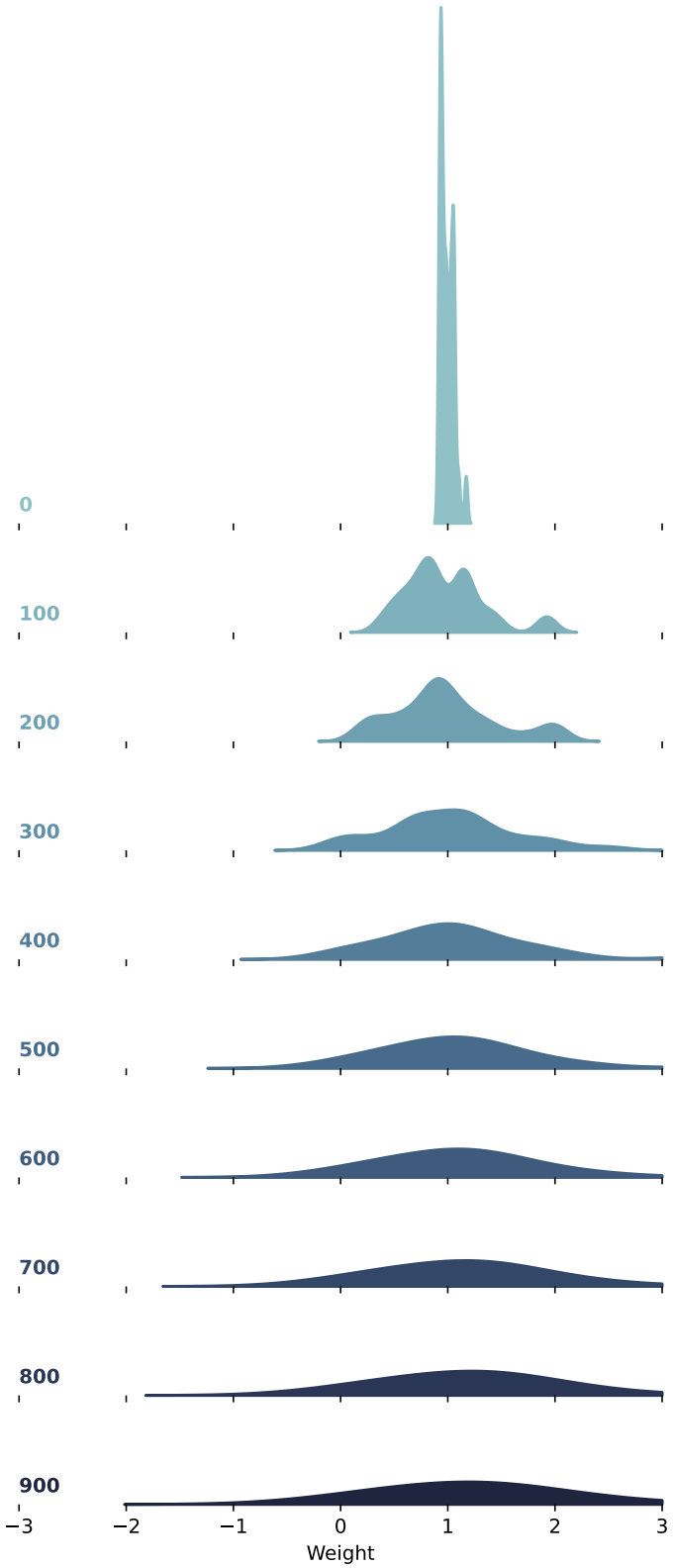


Figure 5.35: KDE plots of the γ parameter of the first BN layer.

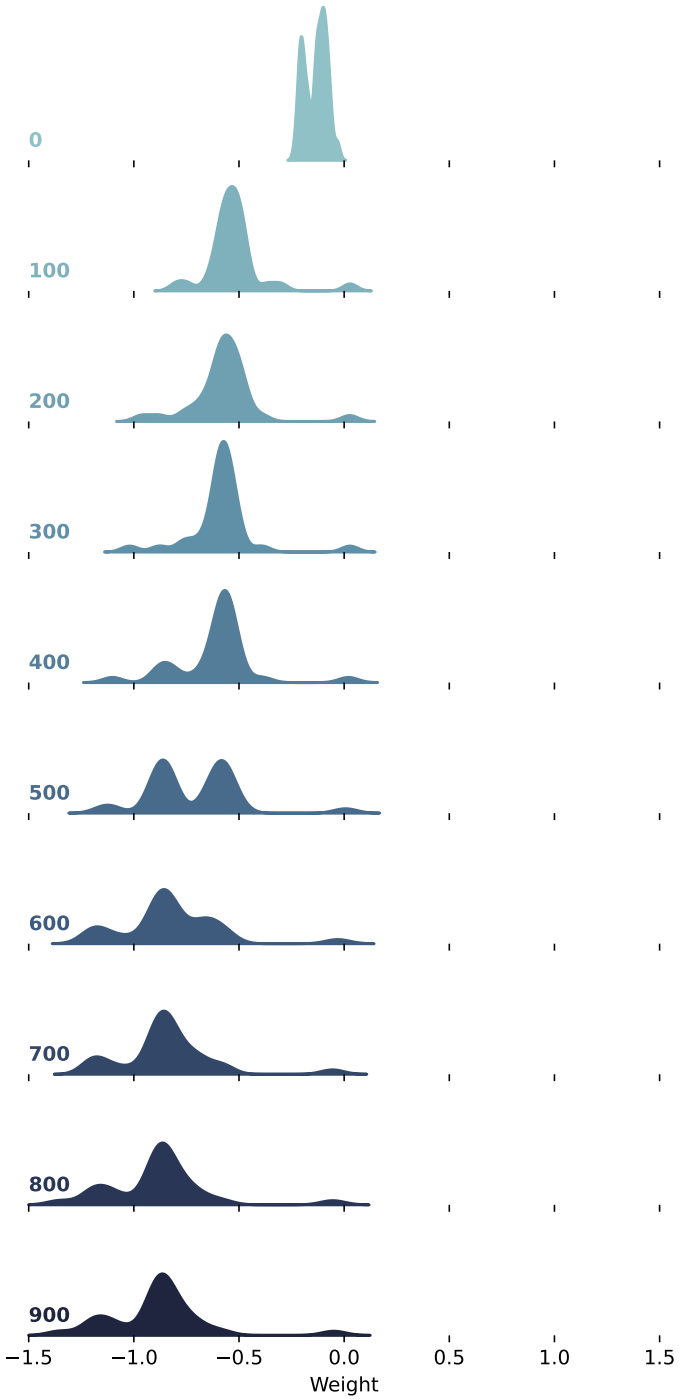


Figure 5.36: KDE plots of the β parameter of the final BN layer.

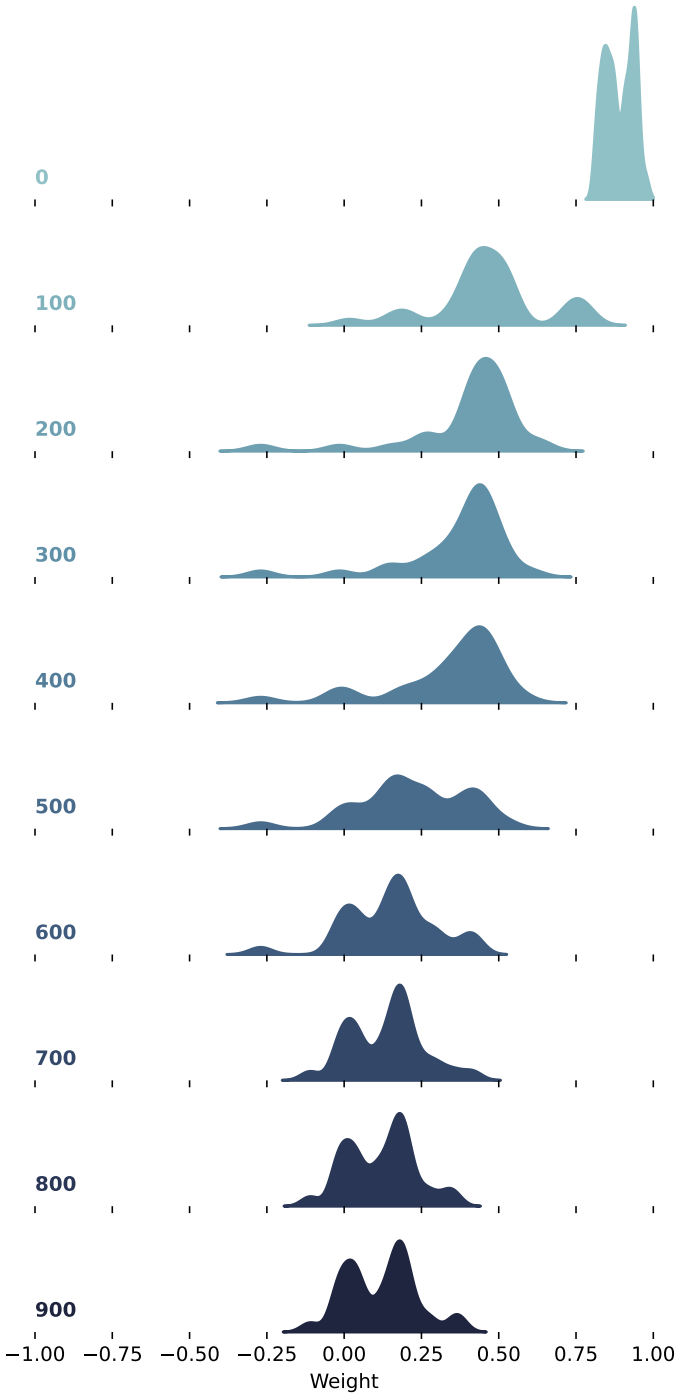


Figure 5.37: KDE plots of the γ parameter of the final BN layer.

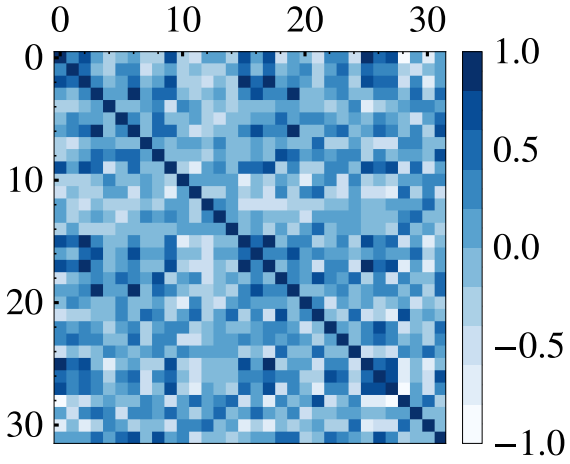


Figure 5.38: Cross-correlation matrix of the filters of the first convolutional layer.

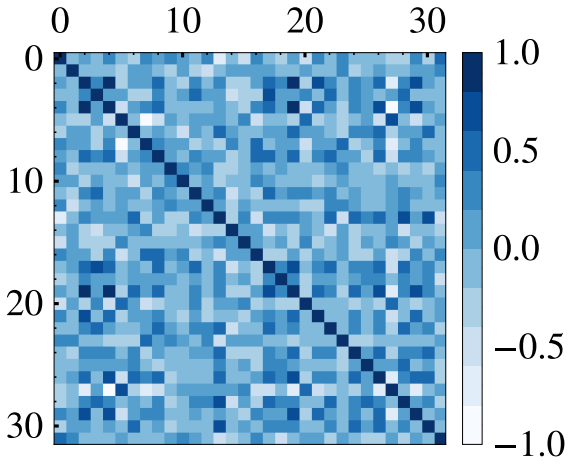


Figure 5.39: Cross-correlation matrix of the filters of the penultimate convolutional layer.

5.9 Conclusion and perspectives

In this chapter, a DL model for the total FSD was trained on the R2 jet flame and evaluated *a priori* on the higher Reynolds number R3 flame. This extends the existing work of Lapeyre et al. [183] to high Reynolds training and test cases. *A priori* evaluations on filtered DNS and LES snapshots underline the excellent performance of the model compared to the state-of-the-art CDSM model. Modifying the nature of the filter and the progress variable formulation to assess the robustness of the model induced minimal losses in accuracy. The effective receptive field of the CNN was shown to cover a large proportion of the flame front, and the depth of the model was justified by visualizations of the weight distributions and channelwise cross-correlations which highlighted a continuous, non-redundant learning process during training. Overall, the success of this approach suggests a strategy to develop DL SGS models for large-

scale LES applications: training the model on the DNS of a reduced configuration seems to generalize well to the full-size simulation.

Evaluating a model on a configuration that only differs by its Reynolds number may be unsatisfying, as one is left to wonder how the model behaves under different pressure, temperature, chemistry, or mixture composition conditions. Generalization to vastly different test cases is still insufficiently explored and will be one of the main axes of Chapter 6.

Chapter 6

Generalization capability of CNN SGS models for the progress variable variance and the filtered reaction rate

Contents

This chapter explores the applicability of DL modeling for PDF turbulent combustion models, with an emphasis on the generalization of deep CNNs far from their training distribution. First, the Pfitzner beta PDF CNN (PB-CNN) model is presented. It relies on the modeling of the SGS progress variable variance by a CNN, which is used in a closed-form expression for the filtered reaction rate based on a presumed beta PDF for the progress variable. The generalization of the CNN to new, complex configurations is assessed by using vastly different training and test datasets. The model is trained on a dataset generated from the DNS of a statistically planar turbulent flame immersed in decaying homogeneous isotropic turbulence. It is then assessed on a highly turbulent jet flame DNS, where its predictions for the SGS progress variable variance are compared to the ground truth of filtered DNS snapshots. The model training procedure follows the same concepts outlined in Chapter 5 for the total FSD model. *A priori* testing of the CNN, and conditions for proper generalization are discussed. Finally, the soundness of the PB-CNN model for the filtered reaction rate is investigated. Most of the developments presented in this chapter have been published in Xing et al. [400].

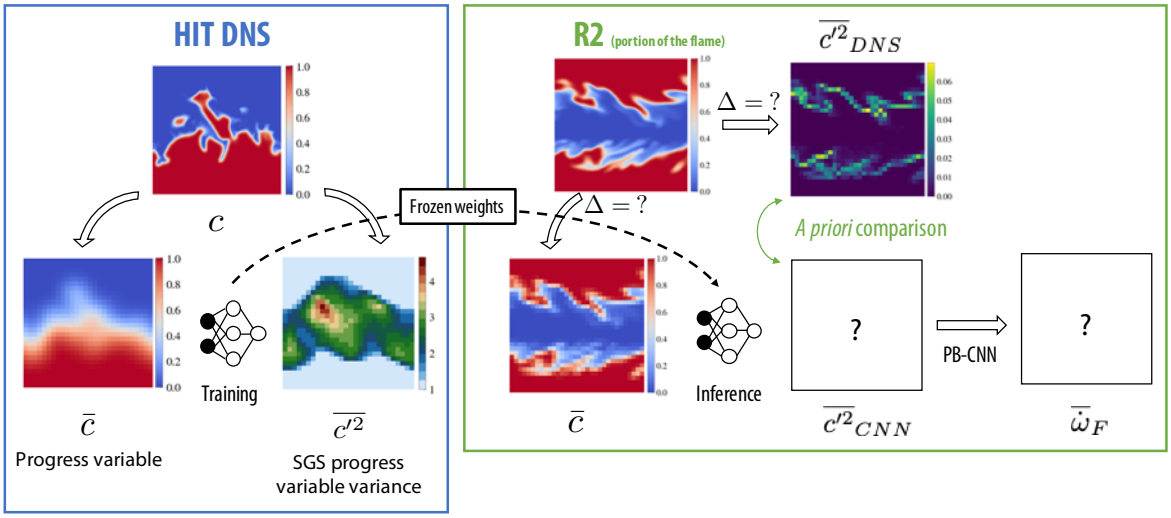


Figure 6.1: Visual outline of Chapter 6.

6.1 The Pfitzner beta PDF CNN model

This section presents a computationally efficient turbulent combustion model which relies on an analytical closure of the filtered reaction rate, and where the main quantity to be closed, the SGS variance of the progress variable $\overline{c'^2}$, is modeled by a CNN. It assumes that combustion can be described by a single-step irreversible chemical reaction, and parametrized by a progress variable $c = 1 - Y_F/Y_F^u$. It is therefore well-suited to large-scale premixed combustion configurations where complex chemistry would be too expensive, and an efficient turbulent combustion model is needed to keep computational costs to a minimum. Due to the analytical nature of the closure and the low overhead of coupling a DL model with AVBP (detailed in Chapter 7), this formulation is a compelling alternative to tabulated chemistry

methods that incur large storage and computational costs [98, 281, 366].

The PB-CNN model is part of the family of PDF models which rely on modeling the subfilter distribution of c, noted as $p(c)$, to express the filtered reaction rate as the expected value of the DNS reaction rate:

$$\overline{\dot{\omega}}_F = \int_0^1 p(c)\dot{\omega}_F(c)\,\mathrm{d}c\,. \tag{6.1}$$

$p(c)$ is the progress variable filtered density function [135, 274, 279] which represents the statistical distribution of the local subfilter state given an LES filter G:

$$p(c = c^*; x) = \int \delta(c - c^*)\, G(x - x')\,\mathrm{d}x'\,. \tag{6.2}$$

$p(c = c^*)\,\mathrm{d}c$ is the G-weighted fraction of fluid around x with a progress variable in the range $[c^*, c^* + \mathrm{d}c]$. $p(c)$ can be modeled as a parametrizable probability distribution. A common choice [35, 36, 69, 182, 286] is the beta PDF that has the expression

$$p_\beta(c) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} c^{a-1}(1-c)^{b-1}\,, \tag{6.3}$$

where Γ is the gamma function, and a and b are two parameters linked to the mean $\overline{c}$ and variance $\overline{c'^2} = \overline{c^2} - \overline{c}^2$ of the distribution by the relations

$$a = \overline{c}\left[\frac{\overline{c}(1-\overline{c})}{\overline{c'^2}} - 1\right], \qquad b = (1-\overline{c})\left[\frac{\overline{c}(1-\overline{c})}{\overline{c'^2}} - 1\right]. \tag{6.4}$$

Some weaknesses of presuming that the SGS distribution of c follows a beta PDF have been pointed out in the literature, namely a departure from the true SGS PDF when the filter size is large and the PDF approaches a bimodel limit [35, 75, 168]. Nevertheless, some experimental evidence points towards the accuracy of the beta PDF up to highly turbulent flame fronts [343, 345], and it has consistently shown good results in LES SGS modeling [71, 184, 286]. A key observation made by Pfitzner et al. [269] is that if a Pfitzner source term is used for the reaction rate (Equation 2.41), then the LES nondimensional reaction rate $\overline{(\omega_m)_\beta}$ has an analytical expression:

$$\overline{(\omega_m)_\beta} = \int_0^1 p_\beta(c)\omega_m(c) \tag{6.5}$$

$$= (m+1)\frac{\Gamma(a+b)}{\Gamma(a)}\left(\frac{\Gamma(a+m+1)}{\Gamma(a+b+m+1)} - \frac{\Gamma(a+2m+1)}{\Gamma(a+b+2m+1)}\right). \tag{6.6}$$

Note that such a simple closed-form formula cannot be found with an Arrhenius source term. This makes the combination of the Pfitzner source term and a presumed beta PDF particularly interesting for computationally efficient evaluations of the filtered reaction rate. Using Equation 2.42, a similar formula can be derived for the fuel reaction rate under the

simplifying assumption $\rho D \approx \overline{\rho D} = \overline{\rho}\widetilde{D}$:

$$\overline{\dot{\omega}}_F = \int_0^1 p_\beta(c)\dot{\omega}_F \,\mathrm{d}c = \Lambda \int_0^1 p_\beta(c)\frac{\omega_m(c)}{\rho(c)D(c)}\,\mathrm{d}c \tag{6.7}$$

$$= \frac{\Lambda}{\overline{\rho}\widetilde{D}}\overline{(\omega_m)_\beta}\,. \tag{6.8}$$

With a Pfitzner beta PDF formulation, the turbulent LES reaction rate can be computed without needing on-the-fly integration of the PDF or tabulated chemistry.

An interesting theoretical remark is that a closed-form formula for the turbulent flame speed associated with the beta PDF source term in Equation 6.5 can be derived via a Kolmogorov-Petrovsky–Piskunov (KPP) analysis. The derivation is presented in Appendix A

6.1.1 Algebraic SGS variance models

To fully close the reaction rate, an SGS model is required for $\overline{c'^2}$. Many closures rely on algebraic expressions based on the gradient of the resolved progress variable. A common gradient model [69, 168, 370] is given by:

$$\overline{c'^2} = C_s\Delta^2|\nabla\overline{c}|^2\,, \tag{6.9}$$

where Δ is the local filter size and C_s is the model parameter. It is important to note that this model was originally proposed for nonreactive scalars, and for reactive scalars such as the progress variable, it should be modified to account for the correlation of the scalar with its source term [271]. Nevertheless, it is still commonly found in the literature whenever an algebraic model for the progress variable is needed [19, 159, 168, 300, 370, 379]. It is therefore used here as a basis of comparison for the CNN, which plays the same role as a traditional algebraic model. The simplest way of estimating C_s is to use a global constant, and values such as 1/12 [19, 168], 0.18 [370], or 0.5 [300] have been proposed in the literature. However, a constant value is not expected to be accurate for all flow configurations and filter sizes [168, 379], and the best model constant is *a priori* unknown. The model of Equation 6.9 with a constant C_s parameter will be called the CST model.

C_s can also be determined by a dynamic procedure [168, 241, 271, 381] recalled in the following. A test filter of size $\hat{\Delta} = 2\Delta$ is used to generate test-filtered quantities, noted as $\hat{\overline{\phi}}$. It is assumed that the model coefficient varies slowly in space, and that Equation 6.9 holds at the filter and test filter levels with the same model coefficient. Its expression at the standard filter level is filtered to the test-filter level:

$$\widehat{\overline{c^2}} - \widehat{\overline{c}}\,\widehat{\overline{c}} = C_s\Delta^2\widehat{|\nabla\overline{c}|^2}\,, \tag{6.10}$$

and independently expressed directly at the test-filter level:

$$\widehat{\overline{c^2}} - \hat{\overline{c}}\,\hat{\overline{c}} = C_s\hat{\Delta}^2|\nabla\hat{\overline{c}}|^2\,. \tag{6.11}$$

After introducing a Leonard term defined as

$$\mathcal{L} = \widehat{\overline{c}\,\overline{c}} - \hat{\overline{c}}\,\hat{\overline{c}}\,, \tag{6.12}$$

and a model term

$$\mathcal{M} = \hat{\Delta}^2 |\nabla \hat{\overline{c}}|^2 - \Delta^2 \widehat{|\nabla \overline{c}|^2}\,, \tag{6.13}$$

the combination of Equations 6.10 and 6.11 leads to:

$$\mathcal{L} = C_s \mathcal{M}\,. \tag{6.14}$$

A least-squares procedure consisting of averaging over homogeneous directions of the flow is used to remedy stability issues [195]. The final expression of the model coefficient is

$$C_s = \frac{\langle \mathcal{L}\mathcal{M} \rangle}{\langle \mathcal{M}^2 \rangle}\,, \tag{6.15}$$

where the averaging operation is denoted by the brackets $\langle \cdot \rangle$. The resulting model for $\overline{c'^2}$ is called DYN.

6.1.2 Convolutional neural networks for SGS variance modeling

The simple algebraic model formulation of Equation 6.9 is driven by sound physical arguments, but also limits the family of $\overline{c'^2}$ fields that these models can fit. As a consequence, algebraic models are often inaccurate [159, 168, 241] and transport equations for $\overline{c'^2}$ are usually preferred [75, 76, 251] but they induce additional computations and require supplementary modeling. To combine computational efficiency and accuracy, $\overline{c'^2}$ is modeled using the U-Net presented in Section 3.4. It is trained on a planar flame wrinkled by homogeneous isotropic turbulence with simplified Pfitzner chemistry and evaluated *a priori* on the DNS of a slot burner jet flame with skeletal chemistry. Since most thermophysical and chemical parameters are purposefully different from one configuration to the other, this is a difficult generalization test for the model. It is assumed that local values of $\overline{c'^2}$ can be determined from three main elements:

1. the profile of the $\overline{c}$ field in the neighboring flame brush,
2. the amount of unresolved SGS scales,
3. the effect of local turbulence on the SGS distribution of c.

If they are known by the model or kept similar in the training and generalization configuration, the model should be able to generalize properly. The essential nature of these elements is highlighted by their presence in the algebraic model of Equation 6.9, respectively, in the form of the terms $|\nabla \overline{c}|$, Δ, and C_s. They therefore guide some of the key choices in this section:

1. A CNN is used because of its ability to accurately learn spatial patterns in an extended area around a location of interest (Section 6.3);

2. the flame fronts seen in the training and evaluation contexts belong to the same turbulent combustion regime (Sections 6.2.1 and 6.2.3);

3. DNS snapshots are filtered and downsampled to a coarse grid with a well-chosen resolution (Section 6.3).

6.2 Training and generalization flow configurations

6.2.1 Training configuration: statistically planar flame in homogeneous isotropic turbulence

The training database is built from the DNS of a freely-propagating, statistically planar flame immersed in decaying homogeneous isotropic turbulence (HIT), a common configuration in the literature [33, 43, 46, 159]. The cubic domain is uniformly meshed with 384^3 hexahedral elements and a mesh size $h = 36\,\mu\text{m}$. In the x direction, NSCBC inlet and outlet boundary conditions are prescribed [277], while the other boundary conditions are periodic (Figure 6.2). The flame front is initialized along the x direction using a DNS solution of a 1D laminar propane–air flame at stoichiometry, temperature $T = 300\,\text{K}$, and pressure $P = 1\,\text{bar}$. Chemistry is represented by a single irreversible reaction $C_3H_8 + 5\,O_2 \longrightarrow 3\,CO_2 + 4\,H_2O$ and the fuel reaction rate is computed using the Pfitzner DNS formulation of Equation 2.42. As key inputs needed in the formulation, the laminar flame speed $s_L = 0.383\,\text{m}\,\text{s}^{-1}$ and the model parameter $m = 3.8$ are chosen to match the laminar flame speed and flame thickness $\delta_{L,c} = 352\,\mu\text{m}$ obtained with single-step Arrhenius chemistry [291]. The flame thickness is based on the gradient of c:

$$\delta_{L,c} = \frac{1}{\max \frac{\mathrm{d}c}{\mathrm{d}x}}\,. \tag{6.16}$$

The initial laminar flame front in the HIT is therefore resolved on approximately 10 mesh points.

In the fresh gases, turbulent velocity fluctuations are superimposed on the velocity field of the laminar flame with the Fourier series decomposition method presented in Section 5.7.1. This time, a von Karman–Pao (VKP) spectrum is used [17]:

$$\mathcal{E}(\kappa) = \alpha \frac{u'^2}{\kappa_e} \frac{(\kappa/\kappa_e)^4}{[1 + (\kappa/\kappa_e)^2]^{17/6}} \exp\left[-2\left(\frac{\kappa}{\kappa_\eta}\right)^2\right]\,, \tag{6.17}$$

where κ_e, κ_η are the wavenumber of the most energetic eddies and wavenumber of the Kolmogorov scale η, respectively. $\alpha = 1.453$ is a numerical constant. The integral length scale l_t is related to κ_e through $l_t \approx 0.747/\kappa_e$ [17]. u' and κ_e are chosen so that

$$u'/s_L = 10\,, \qquad\qquad l_t/\delta_{L,c} = 4.8\,, \tag{6.18}$$

leading to $u' = 3.83\,\text{m}\,\text{s}^{-1}$ and $\kappa_e = 423\,\text{m}^{-1}$. κ_η is obtained via the dissipation rate $\epsilon \approx u'^3/l_t$, which leads to $\kappa_\eta = 2\pi/\eta = 2\pi\nu^{-3/4}\epsilon^{1/4} = 3.49 \times 10^5\,\text{m}^{-1}$. The ratio of the mesh size to the Kolmogorov scale is $h/\eta = 2$, so that the mesh is fine enough to resolve

all the eddies of the initial spectrum. Table 6.1 summarizes the main flame and turbulence parameters at initialization, including the Damköhler number $Da = (l_t/\delta_{L,c})/(u'/s_L)$, Karlovitz number $Ka = (u'/s_L)^{3/2}(l_t/\delta_{L,c})^{-1/2}$, turbulent Reynolds number $Re_t = u'l_t/\nu$, Taylor-scale Reynolds number $Re_\lambda = u'\lambda_g/\nu$ based on the Taylor microscale λ_g, and eddy turnover time $\tau = l_t/u'$.

s_L	$\delta_{L,c}$	u'	l_t	η	Da	Ka	Re_t	Re_λ	τ
$0.383\,\mathrm{m\,s^{-1}}$	352 µm	$3.83\,\mathrm{m\,s^{-1}}$	1.77 mm	18 µm	0.48	14	450	82	0.46 ms

Table 6.1: HIT initial parameters.

The DNS is run using the fully compressible explicit code AVBP [327, 328] with the TTGC Taylor–Galerkin finite element scheme of third-order accuracy in time and space [68]. The simulation is run for 1.84 ms, corresponding to 4-eddy turnover times.

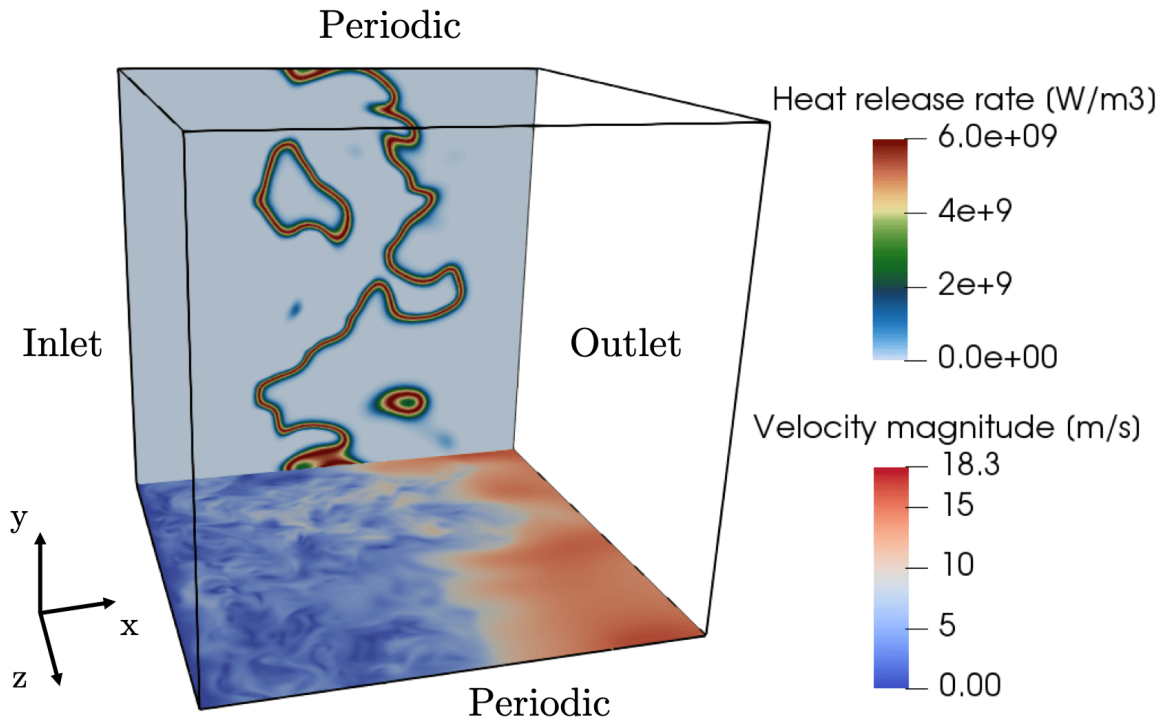


Figure 6.2: Slices of the HIT DNS at $t = 2\tau$: velocity magnitude in the xz-plane and heat release rate in the xy-plane.

6.2.2 Exploration of the HIT dataset

To assess the quality of the HIT DNS, its turbulence and combustion properties are analyzed in post-processing. Its turbulent evolution is studied by performing a cold simulation without combustion, where the flow is an inert decaying homogeneous isotropic turbulence. This cold HIT is run over three eddy turnover times. Then, combustion-relevant quantities and

visualizations are analyzed for the reacting HIT DNS.

Turbulent kinetic energy dissipation The impact of numerical effects on the dissipation of the TKE k is first assessed. Numerical dissipation must remain small relative to physical dissipation for the decaying turbulence to be meaningful. The balance equation for the TKE k can be expressed using the time derivatives of density and momentum:

$$\frac{\partial k}{\partial t} = \boldsymbol{u} \cdot \frac{\partial \rho \boldsymbol{u}}{\partial t} - \frac{\boldsymbol{u} \cdot \boldsymbol{u}}{2} \frac{\partial \rho}{\partial t} . \tag{6.19}$$

Then, a discretized balance equation of the TKE can be derived from the discretized versions of the Navier-Stokes equations 2.3, 2.4 that are solved by AVBP. Details are contained in the PhD thesis of V. Moureau [235]. The rate of evolution of the discretized TKE averaged over the computational domain is finally written as:

$$\left\langle \frac{\Delta k}{\Delta t} \right\rangle = \underbrace{\epsilon_{vis} + \epsilon_{ac}}_{\epsilon_{phys}} + \underbrace{\epsilon_{conv} + \epsilon_{artif}}_{\epsilon_{num}} , \tag{6.20}$$

where various dissipation source terms are identified. The physical dissipation ϵ_{phys} is caused by energy transfers due to molecular diffusion (ϵ_{vis}) and acoustic effects (ϵ_{ac}). The numerical dissipation ϵ_{num} is composed of the dissipation of the convective scheme (ϵ_{conv}) and of the artificial viscosity (ϵ_{artif}). Physical and numerical dissipations are normalized by $\langle \Delta k / \Delta t \rangle$, indicated by the superscript $^+$, and plotted in Figure 6.3. It appears that the numerical dissipation remains two orders of magnitude smaller than the rate of evolution of the TKE and of the physical dissipation. This indicates that the numerical effects in the DNS do not play a large role in the evolution of the TKE.

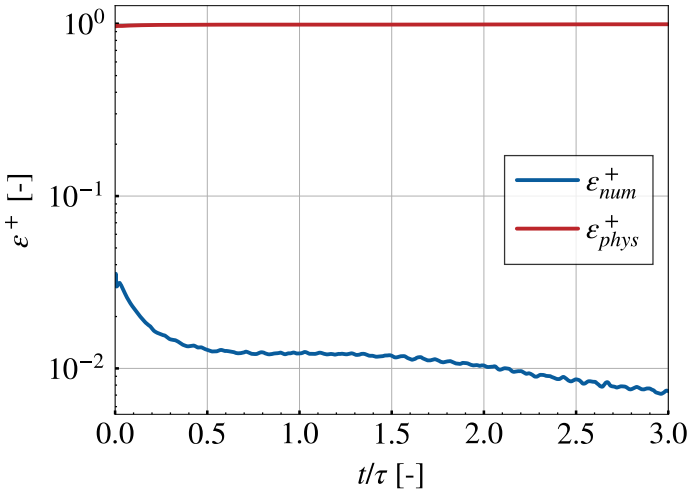


Figure 6.3: Normalized numerical and physical dissipation in the cold HIT.

Turbulent kinetic energy decay Next, the rate of decay of the mean TKE is investigated. For a decaying HIT with a turbulence spectrum that behaves as $\mathcal{E}(\kappa) \sim \kappa^4$ when

$\kappa \to 0$ like the VKP spectrum (Equation 6.17), turbulence theory and numerical evidence indicate that the mean TKE decays asymptotically as $\langle k \rangle \sim t^{-10/7}$ [149]. The time evolution of the mean TKE is reported in Figure 6.4, normalizing the mean TKE by its initial value and time by the eddy turnover time. The rate of decay consistently increases and eventually approaches the theoretical slope of $-10/7$, although this simulation may be too short to make a proper comparison with this theoretical limit. DNS studies of decaying turbulence report decay rates of exactly $-10/7$ only after very long times, *i.e.* $t = \mathcal{O}(10\tau)$ [149]. Given the limited time span of this simulation, the TKE decay rate seems to agree with the usual behavior of decaying turbulence.

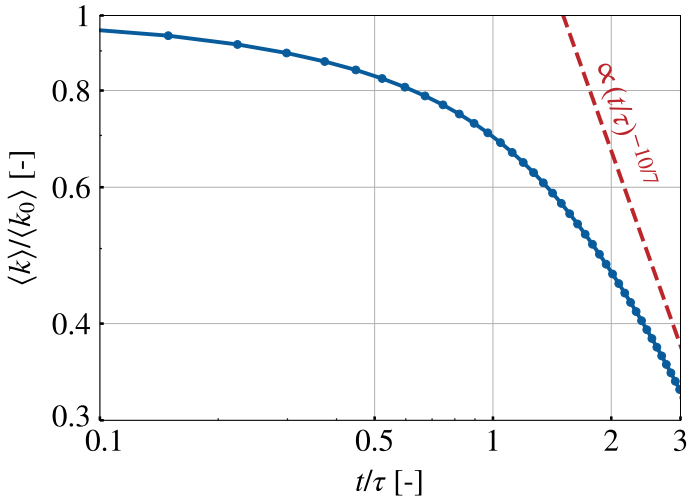


Figure 6.4: Time evolution of the mean turbulent kinetic energy in the cold HIT.

Turbulence spectrum The turbulence spectrum in the HIT at initialization and at later times is shown in Figure 6.5. The energy spectrum function and the wavenumbers are non-dimensionalized by Kolmogorov scales. Compared to the Passot-Pouquet spectrum of Equation 5.9, the VKP spectrum has the advantage of extending far into dissipative wavenumbers, as the wavenumber in the exponential term of the energy spectrum function is normalized by the dissipative wavenumber κ_η instead of the large-scale wavenumber κ_e. The initial spectrum mostly follows the Kolmogorov scaling law $\mathcal{E}(\kappa) \propto \kappa^{-5/3}$, with no clear dissipative range. This dissipative range would only appear in the initial spectrum if the mesh size was significantly smaller than the Kolmogorov scale, an unfeasible resolution constraint in this simulation. A dissipative range appears as soon as $t = 0.1\tau$, indicating that the synthetic turbulence spectrum has relaxed to realistic turbulence. The characteristic lengthscale that separates the inertial and dissipative ranges seems to lie around 5η. It is larger than the laminar flame thickness whose associated wavenumber is indicated by the black dash-dotted line. The range of turbulent eddies that are able to wrinkle the flame front therefore extends until the observed dissipation range. This further justifies the use of a VKP spectrum whose broad spectral range generates eddies up to the smallest wrinkling lengthscale right away, instead of needing to wait for large eddies to break up if a Passot-Pouquet spectrum had been used. Over the course of the simulation ($t = \tau$, $t = 2\tau$), the shape of the turbulence

spectrum stays mostly the same, and the energy decay appears in the downward shift of the spectrum.

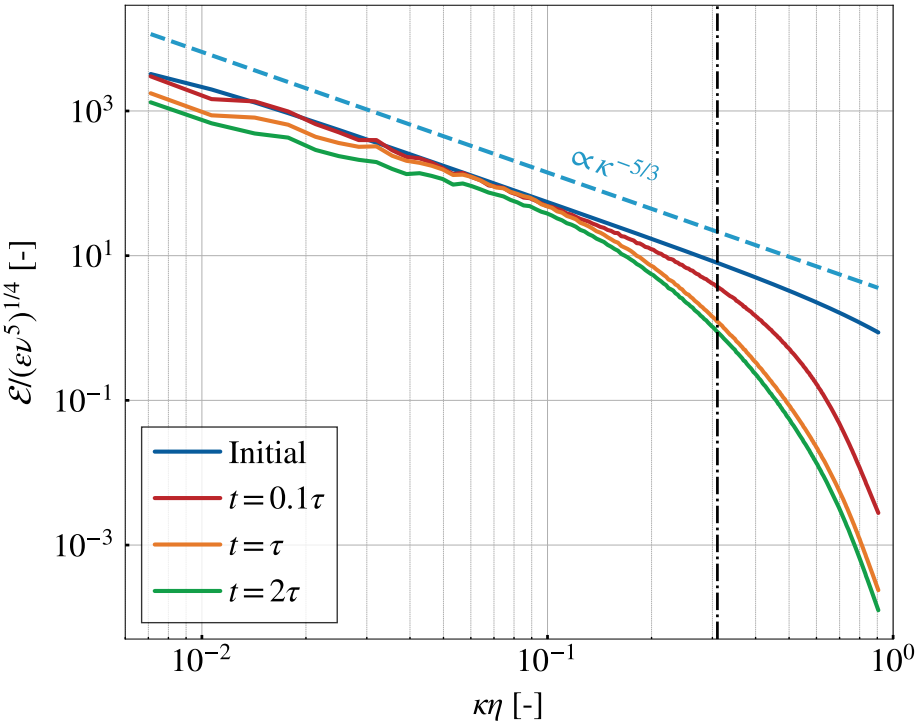


Figure 6.5: Turbulence spectrum normalized by Kolmogorov scales in the cold HIT. The dash-dotted black line marks the wavenumber associated with the laminar flame thickness.

Turbulent evolution In the reacting simulation, turbulent eddies wrinkle the flame front. Figure 6.6 shows a representative slice of the turbulent flow field, illustrated by the normal vorticity component, and the turbulent flame front at $t = 3.9\tau$. Turbulent eddies wrinkle the flame front on multiple scales, and 3D wrinkling effects are evidenced by pockets of $c = 0.1$ and $c = 0.9$ isosurfaces in the fresh and unburnt gases. Turbulence is dissipated through the flame front by the combination of the increase in viscosity with the temperature and dilatational dissipation [318].

The evolution of the reacting HIT in the Borghi-Peters turbulent combustion diagram is plotted in Figure 6.7. In this figure, u' is the density-weighted averaged turbulent velocity fluctuation in the preheat zone (defined as the region where $0.1 < c < 0.5$). l_t is computed as $u'^3/\tilde{\epsilon}$, where the Favre-averaged dissipation rate is given by [318]:

$$\tilde{\epsilon} = \frac{1}{\bar{\rho}}\left[\overline{\mu\boldsymbol{\omega}\cdot\boldsymbol{\omega}} + \frac{4}{3}\overline{\mu(\nabla\cdot\boldsymbol{u})^2} + 2\overline{\mu(\nabla\otimes\nabla):(\boldsymbol{u}\otimes\boldsymbol{u})} - 4\overline{\mu\nabla\cdot(\boldsymbol{u}(\nabla\cdot\boldsymbol{u}))}\right] \tag{6.21}$$

and accounts for both solenoidal and dilatational contributions.

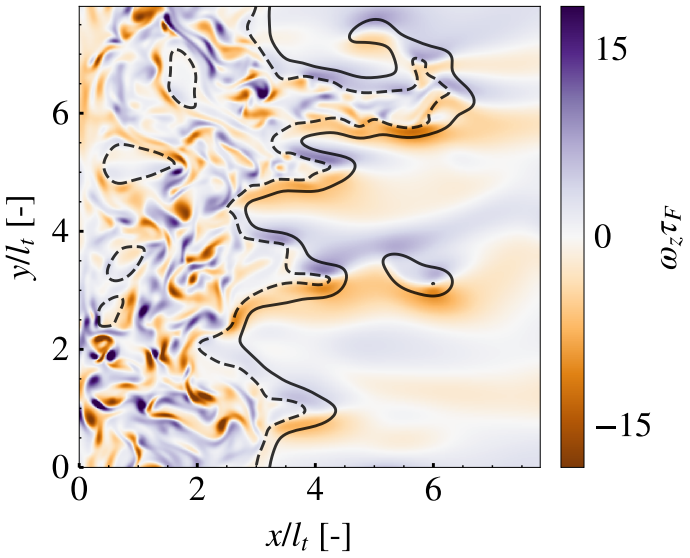


Figure 6.6: z component of the vorticity normalized by the flame time on a z-normal slice at $t = 3.9\tau$, with iso-lines $c = 0.1$ (dashed), $c = 0.9$ (solid). Fresh (resp. burnt) gases are located on the left (resp. right) of this figure.

Throughout the simulation, the flame evolves in the TRZ regime, where in theory a broadening of the turbulent flame brush may be observed but the inner reaction zone should not be significantly perturbed. Interestingly, although there is no forcing term for the turbulence, the computational estimates of u' and l_t increase during the simulation. This may be due to velocity fluctuations generated by the propagation of the wrinkles in the flame front. Note that the computational estimates of u' and l_t lead to different initial values than those reported in Table 6.1 which come from the theoretical VKP spectrum. This discrepancy may be due to the limited extent of the preheat zone in which these quantities are averaged. This should tend to reduce the computational estimates of u' and l_t which is what is observed in Figure 6.7. There is no definite right way of computing these quantities, and both estimates indicate that the flame lies in the TRZ.

Finally, the relative increase in flame surface area and the fuel consumption rate are plotted in Figure 6.8. Turbulent wrinkling increases the flame surface up to a factor 6, and the fuel consumption rate consistently tracks the increase in flame surface area, with deviations no greater than 15%. This indicates that the flame preserves its flamelet structure throughout the DNS, as expected from the TRZ regime. The evolution of the flame surface is thus a good proxy for the evolution of the total burning rate. This property will be used for the CNN model trained in Chapter 8.

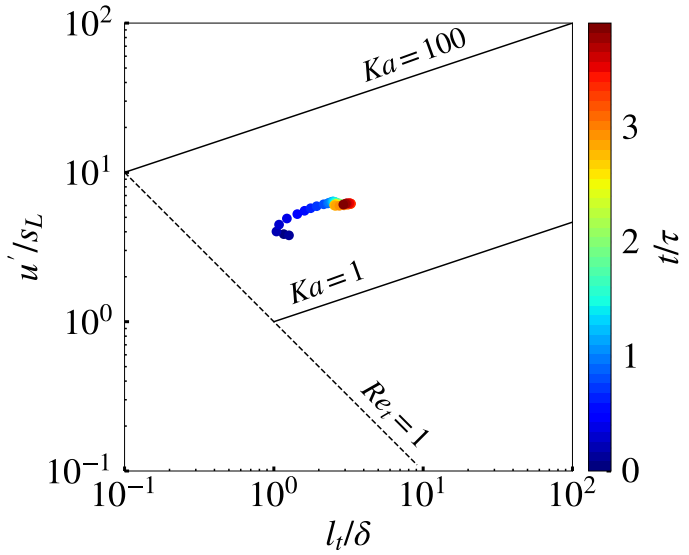


Figure 6.7: Borghi-Peters turbulent combustion diagram, with the location of HIT snapshots colored by time.

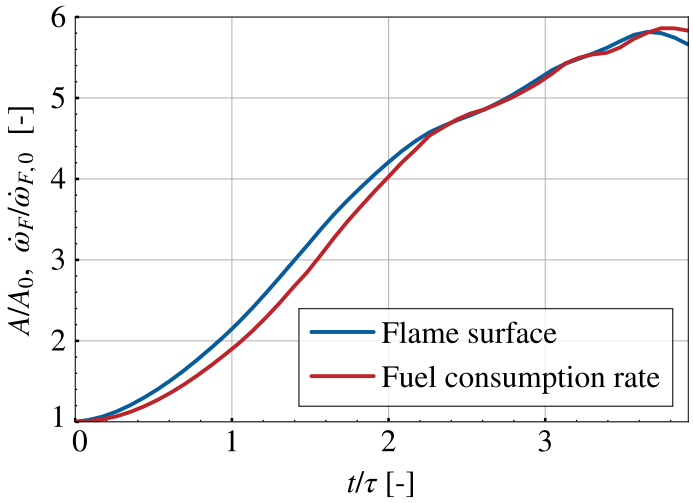


Figure 6.8: Time evolution of the flame surface and fuel consumption rate normalized by their initial values in the reacting HIT.

6.2.3 Generalization configuration: R2 slot burner jet flame

The configuration used to assess the generalization of the CNN is the DNS of the R2 slot jet flame already used in Chapter 5 and briefly recalled here. A fully premixed methane–air mixture at temperature $T = 800\,\mathrm{K}$, pressure $P = 4\,\mathrm{bar}$, equivalence ratio $\phi = 0.7$, and bulk velocity $U = 100\,\mathrm{m\,s^{-1}}$ is injected through a slot of width $H = 1.2\,\mathrm{mm}$ and surrounded by a slow coflow of burnt gases. Chemistry is described by a skeletal chemical mechanism containing 16 species and 72 reactions [207]. The corresponding laminar flame has a speed of

$s_L = 1\,\mathrm{m\,s^{-1}}$ and a progress variable thickness $\delta_{L,c} = 85\,\mathrm{m\,s^{-1}}$. The domain dimensions are 28.8 mm, 19.2 mm, and 5.16 mm in the streamwise (x), crosswise (y), and spanwise (z) directions, respectively. It is uniformly meshed with a resolution $h = 20\,\mu\mathrm{m}$. The progress variable field of the fully developed jet flame is shown in Figure 6.9. 5 uncorrelated snapshots from instants where the jet is fully developed are used in this study, and form what will be called the *generalization* set.

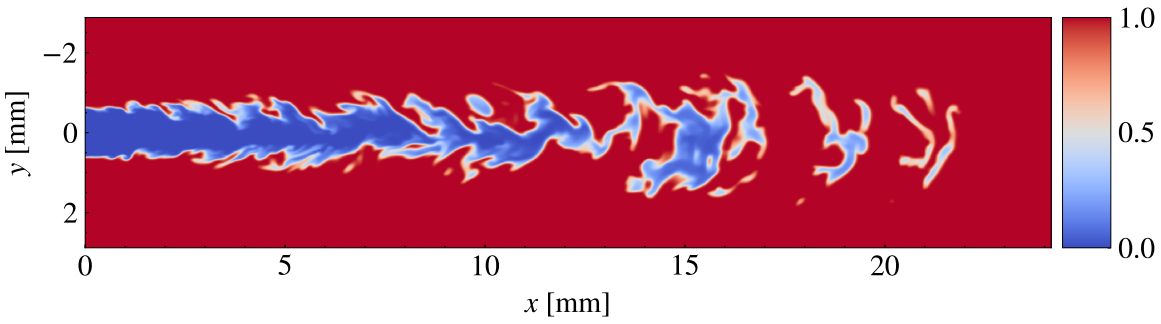


Figure 6.9: Slice of progress-variable field in R2 DNS.

6.2.4 Comments on the differences and similarities between the two configurations

The HIT and R2 flames feature very different domain geometries, flow dynamics, and simulation parameters (Table 6.2). In addition, turbulence in the flame brush is induced by homogeneous isotropic turbulence in one case and shear between the jet and the coflow in the other. Previous studies involving CNNs investigated minor parametric variations in the inlet condition [183], fuel species, and Karlovitz number [403], or turbulence intensity [250, 305]. Substantial generalization was observed by Wan et al. [382] for a fully connected neural network trained as a surrogate model for the filtered reaction rate on a micromixing database.

This chapter investigates the capacity of the CNN trained on the HIT to generalize well to flames in the same premixed turbulent combustion regime, characterized by the velocity and length ratios u'/s_L, $l_t/\delta_{L,c}$ [265, 266, 275]. To this end, in Equation 6.18, the parameters of the initial turbulence spectrum of the HIT were chosen to match the u'/s_L, $l_t/\delta_{L,c}$ values computed by Luca et al. [209] at the crosswise location of maximum heat release and 60% of the flame length. They are considered to be representative of the turbulent combustion regime of the R2 flame.

Finally, three orders of magnitude separate the computational cost of the two DNS simulations. To simulate 0.9 ms of physical time, R2 requires 1 million wall clock hours on an Intel Xeon Haswell-based supercomputer [208], while the HIT only needs 1600 wall clock hours on a comparable Intel Xeon Skylake-based cluster. This highlights the greatly reduced computational demands of training an ML model on simple canonical configurations instead of full-scale realistic flames.

	HIT	**R2**
Fuel	C_3H_8	CH_4
Reactions	1	72
Species	5	16
ϕ	1	0.7
T	300 K	800 K
P	1 bar	4 bar
s_L	$0.383\,\mathrm{m\,s^{-1}}$	$1\,\mathrm{m\,s^{-1}}$
$\delta_{L,c}$	352 µm	85 µm
h	36 µm	20 µm

Table 6.2: Main differences in the simulation parameters of the HIT and R2 DNS.

6.3 Data preparation and model training

The CNN trained in this study is the same U-Net architecture used in Chapter 5 to model the total FSD. It is trained to process $\overline{c}$ input fields and predict associated $\overline{c'^2}$ output fields. Following the LES framework, the filtered fields are represented on a grid of size h that is coarser than the DNS mesh. To generate these filtered coarse fields, 3D fields of $c = 1 - Y_F/Y_F^u$ are first extracted from instantaneous snapshots of the DNS simulations. They are then filtered with a Gaussian filter (Equation 4.10) and downsampled to a coarse mesh of size h_{coarse}. The filtering and coarsening procedure for HIT and R2 is illustrated in Figure 6.10. The PB-CNN model does not involve thickening, so here the combustion filter size Δ is the LES filter size, equated with the coarse mesh size resolution h_{coarse} [313]. Following Propositions 4.2 and 4.3, the value of the coarse mesh resolution for the HIT and R2 is chosen to ensure that the ratio $\Delta/\delta_{L,c} = 0.81$ is the same for both cases, leading to $h_{coarse} = \Delta = 288\,\mu\mathrm{m}$ and 70 µm for the HIT and R2, respectively.

The training database is built from 46 instantaneous snapshots of the HIT DNS, extracted at regular intervals of 0.04 ms. The first 38 constitute the training set for the CNN, while the next 4 form the validation set and the final 4 are kept as a held-out test set for the results shown below. Instead of directly predicting $\overline{c'^2}$, the CNN is trained to predict the unmixedness factor $\overline{c'^2}/[\overline{c}(1-\overline{c})]$, a normalized version of the variance that was empirically found to lead to slightly better results. As a preprocessing step, the input and output fields are periodically padded in the y and z directions to enforce the periodicity boundary conditions of the DNS.

At each training iteration, 16 snapshots from the training set are sampled and, for each snapshot, 4 randomly cropped 32^3 cubes are passed through random 90° rotations and mirror operations before being added to the training batch. Gradient descent optimization is performed by an Adam optimizer [164] with an initial learning rate of 0.01. A mean squared error loss function encourages the CNN outputs to match the ground truth $\overline{c'^2}$ fields. Training was performed with the Tensorflow 2 Python library and stopped after 10,000 iterations when convergence was well-established. On an NVIDIA Tesla V100 GPU, this equates to one hour of training time. This training procedure only needs to be done once, after which the weights

of the CNN are frozen and the trained model can predict $\overline{c'^2}$ from a given $\bar{c}$ field in a few milliseconds.

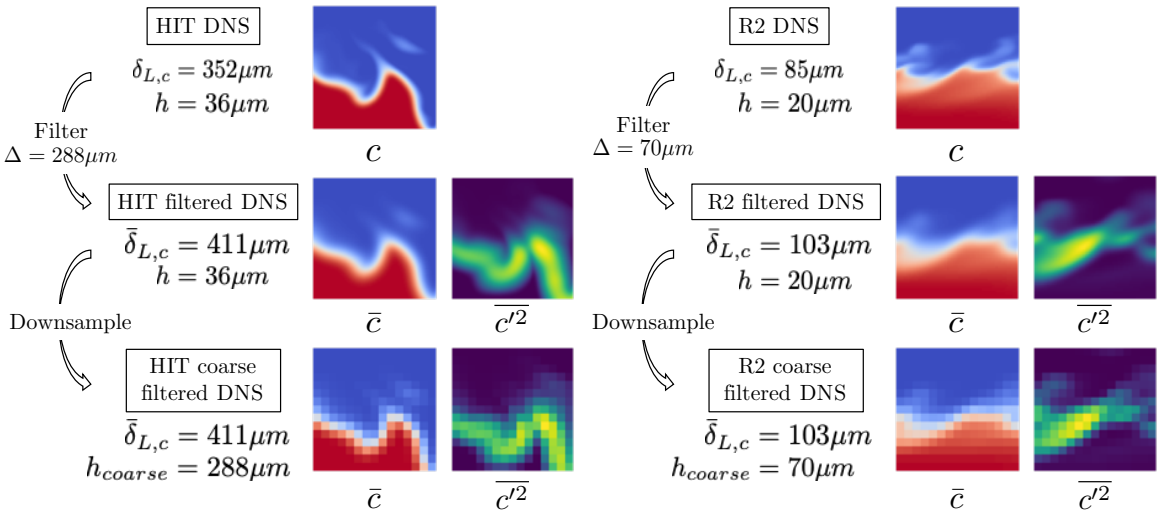


Figure 6.10: Summary of the filtering process for the HIT and R2 for $\Delta/\delta_{L,c} = 0.81$, illustrated with enlargements of the corresponding flame.

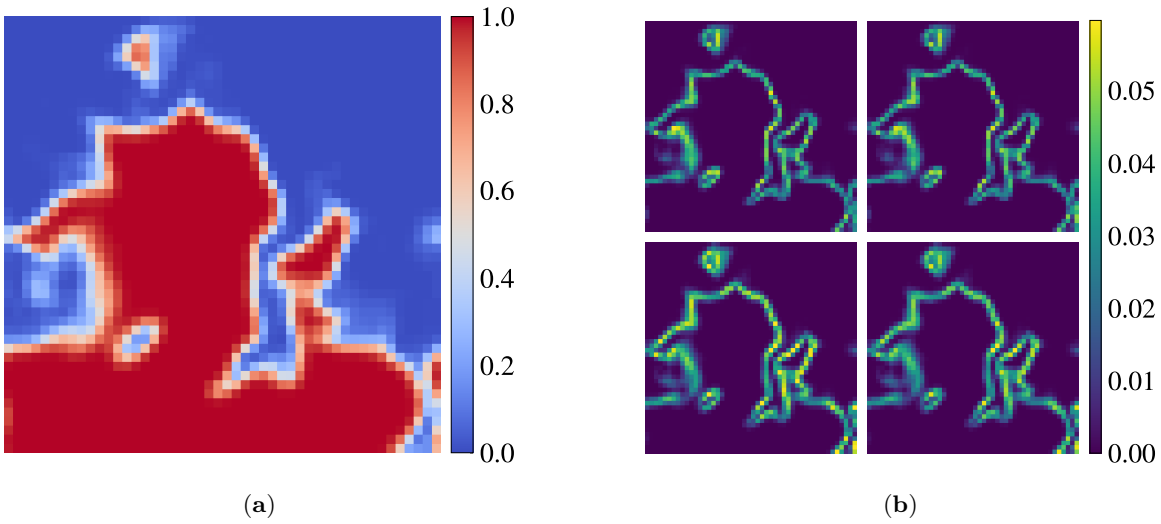
6.4 *A priori* evaluation of the SGS variance model

6.4.1 Evaluation on the HIT test set

The *a priori* performance of the CNN on the test set of the HIT is assessed to ensure that it has successfully learned to represent $\overline{c'^2}$ on samples from the same distribution as the training database. The CNN model predictions are compared to ground truth $\overline{c'^2}$ values computed from the filtered downsampled DNS, called *true* values, and to the CST model predictions. The constant coefficient $C_s = 0.18$ [370] was chosen by picking the value in the literature [19, 168, 300, 370] that best matched the true $\overline{c'^2}$ values. For the DYN model, since there are no fully homogeneous directions, the averaging procedure is conducted on isocontours of $\bar{c}$. Another possibility is to perform averaging over the whole model, leading to a dynamically determined constant $C_s = 0.17$, which is very close to the value chosen for the CST model.

Figure 6.11 shows sample slices of $\bar{c}$, true $\overline{c'^2}$, and modeled $\overline{c'^2}$ fields from the test set. The CNN matches the ground truth faithfully in all regions of the flow, while the gradient model does not seem as accurate.

Model predictions are aggregated on the whole test set and plotted against true values in Figure 6.12. The excellent performance of the CNN is confirmed by the tight spread of its predictions around the perfect model line, and the lack of bias of their linear trend. In comparison, CST and DYN model predictions display significantly higher variance and nonzero bias. Note that with prior knowledge of the ground truth, a C_s value that leads to an unbiased CST model can be chosen but this would have little effect on the variance. The dynamic procedure proves to be effective at estimating a local model coefficient without any prior knowledge and leads to a slightly smaller error than the CST model.



(**a**) (**b**)

Figure 6.11: Sample slices of $\overline{c}$ (**a**), and $\overline{c'^2}$ (**b**): ground truth (top-left), CNN (top-right), CST (bottom-left), and DYN (bottom-right) model on the HIT test set.

Most of the data points plotted in Figure 6.12 are in regions of the flow where no flame front is present and using $\overline{c'^2}$ for SGS closure is not important. The critical zones for $\overline{c'^2}$ models are located in the reaction zone of the flame, identified as the regions where the laminar Pfitzner reaction rate is greater than 10% of its maximum value:

$$\mathcal{Z} = \{\overline{c} : \dot{\omega}_F(\overline{c}) \geq 0.1 \times \max_{0 \leq c \leq 1} (\dot{\omega}_F(c))\} . \tag{6.22}$$

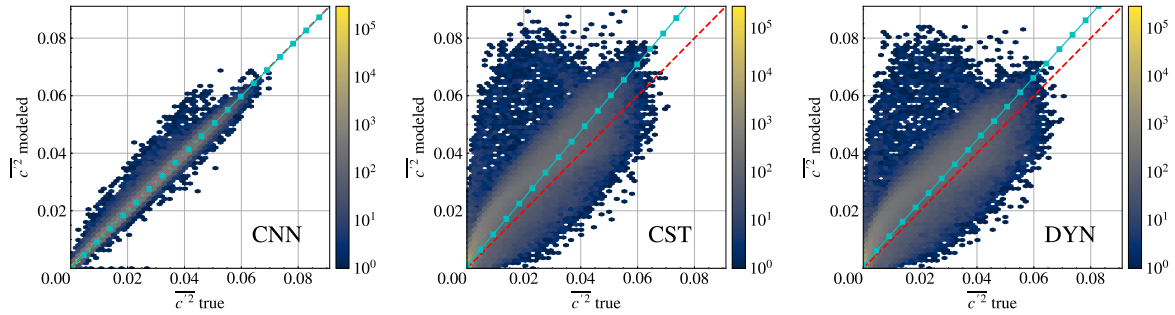
Hexplots conditioned on $\overline{c} \in \mathcal{Z}$ are plotted in Figure 6.13, and error values are also reported in Table 6.3. It is interesting to note that the CNN performs even better in these critical regions, and is very close to a perfect model. On the other hand, the performances of the gradient models are significantly worse. Error values computed using a normalized mean square metric:

$$\epsilon = \frac{\sum (y_{model} - y_{true})^2}{\sum y_{true}^2} \tag{6.23}$$

are reported in Table 6.3 and confirm the excellent *a priori* performance of the CNN on the HIT test set.

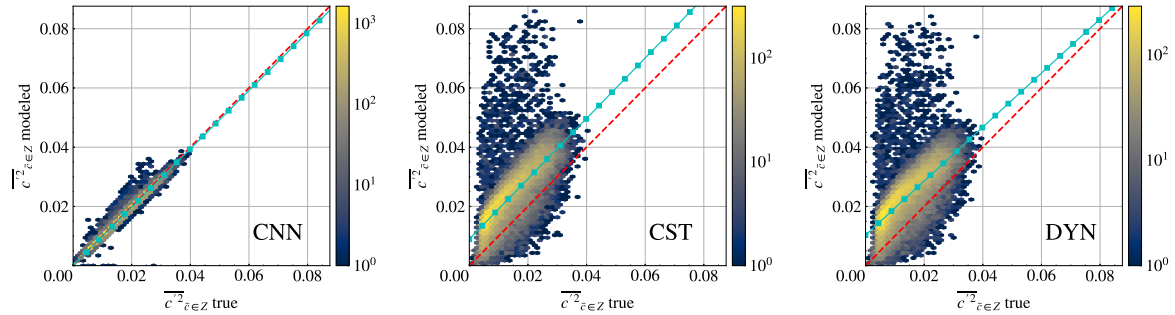
Model	ϵ	$\epsilon, \overline{c} \in \mathcal{Z}$
CNN	0.060	0.054
CST	0.417	0.733
DYN	0.398	0.730

Table 6.3: HIT test set error values for $\overline{c'^2}$.



(**a**) CNN (**b**) CST (**c**) DYN

Figure 6.12: Hexbin plots of model predictions against ground truth: (**a**) CNN model, (**b**) CST model, (**c**) DYN model. Data points are gathered in hexagonal bins and colored according to the number of data points they contain. The red line $y = x$ represents a perfect model. Cyan lines represent linear trends of the model.



(**a**) CNN (**b**) CST (**c**) DYN

Figure 6.13: Hexbin plots of model predictions against ground truth conditioned on $\overline{c} \in \mathcal{Z}$: (**a**) CNN model, (**b**) CST model, (**c**) DYN model.

6.4.2 Evaluation on the R2 generalization set

The various models are now assessed on the 5 snapshots of R2 composing the generalization set. For the DYN model, averaging is performed in the spanwise direction of the burner.

Figure 6.14 shows a sample slice of the $\widetilde{c}$ field in a filtered snapshot of R2 and the predictions of the CNN, CST, and DYN model for the corresponding SGS variance. Once again, the value $C_s = 0.18$ is retained for the CST model. All models reproduce qualitatively the evolution of the filtered DNS $\overline{c'^2}$. The hexplots of Figure 6.15 and the error values reported in Table 6.4 reveal that the CNN performs the best out of all the models, which is a demonstration of its ability to transfer its predictive power from the HIT flame to the R2 flame. However, it is naturally less accurate than on the test of the HIT, which contained samples from the same distribution as its training set. In comparison, the CST and DYN models show zero bias but a higher variance. Unlike in the HIT, all the models perform similarly when evaluated on the whole domain or only the reaction zone $\mathcal{Z}$, as indicated by the error values in Table 6.4.

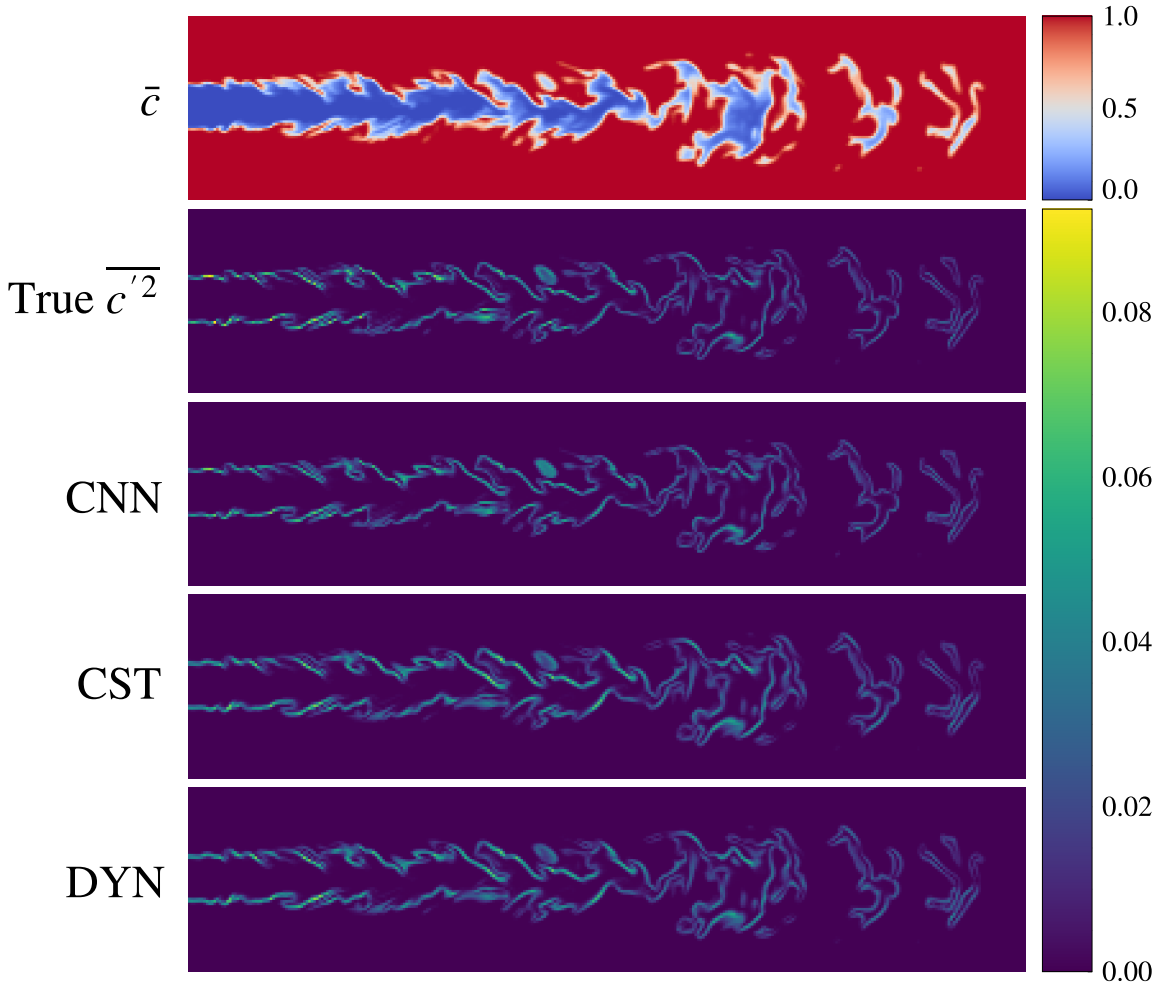
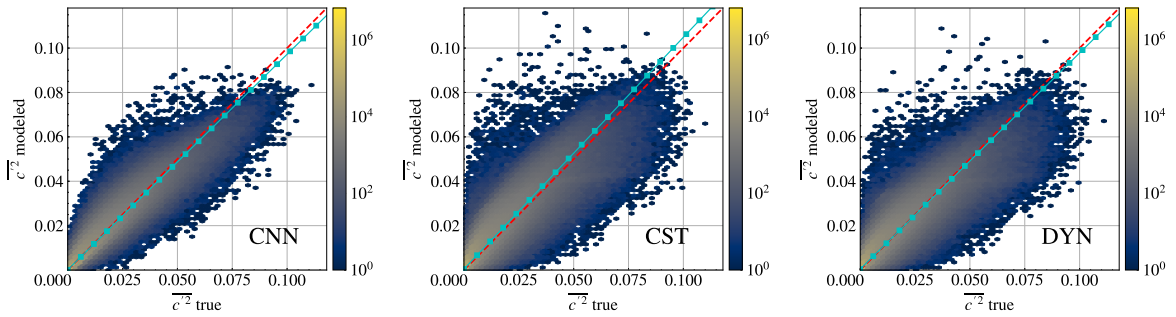


Figure 6.14: Sample R2 slices of $\bar{c}$, true $\overline{c'^2}$, CNN, CST, and DYN model predictions.



(**a**) CNN (**b**) CST (**c**) DYN

Figure 6.15: Hexbin plots of model predictions against ground truth on the R2 generalization set: (**a**) CNN model, (**b**) CST model, (**c**) DYN model.

Model	ϵ	$\epsilon, \overline{c} \in \mathcal{Z}$
CNN	0.241	0.214
CST	0.371	0.331
DYN	0.358	0.322

Table 6.4: R2 error values for $\overline{c'^2}$.

For a sample snapshot, the spatial distribution of model predictions is investigated by plotting $\overline{c'^2}$ values averaged on transverse slices in Figure 6.16. The CNN recovers the filtered DNS values nearly perfectly, while the algebraic models consistently overestimate the variance.

This *a priori* evaluation of $\overline{c'^2}$ models has shown that the CNN trained on the HIT configuration was able to generalize accurately to the R2 flame and outperform constant coefficient and dynamic algebraic models.

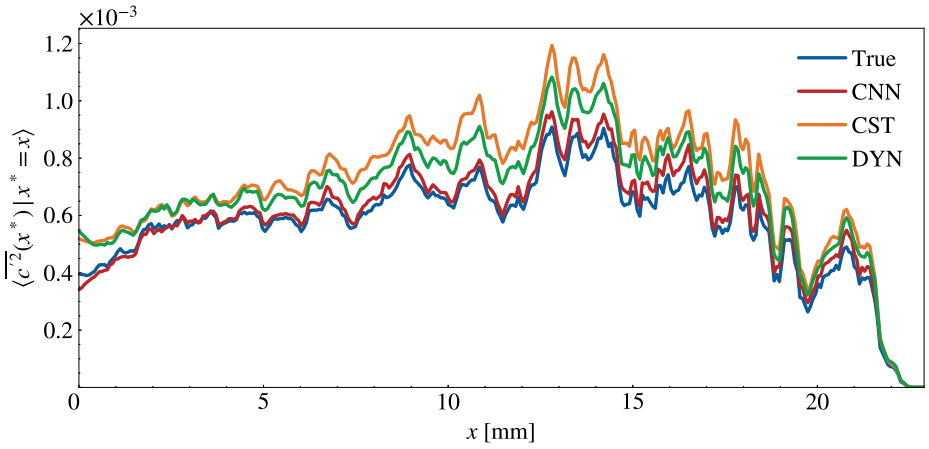


Figure 6.16: Evolution of transverse averages of the true $\overline{c'^2}$ and model predictions along the streamwise direction x.

6.4.3 Discussion on the conditions for generalization

Between the HIT training/test sets and the R2 generalization set, three key ratios are conserved: u'/s_L, $l_t/\delta_{L,c}$, and $\Delta/\delta_{L,c}$. It is important to understand the sensitivity of the generalization of the CNN to these ratios, as they could place strict limits on the practical applicability of the model.

The influence of u'/s_L and $l_t/\delta_{L,c}$ is analyzed by training a second CNN on the same HIT configuration with halved initial values of u'/s_L, $l_t/\delta_{L,c}$, leading to lesser wrinkling in the turbulent flame front. When evaluated on R2, this second CNN had an error $\epsilon = 0.301$, which is 30% higher than the reference CNN. Figure 6.17 shows that the effect of training on a weaker HIT is noticeable in the first 4 mm of the flame near the inlet. The second CNN underpredicts the SGS variance compared with the reference CNN and the filtered DNS. This is coherent with the fact that it is trained on a weaker HIT, which contains lower $\overline{c'^2}$

values than the original configuration. This brief study seems to indicate that the choice of u'/s_L, $l_t/\delta_{L,c}$ is indeed impactful on the generalization accuracy of the CNN, but more extensive work should be performed to fully understand the effect of these ratios.

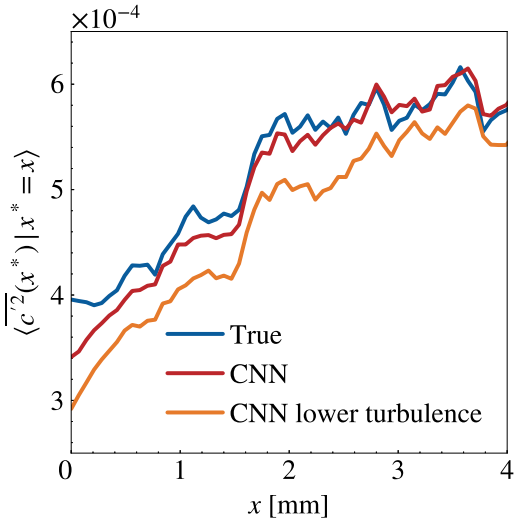


Figure 6.17: Evolution of transverse averages of the true $\overline{c'^2}$ and both CNN predictions in the first 4 mm of the R2 flame.

The restriction to relying on a single value of $\Delta/\delta_{L,c}$ can be relaxed by training the CNN on a range of filter size values. An instance of the CNN was trained on a dataset comprising the collection of training HIT snapshots, which were filtered at $\Delta/\delta_{L,c} = 0.8$, 1.2 and 1.6, for a total of 126 snapshots with 3 separate filter sizes. It was then tested on R2 for the same filter sizes, as well as unseen values $\Delta/\delta_{L,c} = 1$, 1.4, and 2. Figure 6.18 shows the evolution of the error made by the CNN and DYN models with $\Delta/\delta_{L,c}$. When $\Delta/\delta_{L,c}$ increases, the performance of the DYN model deteriorates as the SGS modeling task becomes more difficult. In contrast, the error made by the CNN is stable across all filter sizes, including unseen ones. This suggests that with proper training, the CNN can be accurate on a range of $\Delta/\delta_{L,c}$ values instead of a single one. Note that the error made by this CNN on $\Delta/\delta_{L,c} = 0.8$ is higher than the one reported in Table 6.4 for the CNN solely trained on this filter size. This implies that a balance must be found between accuracy at a single filter size, and validity across a range of values. Interestingly, providing the $\Delta/\delta_{L,c}$ value as an additional input channel for the CNN did not improve the results. The rest of this work is presented with the original CNN trained with $\Delta/\delta_{L,c} = 0.8$.

6.5 *A priori* evaluation of the PB-CNN model

6.5.1 Evaluation on the HIT test set

In the previous section, the CNN was shown to be able to learn a model for $\overline{c'^2}$, which is accurate on the test and generalization configurations. In the following, modeled values of $\overline{c'^2}$ are incorporated in the presumed beta PDF approach detailed in Section 6.1 to form the

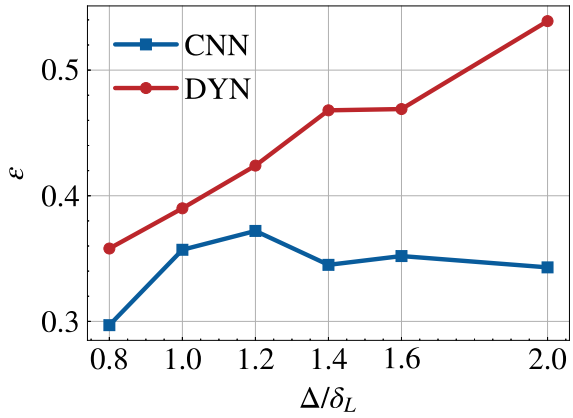


Figure 6.18: CNN and DYN model errors on R2 for various filter sizes.

PB-CNN model for $\overline{\dot{\omega}}_F$. Beta PDFs are known to sometimes be inaccurate models for the SGS distribution of c in premixed combustion [35, 75, 168]. Therefore, the presumed beta PDF approach needs to be justified independently from the model for $\overline{c'^2}$. For the HIT, Figure 6.19 shows the conditional median, 5% and 95% quantiles plotted against $\bar{c}$ for the true filtered reaction rate, and the Pfitzner beta PDF reaction rate with true $\overline{c'^2}$ values. The beta PDF appears to be a good approximation for the true SGS PDF, showing that the presumed beta PDF assumption is sound in the case of the HIT.

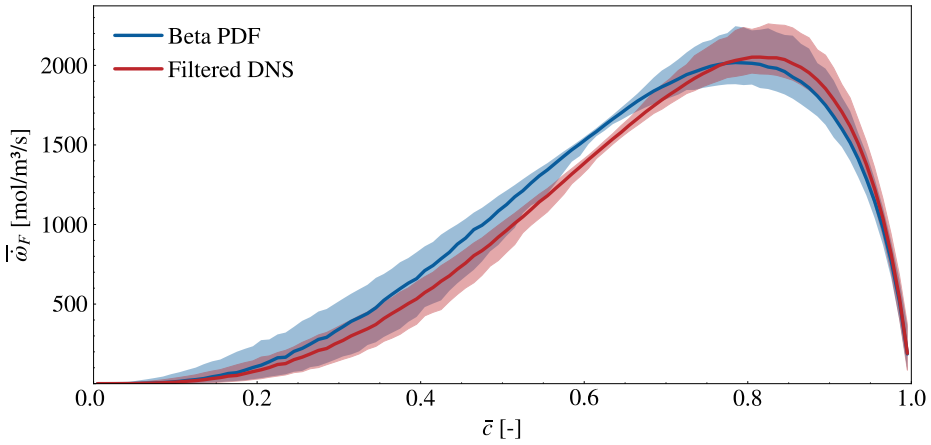


Figure 6.19: Distribution of the filtered reaction rate computed from a beta PDF assumption with true $\overline{c'^2}$ (blue) and the filtered DNS (red): conditional mean (solid line); 5% and 95% conditional quantiles (delimiting the shaded area). The statistics are aggregated over the HIT test set.

Given the accuracy of the CNN model, replacing the true $\overline{c'^2}$ values with the CNN predictions does not noticeably affect the shape of the beta PDF. The isolated effect of the CNN model on the beta PDF reaction rate appears minimal in Figure 6.20, where the beta PDF reaction rates with CNN values of $\overline{c'^2}$ are compared with beta PDF reaction rates with true values of $\overline{c'^2}$.

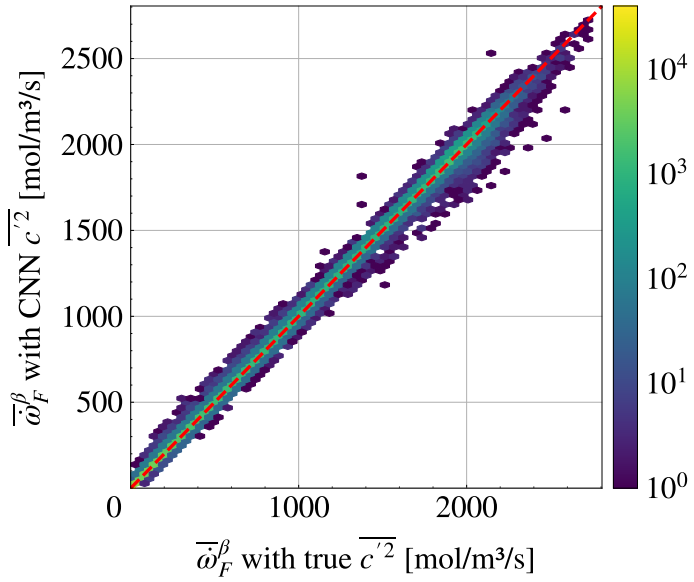
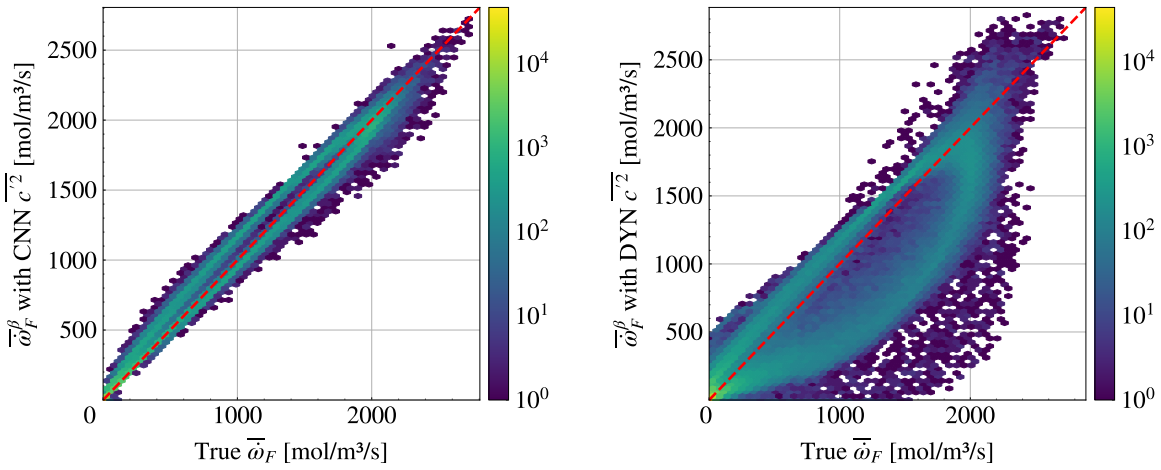


Figure 6.20: Hexbin plot of beta PDF reaction rates with CNN values of $\overline{c'^2}$ against beta PDF reaction rates with true values of $\overline{c'^2}$.

Finally, the combined effect of the beta PDF assumption and the $\overline{c'^2}$ model is shown in Figure 6.21. The high variance values predicted by the DYN model lead to underpredictions of the reaction rate when combined with the Pfitzner beta PDF closure. In contrast, using CNN predictions leads to a much closer fit to the true filtered reaction rate. Figure 6.22, where true and modeled reaction rates averaged on yz-planes are plotted against the streamwise x coordinate, illustrates how the quality of the CNN model for $\overline{c'^2}$ leads to a good very estimate of fuel consumption rates using a beta PDF assumption.



(**a**) CNN (**b**) DYN

Figure 6.21: Hexbin plots comparing true filtered reaction rates to beta PDF reaction rates using modeled $\overline{c'^2}$ values: (**a**) CNN model, (**b**) DYN model.

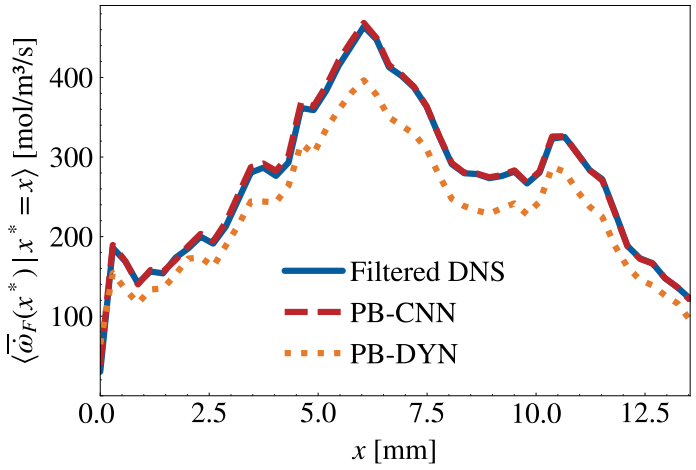


Figure 6.22: Evolution in the streamwise x direction of the true and modeled reaction rates averaged on yz-planes, for a sample in the HIT test set. PB-DYN denotes the Pfitzner beta PDF closure with the DYN model for $\overline{c'^2}$.

6.5.2 Evaluation on the R2 generalization set

The PB-CNN closure is finally evaluated on R2. From the R2 coarse fields of $\overline{c}$, beta PDF reaction rates are computed according to Equation 6.7 and using either true $\overline{c'^2}$ or predictions from the CNN. A reference 1D laminar flame matching the conditions of the R2 simulation is used to compute the prefactor $\Lambda = (\rho_f s_L)^2 Y_F^u$. The value for the Pfitzner model parameter $m = 4.7$ was chosen so that the total fuel consumption rate of the reference 1D flame is matched by the Pfitzner reaction rate based on the c field of the reference 1D flame. As a result, by construction, both reaction rates would lead to the same total fuel consumption rate on a planar laminar flame. Figure 6.23 shows the close match of the resulting Pfitzner reaction rate to the fuel reaction rate of the skeletal mechanism.

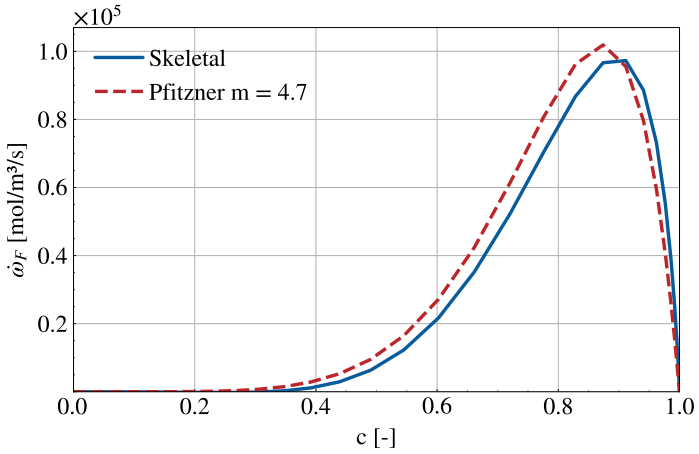


Figure 6.23: DNS fuel reaction rates for skeletal and Pfitzner chemistry vs. progress variable in a 1D laminar flame.

Figure 6.24 compares the PB-CNN reaction rates with the true filtered values of R2. For clarity, only the points where the filtered DNS reaction rate is greater than 10% of its maximum value are plotted. This corresponds to the same condition as Equation 6.22. Compared with the HIT flame, the scatter plot still exhibits low bias, with a large portion of the predictions concentrated around the reference linear trend, but it also has a higher variance, which is emphasized by the logarithmic scale of the color bar. This indicates that some local values of the reaction rate are not well predicted by the PB-CNN model.

One particular group of mispredicted points is located on the x-axis of the plot and corresponds to regions in the reaction zone where the PB-CNN model erroneously predicts a reaction rate of exactly 0. Analyzing the reason for this misprediction is insightful to understand the behavior of the model in difficult edge cases. This group of points is characterized in Figure 6.25 by the distribution of their values of the true filtered reaction rate, filtered progress variable, true SGS variance, and CNN predictions for the variance. It appears that these points are located in the postflame region, where $c \approx 1$ and the skeletal mechanism still predicts some chemical activity. On the other hand, the SGS variance is extremely small and estimated by the CNN to be 0, causing the beta PDF-based reaction rate to also be exactly 0. This highlights a limitation of comparing a simplified reaction rate formulation relying on the SGS variance against a skeletal mechanism. However, these mispredictions are a rare occurrence since this group of points only constitutes 0.1% of the points in the reaction zone.

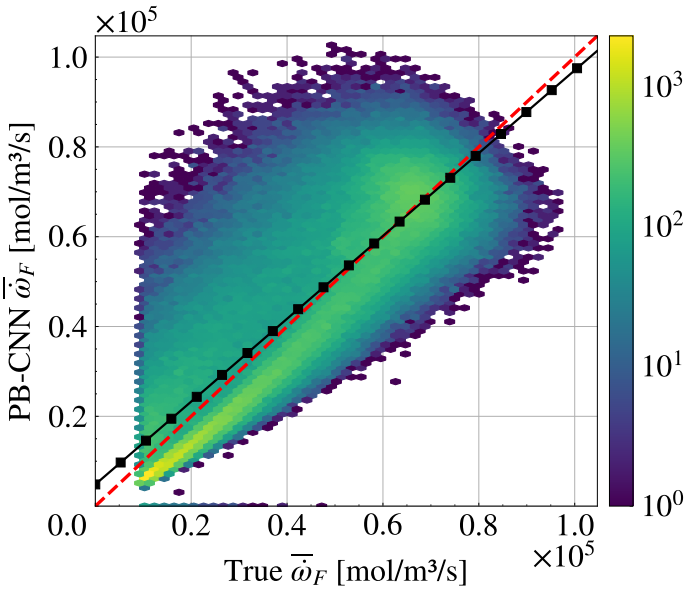


Figure 6.24: Hexbin plots comparing true filtered reaction rates with PB-CNN reaction rates for points in the reaction zone of the filtered DNS of R2. The black line represents the linear trend of the model.

Despite discrepancies in local predictions of the reaction rate, the PB-CNN model provides correct estimates of spatially averaged reaction rates. This is important as it shows that the model would be useful in LES to recover key integral quantities of the burner such as

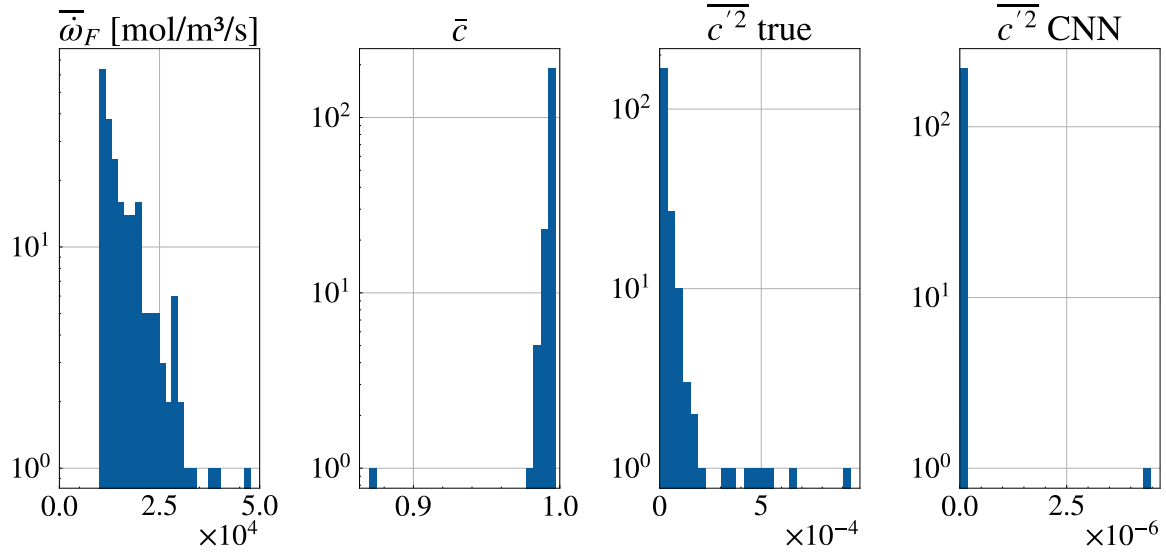


Figure 6.25: Histograms of $\overline{\dot{\omega}}_F$, $\bar{c}$, true $\overline{c'^2}$, and CNN $\overline{c'^2}$ for the points in the reaction zone where the PB-CNN reaction rate is exactly zero.

the mean length of the flame. This is shown in Figure 6.26, which compares the streamwise evolution of the filtered DNS and PB-CNN reaction rates averaged on transverse slices. The PB-CNN closure recovers the correct averaged reaction rate in all regions of the flame. This leads to a fuel consumption rate in the whole domain within 5% of the reference value. *A posteriori* simulations could be carried out in the future to assess if the PB-CNN reaction rate leads to an LES flame with the same length and total burning rate as the R2 DNS.

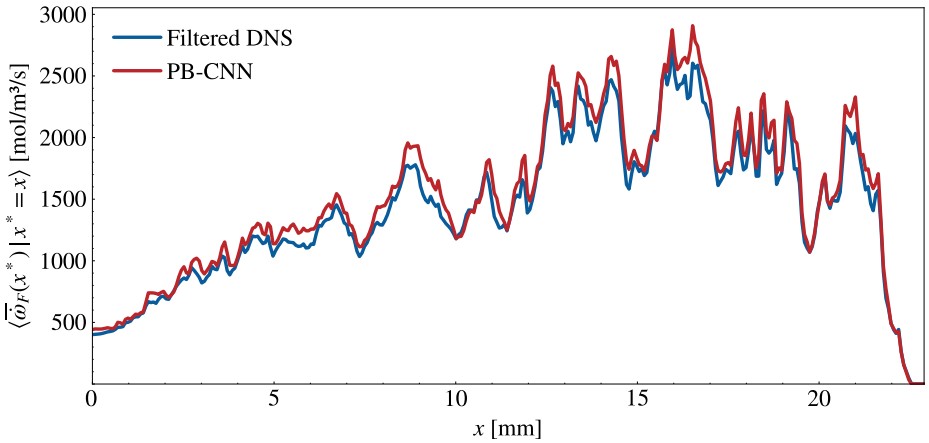


Figure 6.26: Evolution of transverse averages of skeletal and beta PDF reaction rates along the streamwise direction x.

6.6 Conclusion and perspectives

A deep convolutional neural network was trained to model the progress variable SGS variance $\overline{c'^2}$ on a statistically planar flame in homogeneous isotropic turbulence, and was shown to generalize accurately to the R2 jet flame in an *a priori* evaluation. Despite very different flow dynamics, thermophysical parameters, and chemistry, the conservation of the normalized filter size $\Delta/\delta_{L,c}$ and the turbulence–flame interaction ratios u'/s_L, $l_t/\delta_{L,c}$ were assumed to be key to the generalization of the CNN, and their influence on the results was discussed. This work indicates that CNNs can be trained on canonical simple cases and used in practical configurations. Future studies could focus on applying this methodology to different combustion models. The universal nature of the conditions for generalization that are exhibited here should also be investigated, as well as ways to integrate them into the machine learning model.

In an effort to combine computational efficiency and modeling accuracy, the PB-CNN closure for the mean LES fuel reaction rate was proposed. It combines the reaction source term formulation of Pfitzner et al. [269, 270], a presumed beta PDF assumption, and the CNN model for $\overline{c'^2}$ in an analytical formula for the reaction rate. It showed excellent *a priori* results on the R2 flame when compared with the burning rate produced by the original skeletal chemistry. In the future, an *a posteriori* evaluation would be a challenging test of the accuracy of the $\overline{c'^2}$ model given by the CNN and the PB-CNN closure for $\overline{\dot{\omega}}_F$. This formulation would be especially well suited to simulating large-scale premixed combustion configurations, such as explosions in venting chambers. The resulting overpressure can be accurately captured using simplified chemistry, but remains a challenge at scale due to insufficient mesh resolution, prohibitive computational costs, and strong sensitivity to turbulent reaction rates [291, 369]. Such applications are investigated in Chapter 8.

Part III

A posteriori evaluation

Chapter 7

AVBP-DL: hybrid coupling of a CFD solver to deep neural networks

Contents

This chapter presents the AVBP-DL coupling strategy which allows the AVBP solver to query predictions made by deep neural networks on-the-fly during an LES simulation. Coupling a massively parallel fluid solver with deep CNNs requires dedicated high performance computing efforts, as AVBP is still mostly bound to central processing units (CPUs) while neural networks operate most efficiently on GPUs. It is a mandatory step towards using deep learning for practical LES computations. The computational overhead of the coupling must remain competitive against other modeling alternatives. A global overview of the AVBP-DL coupling strategy is first provided, followed by details of its implementation, especially for the mesh localization and temporal loop. A performance and scalability study is then carried out to estimate the computational overhead of the coupling.

7.1 Global overview

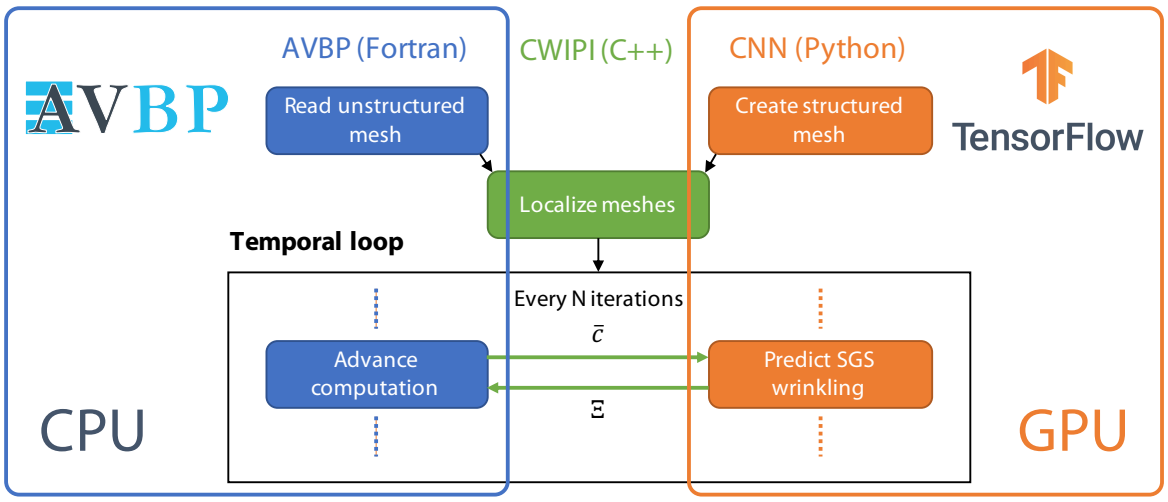


Figure 7.1: High-level overview of the AVBP-DL coupling strategy.

The AVBP-DL coupling strategy follows a Multiple Program Multiple Data (MPMD) approach which involves two distinct solvers. AVBP is the fluid solver that handles time advancement of the LES reactive Navier-Stokes equations. A DL solver is run in parallel and can pilot deep learning models that may replace any subgrid-scale model needed by AVBP. Both AVBP and the DL solver involve a large number of processes that perform distributed computation. Parallel asynchronous communications are used to exchange inputs and outputs of the deep learning model between AVBP and the DL solver.

This chapter will focus on CNN models for turbulent combustion modeling, although AVBP-DL can handle any generic data-driven model. CNNs introduce some specificities in the coupling strategy. Because they operate on voxels, they cannot immediately process AVBP data which is unstructured and discretized with arbitrary finite element types. Additionally, complex geometries featuring curved boundaries and non-convexities require special treatment. These specificities motivate the use of the CWIPI (Coupling With Interpolation Parallel Interface) library [303] to manage communications between the solvers. CWIPI enables efficient coupling between parallel solvers through MPI communications. It can handle non-matching element types and non-coinciding domains by interpolating data between the domains.

AVBP-DL greatly benefits from the emergence of hybrid CPU/GPU nodes in modern supercomputing clusters. Hybrid architectures have come to dominate supercomputing rankings[1] and enable deep learning models that can efficiently use GPU acceleration. In a coupled simulation, processes are split between AVBP and the DL solver, with typically many more AVBP processes than DL processes. Each DL process controls a copy of the deep learning model via Python code.

[1] In the June 2022 TOP 500 list, the top-ranking clusters in France, Europe and worldwide were all equipped with hybrid nodes.

In this chapter, AVBP-DL coupling is applied to the *a posteriori* investigation of a CNN model for the SGS wrinkling factor in the Masri vented obstructed gas explosion. Numerical and physical details of the configuration and of the model are presented in Chapter 8. The CNN model takes a progress variable field c as input and returns a wrinkling factor field Ξ. The 3D c and Ξ fields are the only data exchanged at every step. This chapter will focus on detailing the AVBP-DL coupling strategy outlined in Figure 7.1 and measure the computational overhead compared to a non-coupled AVBP simulation. The physical analysis of the results of the simulation are found in Chapter 8.

7.2 Detailed overview

7.2.1 Mesh localization

AVBP and the DL solver can operate on non-coinciding meshes. AVBP regions not covered by the DL mesh may correspond to parts of the domain where the DL model is not needed. In the Masri configuration, this corresponds to the plenum (Figure 8.1) where turbulent combustion does not affect the quantity of interest of the simulation. The DL mesh therefore only covers the extent of the obstructed chamber. On the other hand, the DL mesh may extend outside the AVBP mesh if the domain geometry does not match a structured grid. Since CNNs work on structured arrays, the DL mesh is defined as the union of disjoint 3D rectangular domains, generally forming one contiguous rectangular box. In the Masri configuration, the DL domain is therefore defined as the bounding box of the main chamber, including obstacles that are excluded from the AVBP mesh as they are outside the fluid domain. The AVBP unstructured mesh is composed of 28 million tetrahedral elements and 5.1 million nodes, with a typical mesh size $h = 0.5\,\text{mm}$. The DL structured mesh is composed of $500 \times 100 \times 100 = 5$ million hexahedral elements with the same mesh size h.

Each solver uses its own domain decomposition method. In AVBP, the ParMETIS [156] library performs parallel partitioning of the unstructured mesh. Since the DL domain is a rectangular box, DL partitions are defined using a recursive coordinate bisection algorithm. Figure 7.2 shows a y-normal slice offset 1 cm from the centerplane of the Masri configuration, with the partitioning of the AVBP and DL meshes for a total of 1280 processes (1264 AVBP/16 DL).

Since both solvers can have non-coinciding domains which are discretized using different element types and domain decomposition algorithms, linear interpolation is applied to the data that is exchanged between the solvers to convert between its representation on both meshes. On either mesh, points that are contained within the extent of the distant mesh are called **localized** points. They belong to regions where both meshes overlap. Meshes remain static during the whole simulation, so interpolation weights for localized points can be precomputed and saved before the temporal loop. This is the **localization** step. This parallel procedure follows the inter-code communication scheme algorithm detailed in Duchaine et al. [78].

Figure 7.2: AVBP and DL partitioning on a y-normal slice at $y = 1\,\text{cm}$. Top: AVBP partitioning (the white strips correspond to the 3 obstacle grids) with 1264 processes. Bottom: DL partitioning with 16 processes.

7.2.2 Temporal loop

Figure 7.3 breaks down the timeline of an AVBP-DL temporal iteration. The placement of MPI communications is optimized to hide as much of the computational overhead as possible.

The iteration starts with the computation of the progress variable c in AVBP. It is then sent to the DL solver via the CWIPI communication graph built in the localization step. CWIPI performs data interpolation and parallel exchanges from the AVBP partitions to the appropriate DL partitions. The exchange is a non-blocking synchronous send operation (`MPI_ISsend`). Non-blocking means that the operation returns immediately and allows AVBP computations to continue without waiting for reception on the DL side. Synchronous indicates that the waiting operation that checks for completion `Wait_ISsend` will return only when the data has started to be received. The non-blocking receive operation for Ξ (`MPI_Irecv`) is called right after. This buffers the reception of Ξ, allowing it to be received by MPI as soon as possible.

For the DL solver, c is received by a matching receive operation. Some processing steps are needed before running the CNN inference. Non-localized DL points did not receive c values and are filled in using nearest neighbor interpolation from the set of localized points. These points correspond to the obstacles inside the Masri chamber. Their treatment is important because the DL solver takes an input domain which includes these points. Alternatives to nearest neighbor interpolation could be trilinear interpolation or filling in a constant default value. Trilinear interpolation was found to be computationally expensive and had minimal impact on the resulting field. Filling in a constant value would introduce sharp transitions in the c field at the edge of the obstacles which would induce errors in the CNN predictions. They are avoided with nearest neighbor interpolation, although discontinuities may still exist inside the obstacles. Their effect on the physical results is discussed in Section 8.8.

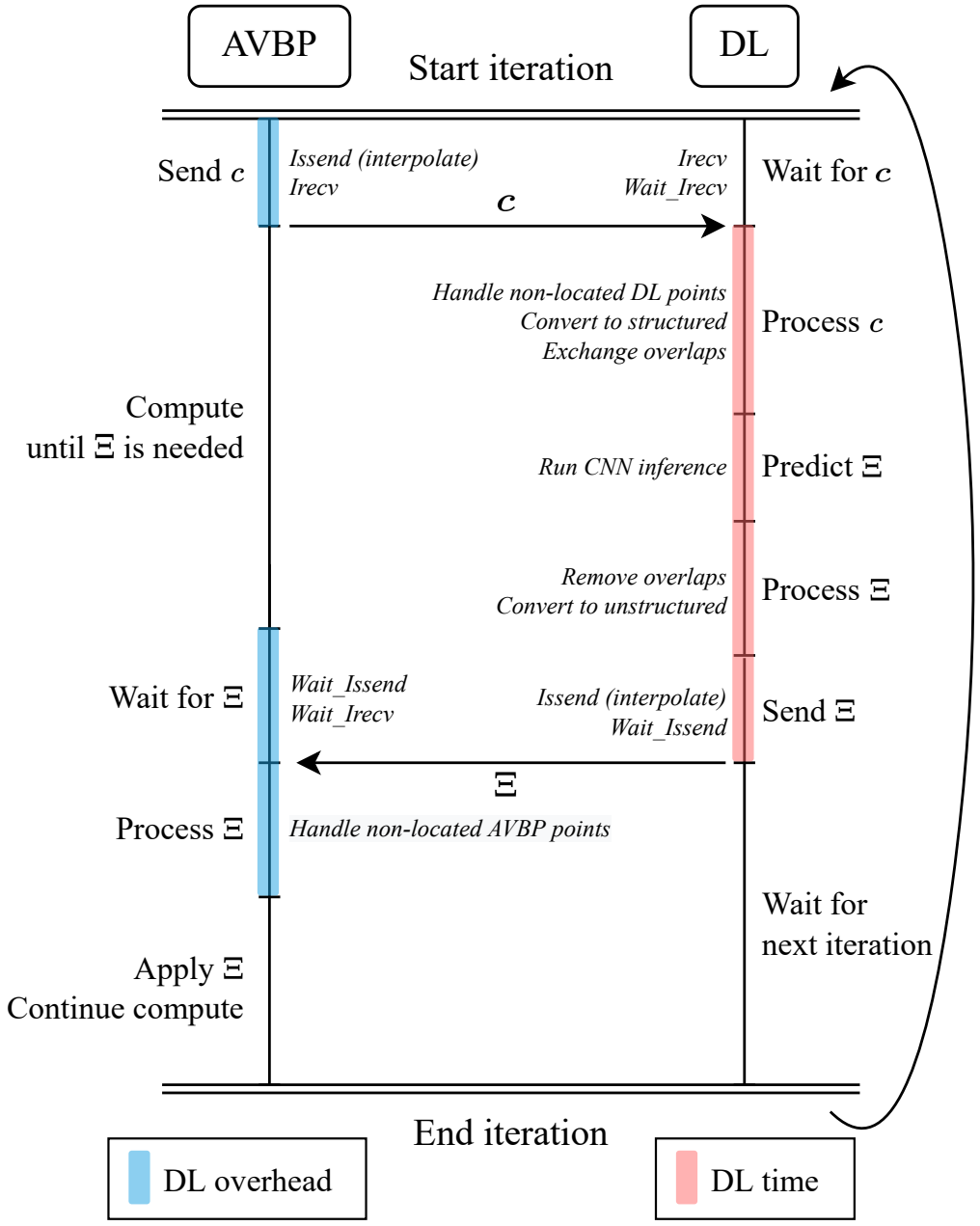


Figure 7.3: Timeline of an AVBP-DL temporal iteration.

Because DL partitions are disjoint, CNN predictions on either side of partition boundaries may differ. Indeed, predictions on either side of the boundary are generated by different input fields. Existing work has addressed this issue by implementing the exchange of partition margins before every convolution operation of a 3D U-Net to perform distributed training [143]. This work pursues a simpler approach illustrated in Figure 7.4. 3D rectangular **overlaps** are exchanged between neighboring partitions via MPI before the inference of the CNN. Overlaps have a size s in the normal direction to the partition boundary, and extend to the bounds of the domain in the other directions. If s is larger than half of the effective

receptive field of the CNN, Figure 7.4 shows that the overlap zone encompasses all the input points that can influence CNN predictions at the edge of the partition. This approach was preferred as the ERF of the U-Net architecture of the model is not too large. Given the size of the ERF, $s = 12$ is a suitable overlap size for the Masri simulation.

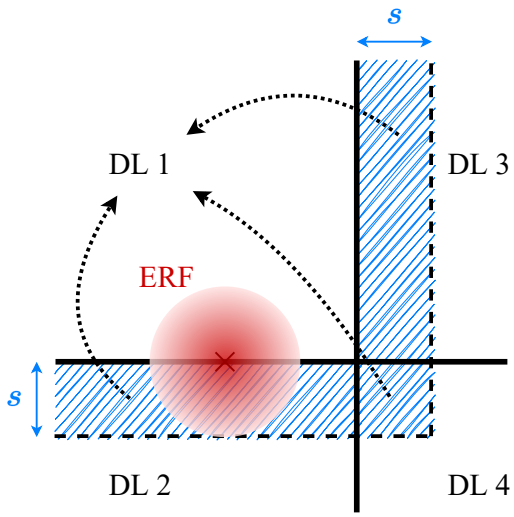


Figure 7.4: Exchange of DL partition overlaps. In this example, only exchanges to DL process 1 are drawn. Overlaps from DL processes 2, 3, 4 are communicated to process 1. The ERF of the CNN at the partition boundary is also pictured.

Figure 7.5 illustrates the impact of overlaps on CNN predictions. Without overlaps, predictions in the flame front near the boundaries are generally lower than when sufficient overlap is given to account for the whole field. Discrepancies would lead to discontinuities across DL partition boundaries up to values of 1, which is the order of magnitude of wrinkling values. This underlines the necessity to use an overlap to preserve the integrity of the CNN predictions near partition boundaries.

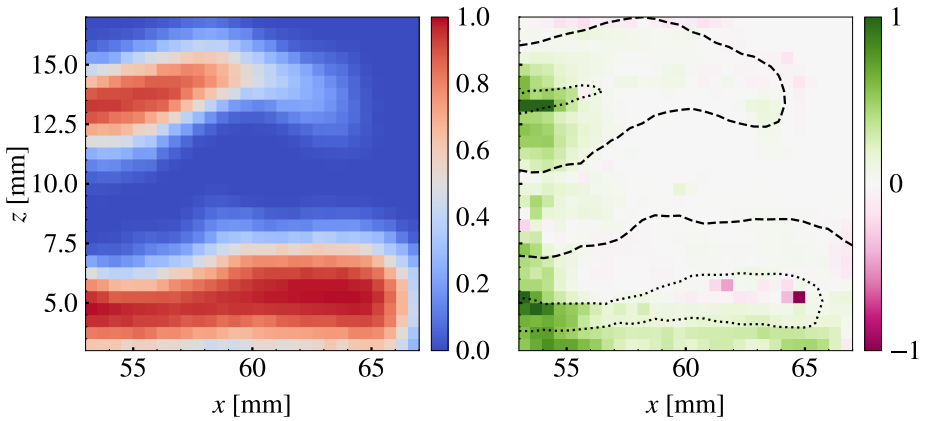


Figure 7.5: Effect of overlaps on CNN predictions. Left: sample cropped progress variable slice from the Masri simulation. Right: difference between CNN prediction with overlaps ($s = 12$) and without overlaps, with progress variable iso-lines $c = 0.1$ (dashed), $c = 0.9$ (dotted).

In each DL partition, the progress variable field produced by these processing steps is mapped to a SGS wrinkling factor field Ξ by an instance of the trained CNN. Overlaps are then removed, and Ξ is sent by a non-blocking synchronous send operation to AVBP.

The matching `Wait_Irecv` operation in AVBP is placed just before Ξ is needed for the efficiency. This minimizes the waiting time for AVBP, which can be zero if intermediate fluid computations take longer than the data processing and inference of the DL solver. The Ξ value of non-localized AVBP points is set to one.

Compared to a non-coupled AVBP simulation, the **DL overhead** is defined as the duration of all the additional operations in the AVBP timeline (colored in blue in Figure 7.3). This computational overhead should remain as low as possible, and it will be used to assess the performance of the coupling. It is controlled by the **DL time**, defined as the time taken by the DL solver to output Ξ after it has received c and colored in red, and by the duration of the communications between the two solvers.

7.3 Performance and scalability

7.3.1 Performance benchmark setup

AVBP-DL is benchmarked on the Jean Zay supercomputing cluster[2]. As of June 2022, Jean Zay has a cumulative peak performance of 36.85 petaflops per second and is ranked 114th on the TOP 500 list. Computations are carried out on its accelerated hybrid partition whose hardware and software specifications are listed in Table 7.1.

CPU processor	Intel Cascade Lake 6248, 2.5 GHz (2 per node)
Cores per node	40
GPU model	Nvidia Tesla V100 SXM2 16GB
GPUs per node	4
Communication architecture	Intel Omni-Path 100 GB/s
Tensorflow version	2.6.0
CUDA version	11.2
cuDNN version	8.1.1
MPI library	Open MPI 4.1.1

Table 7.1: Hardware and software specifications for the performance benchmark.

Hyperthreading is disabled, so there is a one-to-one relationship between computing cores (hardware) and MPI processes (software). 40 processes are launched on each node and can either be assigned to AVBP or to the DL solver. DL processes each launch one CNN instance on one GPU, and each GPU executes at most one CNN instance. Four process distributions are benchmarked (Table 7.2). D1, D2, D4 use respectively 1, 2, and 4 GPUs per node. D0.5 uses half the number of DL processes as D4. In this case, the DL processes are all grouped on a few nodes which run 4 DL processes each. The rest of the nodes only contain AVBP processes.

[2] http://www.idris.fr/eng/jean-zay/jean-zay-presentation-eng.html

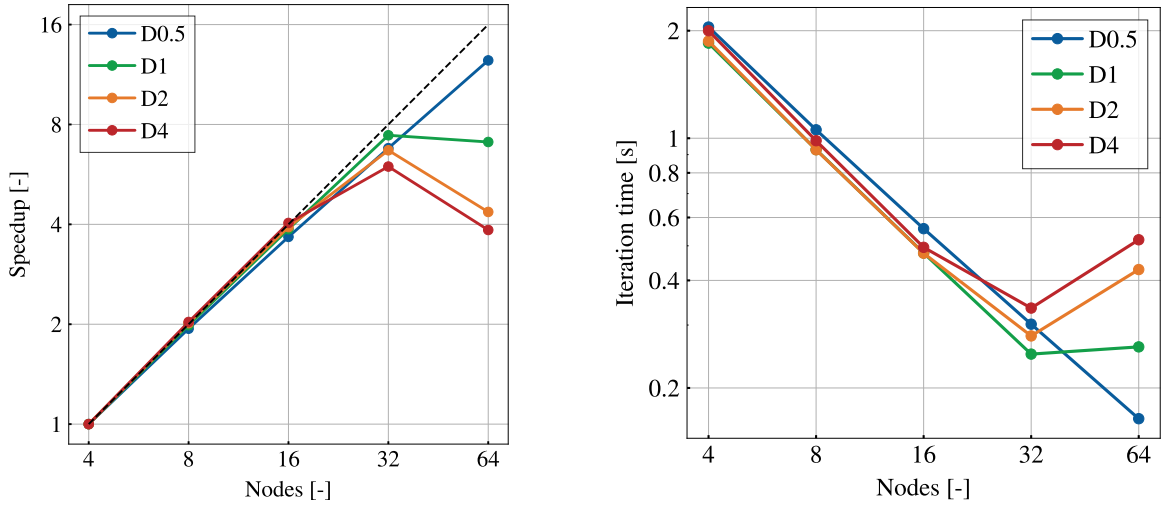
Distribution name	D0.5	D1	D2	D4
Number of AVBP processes	$39.5n$	$39n$	$38n$	$36n$
Number of DL processes	$0.5n$	n	$2n$	$4n$

Table 7.2: Process distributions for AVBP-DL benchmarking.

The scalability of AVBP-DL is assessed by a strong scaling study, where the problem size is kept constant while the number of resources is increased. The base amount of computing resources is $n = 4$ hybrid nodes of the Jean Zay cluster and is increased to 8, 16, 32, and 64 nodes. For each number of nodes and for each process distribution, five identical AVBP-DL simulation of the Masri configuration are run for 100 iterations, and the DL solver is queried at each iteration to provide Ξ. Timers for all the steps of the AVBP-DL temporal loop are aggregated over all the iterations and all the processes of the five executions. The duration of the initialization (before the temporal iterations) and termination (after the temporal iterations) of the computation is not included in this study.

7.3.2 Speedup

Speedup is defined as the acceleration relative to the base case due to the increase in resources. Perfect scalability is characterized by a linear relation between speedup and the number of nodes. Figure 7.6 reports the speedup and the duration of the temporal iteration. All distributions achieve perfect scalability until 16 nodes. After 32 nodes, they all see a clear drop-off except for D0.5. The more GPUs used, the sharper the drop-off, and Figure 7.6b shows that this loss of scalability is not compensated by faster iteration times at low node counts. D1, D2, D4 take longer to perform an iteration with 64 nodes than with 32 nodes. As a consequence, practical AVBP-DL computations in Chapter 8 are carried out with distribution D0.5.



(**a**) Speedup

(**b**) Iteration time

Figure 7.6: Strong scaling results.

7.3.3 DL overhead

To understand the scalability results of D4 and why D0.5 behaves otherwise, focus is placed on the DL overhead. Since both D4 and D0.5 use roughly the same amount of AVBP processes, the difference in scalability must come from the coupling with the DL solver and not from the intrinsic behavior of AVBP.

In practical computations, the DL model does not need to be queried at every iteration. This is also true for the CDSM model and was explained in Section 2.8: the global timestep is generally imposed by the convective scheme via the CFL number, and this acoustic time is typically orders of magnitude smaller than the characteristic evolution time of the wrinkling field. DL overhead values for this benchmark should therefore not be interpreted as the true overhead encountered in practice. Instead, $\theta(N)$ is defined as the DL overhead divided by the full iteration time when querying the DL model every N iterations. When $N > 1$, the $N-1$ iterations between every exchange do not incur any overhead as the previous field of Ξ is simply reused. The DL overhead measured in this benchmark corresponds to $\theta(1)$. $\theta(N)$ is then deduced via:

$$\theta(N) = \frac{\theta(1)}{N[1-\theta(1)]}\,. \tag{7.1}$$

In the Masri computation, the timestep is approximately $\Delta t = 5 \times 10^{-8}\,\mathrm{s}$, so with a characteristic flow velocity $||\boldsymbol{u}|| = 100\,\mathrm{m\,s^{-1}}$ at the peak flame acceleration and a cell size $h = 0.5\,\mathrm{mm}$, it takes about $h/(||\boldsymbol{u}||\Delta t) = 100$ iterations for the flame to move across one cell. $N = 100$ will therefore be used to quantify the practical DL overhead in this simulation. It is also the order of magnitude of the wrinkling model update frequency typically used for the CDSM model. Volpiani et al. [377] benchmarked the overhead of the CDSM model with $N = 100$ and 250. For $N = 100$, they report an overhead of 24.5% compared to the static Charlette model. The CDSM overhead is due to the many expensive test-filtering operations used by the model. Note that two elements may bias the comparison of their CDSM overhead with the CNN overhead. First, the details of the CPU hardware they use are unclear, so the ratio between CPU processing power and the number of mesh nodes per core cannot be compared with the values of this study. Second, the base case in Volpiani et al. incurs the cost of the static Charlette model, whereas in AVBP-DL, the base case does not require any computation for the efficiency model as the previous efficiency field is simply reused. Still, the values reported by Volpiani et al. are good orders of magnitude of the computational overhead of the CDSM model in practical LES simulations.

Table 7.3 reports $\theta(1)$ and $\theta(100)$ averaged among all AVBP processes and all iterations, for each number of nodes. For D4, $\theta(1)$ is very low until 32 nodes, then climbs to very high values. This is linked to the speedup drop-off observed in Figure 7.6. On the other hand, $\theta(1)$ values for D0.5 steadily increase. In general, $\theta(100)$ takes very low values, remaining below 1% except for the 64 node case of D4 where it reaches 2.9%. The overhead incurred by the use of a deep CNN as a wrinkling model is therefore negligible in practical computations. For the node configuration used in the Masri simulations of Chapter 8, which is 32 nodes and distribution D0.5, a direct comparison between the CDSM and AVBP-DL iteration times is done. When exchanging every iteration, the average iteration time AVBP-DL is 0.28 s compared to 3.96 s for CDSM. AVBP-DL therefore leads to a 14-fold speedup of the

turbulent combustion model compared to CDSM. The overall speedup of the computation is inversely proportional to N per Equation 7.1. For $N = 100$, a speedup of 16% would therefore be expected over the whole simulation.

Nodes	4	8	16	32	64
$\theta(1)$ [%]	0.4	0.5	0.7	24	74.3
$\theta(100)$ [%]	≈ 0	≈ 0	≈ 0	0.3	2.9

Nodes	4	8	16	32	64
$\theta(1)$ [%]	11.1	12.7	15.8	22.8	23.1
$\theta(100)$ [%]	0.1	0.1	0.2	0.3	0.3

Table 7.3: Fraction of DL overhead in temporal iteration time for D4 (top) and D0.5 (bottom).

Beyond mean values, Figure 7.7 provides a full description of the distribution of DL overhead values. For D4, it is interesting to note that the large increase in mean and median overhead at 32 nodes also brings about an increase in variability. For D0.5, the variability stays roughly constant. Data points outside of the box plot whiskers can be considered outliers [364]. This is the case for most minimum and maximum values and is especially noticeable for the maxima at 64 nodes. This suggests that stochastic events such as communication delays due to network bandwidth bottlenecks may play a large part in the DL overhead.

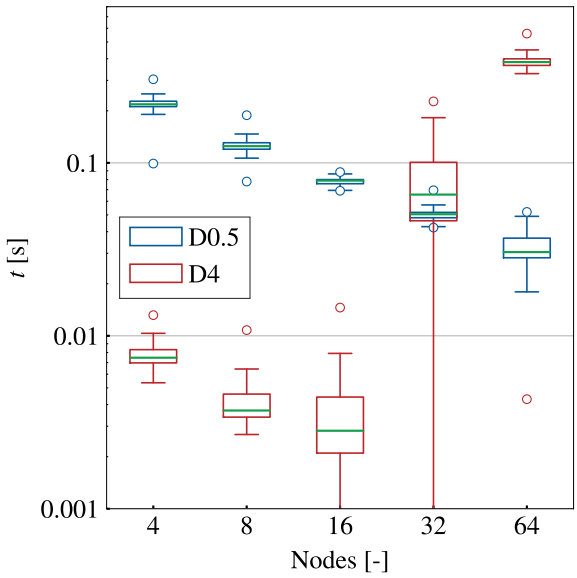


Figure 7.7: Box plots of DL overhead times for D4 and D0.5. Green lines indicate the median, boxes spread from the first quartile Q_1 to the third quartile Q_3, and whiskers extend from $Q_1 - 1.5(Q_3 - Q_1)$ to $Q_3 + 1.5(Q_3 - Q_1)$. Minimum and maximum values are shown as empty circles.

7.3.4 Understanding the DL overhead

To understand why D4 and D0.5 scale differently, the DL overhead and its components are aggregated among all processes and plotted for every iteration. Figure 7.8 shows that the DL overhead grows to very large values for D4 32 and 64 nodes. For D0.5, it decreases on average with the number of nodes, but shows intense fluctuations across iterations for 64 nodes compared to 32 nodes. Rarely, for 16 and 32 nodes, the DL overhead increases by a order of magnitude for a few consecutive iterations.

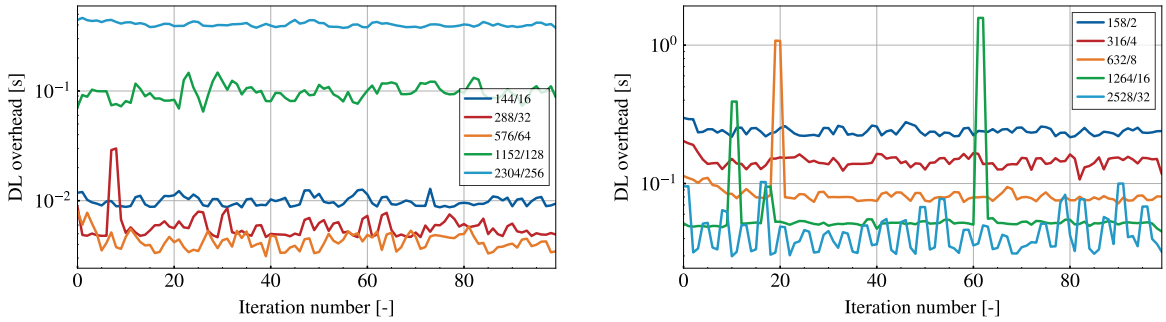
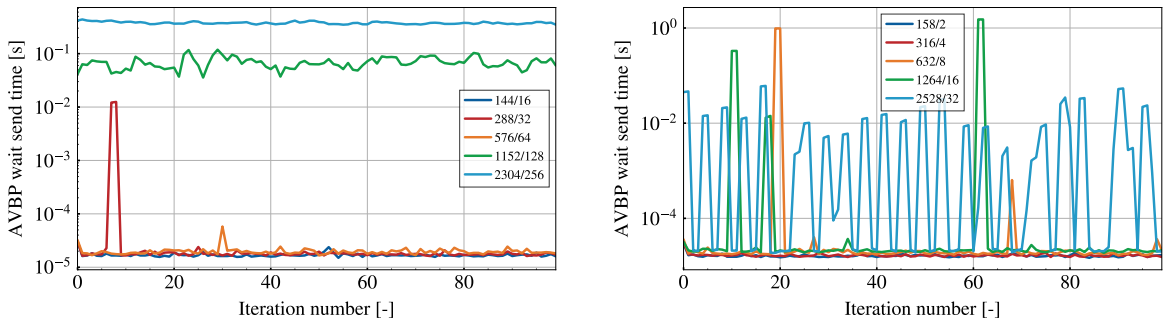


Figure 7.8: DL overhead per iteration for D4 (left) and D0.5 (right) (average over five executions of the maximum among all AVBP processes). Each configuration is labeled by its AVBP/DL process distribution.

According to Figure 7.3, the DL overhead is composed of four distinct steps: a send operation including interpolation to the DL mesh, a wait send operation, a wait receive operation, and a Ξ processing step. Timers for the send and processing operations are reported in Section B.1 of Appendix B and show good scalability.



(**a**) (**b**) D0.5

Figure 7.9: AVBP wait send time per iteration for D4 (left) and D0.5 (right) (average over five executions of the maximum among all AVBP processes).

Non-negligible wait send times indicate that AVBP communications have not begun to be received by the DL solver by the time AVBP has finished intermediate computations while waiting for Ξ. Figure 7.9 shows that this is the case for the D4 32 and 64 node cases, where large DL overheads are observed. Intermittent peaks for the D0.5 distribution also correlate with DL overhead increases in Figure 7.8. The full breakdown of the DL overhead is

available in Section B.3, where it appears that the wait send operation is responsible for most of the DL overhead in the D4 32 and 64 node cases. This clearly points to communication bottlenecks which are mitigated in D0.5 since the number of communications scales with $n_{AVBP} \times n_{DL}$ which is smaller in D0.5 than in D4. It would also explain why they would start to arise for D0.5 64 nodes, when $n_{AVBP} \times n_{DL}$ starts to become significant.

Non-negligible wait receive times may indicate that the DL solver is not ready to send Ξ, or that communication bottlenecks delay the exchange. In the first case, it may be caused by increased computational time from the DL solver, *i.e.* the DL time colored in red in Figure 7.3, or a delay in the start of the DL solver iteration. Figure 7.10 shows that the wait receive times for D4 behave like the wait send times, suggesting that once again communication bottlenecks are to blame. The situation is different for D0.5, where a clear speedup of the wait receive time is observed as the number of resources increases. The fluctuations seen for 16, 32, and 64 nodes are correlated with the wait send fluctuations, indicating that communication issues also affect the D0.5 wait receive times, but to a lesser extent than D4.

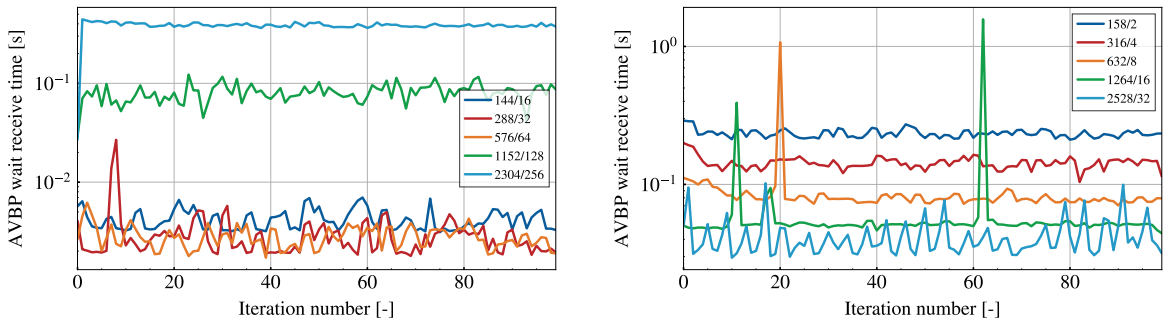


Figure 7.10: AVBP wait receive time per iteration for D4 (left) and D0.5 (right) (average over five executions of the maximum among all AVBP processes).

The DL time is plotted in Figure 7.11, and a breakdown its composition is also available in Section B.2. In all configurations, the DL time decreases as the number of nodes increases. Saturation is observed for D4 at large node count. In this case, it is dominated by the prediction time which does not scale well due to incompressible overhead from Tensorflow that is more visible for small partition sizes. Unlike the DL overhead, the DL time for high node count is still lower than for low node count. The DL time should therefore not be the main reason for high wait receive times in D4.

Finally, Figure 7.12 measures the time from the beginning of the full iteration to the DL receive operation. It indicates that there is no such delay for D4 at high node count, thus suggesting that its high wait receive times are due to communication bottlenecks.

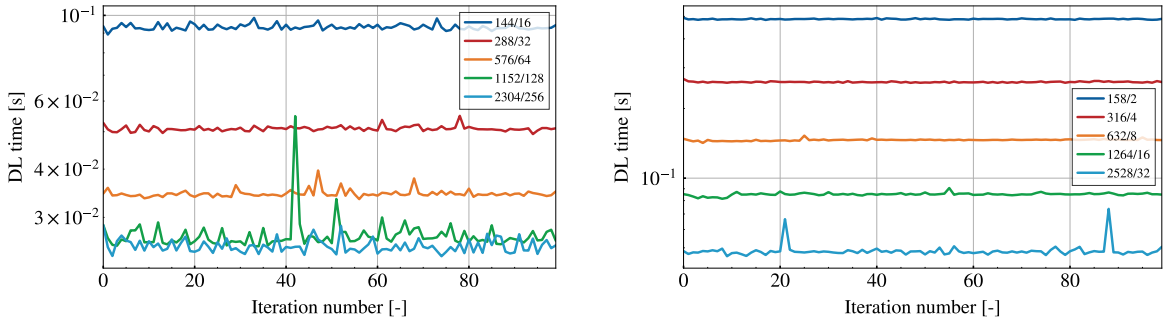


Figure 7.11: DL time per iteration for D4 (left) and D0.5 (right) (average over five executions of the maximum among all DL processes).

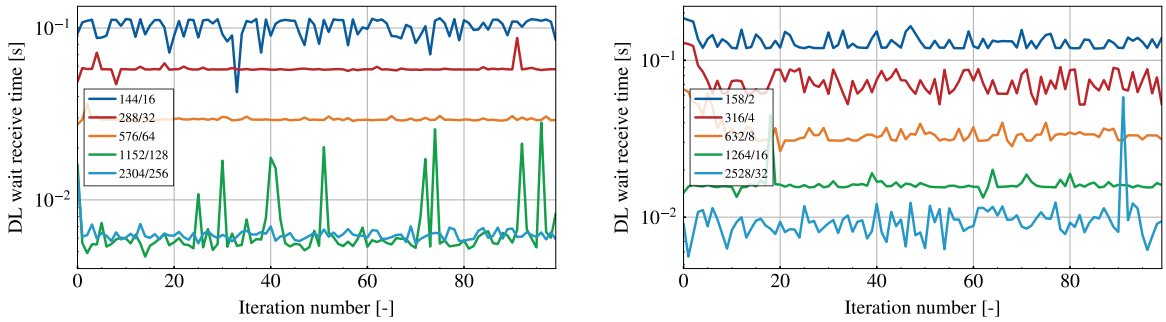


Figure 7.12: DL wait receive time per iteration for D4 (left) and D0.5 (right) (average over five executions of the maximum among all DL processes).

7.4 Conclusion and perspectives

AVBP-DL enables massively parallel coupling of AVBP with a deep neural network for on-the-fly DL predictions with virtually no computational overhead. It relies on the CWIPI library to accommodate the different data structures of AVBP and the DL model via mesh-to-mesh interpolation, and on MPI communications for decentralized data exchanges. An asynchronous coupling strategy is developed to hide the computational cost of the DL model. In practical computations, the overhead compared to a non-coupled AVBP simulation is estimated to be $\approx 1\%$, making this strategy viable for industrial LES simulations and improving on the performance of the CDSM model. This overhead is only attainable on hybrid nodes combining CPUs and GPUs, which are becoming the standard in new supercomputing clusters built today. A scalability study reveals that using less DL processes and thus less GPUs per AVBP process leads to better performance. An in-depth investigation of all the coupling steps indicates that an abundance of DL processes induces communication bottlenecks that slow down the computation, although the impact on the practical DL overhead is minimal. Analyzing execution traces with tools such as Extrae[3] could allow for a finer understanding of these issues.

[3] https://tools.bsc.es/extrae

Future work on AVBP-DL may focus on enabling more flexible neural network architectures. CNNs place constraints on the DL mesh which could need to be relaxed when dealing with complex geometries. Graph neural networks (GNNs) [23, 397] may be interesting alternatives as they could directly operate on the data represented on the unstructured mesh, thereby bypassing the need for the creation of a separate structured mesh and back and forth interpolation. MeshGraphNets [268] are GNN variants that have been used to simulate temporal dynamics of physical systems discretized on surface meshes. When temporal dynamics are defined by a closed-form ODE, graph neural networks can be trained as an ODE integrator using a Hamiltonian approach [321]. Numerical errors stemming from imperfect approximations of differential operators on unstructured meshes have been addressed by specific graph architectures [332]. Aside from learning temporal dynamics, direct modeling of physical quantities (without explicitly enforcing compliance to the ODE) have also been investigated using graph convolutional networks [24] and neural operators [192]. This task matches the scope of this work which is to learn an approximation of an LES SGS model. However, GNNs come with scalability issues that have not yet been fully addressed. Unlike CNNs which benefit from deeper architectures, GNNs are known to struggle when model depth is increased [191]. This is caused by the over-smoothing phenomenon, which is the tendency in deep GNNs of node features to collapse to similar values due to the diffusive action of the succession of message passing operations. This restricts GNNs to shallow architectures with limited expressivity and a narrow receptive field, hence limiting their appeal for context-based modeling compared to CNNs. Additionally, scaling GNNs to very large graphs containing $\mathcal{O}(10^6 - 10^8)$ nodes, around the order of magnitude of the size of an LES mesh, is still challenging. To this end, open benchmarks featuring large-scale graph datasets have recently emerged to accelerate research in this direction [3]. To overcome scalability barriers, node clustering [57] or subsampling [130] are common methods which have the drawback of altering the structure of the graph. All in all, developing GNN architectures which efficiently scale to graphs of the scale of typical LES unstructured meshes is still an active research topic [51, 200, 413].

Boundary conditions set by AVBP are not accounted for by AVBP-DL or the CNN. For a SGS wrinkling model, the nature of the boundary conditions is not expected to significantly affect model predictions. Other cases might nevertheless be sensitive to the choice of boundary conditions. For instance, when training a CNN to predict spatio-temporal dynamics of the heat equation or of acoustic wave propagation, proper handling of the boundary conditions was shown to be essential to the success of the model [7]. In such cases, coupled simulations with a fluid solver should include proper encoding of the boundary condition inside the DL model.

Chapter 8

LES of gas explosions with a wrinkling model

Contents

This chapter investigates the application of a CNN SGS model to vented obstructed gas explosions of premixed gases in an industrial-scale configuration. The presence of obstacles in a vented domain amplifies the damage caused by explosions that can occur due to accidental ignition of a premixed gas mixture in the domain. The main quantity of interest that determines the severity of the explosion is the peak overpressure induced by the sudden release of burnt gases. LES simulations heavily rely on turbulent combustion models to accurately predict this quantity. In this chapter, a CNN is trained to predict the SGS wrinkling factor directly, instead of the total FSD. The training set is the HIT dataset presented in Section 6.2.1, a choice which is discussed in the developments. The CNN is tested on the well-known Masri configuration. It demonstrates excellent *a priori* performance on filtered data from a Masri very resolved LES. The first *a posteriori* analysis of a deep CNN SGS model on an industrial-scale configuration is then carried out. The behavior of the CNN is analyzed in detail using comparisons with the reference CDSM model presented in Section 2.7.5. The CNN predicts the correct peak overpressure, but also incorrectly accelerates the flame in the initial laminar propagation phase. To correct this behavior, variants involving delaying the use of the CNN model and adding more physical information to the inputs of the neural network are investigated.

8.1 Overview of LES for explosions

8.1.1 Context

This chapter focuses on applications of numerical combustion to the safety of industrial processes. Industrial facilities in the energy sector are exposed to explosion hazards during the extraction, transport and storage of combustible energy vectors such as natural gas and hydrogen. They are found in their gaseous phase at standard temperature and pressure conditions and are usually compressed to very high pressures to minimize their volume. This increases the likelihood of explosions caused by the ignition of accidental leaks, which can lead to catastrophic material and human losses. As experimental data of realistic large-scale industrial explosions are scarce, numerical simulations can offer unique insights into the phenomenology of such an explosion, and help guide design choices to mitigate the associated risk.

This study focuses on explosions without transition to detonation, occurring in a domain initially filled with premixed gases at rest, with limited venting capacity through restricted openings to the outside atmosphere. After ignition and as the flame propagates in the domain, premixed gases are replaced with burnt gases which are an order of magnitude less dense. Because gases can only escape the domain through restricted venting outlets, this results in a build-up in pressure inside the chamber. The corresponding pressure differential relative to the initial pressure, or *overpressure*, is one of the key indicators of the severity of an explosion. Table 8.1 presents the human and material consequences of a few overpressure levels generated by an explosion.

The overpressure generated by an explosion is increased when it occurs in an obstructed domain. In the presence of obstacles inside the domain, the outward flow induced ahead of the propagating flame front generates turbulence in the wake of the obstacles. These

turbulent eddies wrinkle the flame front and accelerate its propagation. Obstacles can also locally increase the blockage ratio of the domain, which exacerbates the effects of confinement on the overpressure.

To further understand the mechanisms behind the generation of overpressure, it is useful to examine the balance equation for the pressure P, assuming it is homogeneous [290]:

$$\frac{D}{Dt}\left\langle \frac{1}{\gamma-1}P \right\rangle_{chamber} = \langle \dot{\omega}_T \rangle_{chamber} \mathcal{V}_{chamber} - \langle \rho c_p T U_n \rangle_{outlet} \mathcal{S}_{outlet} \,, \tag{8.1}$$

where U_n is the flow velocity normal to the outlet, $\langle \cdot \rangle_{chamber}$ denotes volume averaging in the chamber of volume $\mathcal{V}_{chamber}$, and $\langle \cdot \rangle_{outlet}$ denotes surface averaging on the chamber outlet of area $\mathcal{S}_{outlet}$. The first term of the right hand side is a turbulent combustion term enhanced by the wrinkling of the flame front as it propagates past and interacts with the obstructions in the domain. The second term is a venting term related to the mass flow rate of the gases exiting the domain. Equation 8.1 shows that predicting P in the chamber relies on predicting the chamber-averaged heat release rate $\langle \dot{\omega}_T \rangle_{chamber}$ which is one of the most difficult quantities to evaluate in turbulent combustion models. Based on Equation 8.1, empirical 0D models can be developed to estimate the overpressure [290] but they cannot account for the geometry of the chamber which plays a crucial role in the final result. Since flame acceleration is an unsteady process which depends on fine-grained flame-turbulence interactions, LES is the method of choice to simulate this phenomenon. The main challenge lies in the large separation of scales between the characteristic size of the laminar flame front ($\delta_L \sim 0.1\,\text{mm}$) and the size of the domain (up to a few meters). This greatly constrains how small the mesh size can be chosen to maintain a tractable computational cost. LES must therefore produce predictive simulations in spite of severe resolution limitations, which places a heavy burden on the turbulent combustion model.

Overpressure magnitude	**Consequences**
3 mbar	Loud noise (143 dB)
70 mbar	Shattered windows (90%) People are thrown off their feet
170 mbar	Destroyed houses (50%)
430 mbar	Ruptured eardrums (50%) Destroyed houses (∼100%)
840 mbar	Ruptured eardrums (90%)
1.75 bar	Death by direct blast (90%)

Table 8.1: Human and material consequences of the overpressure induced by an explosion, with indicative probabilities in parentheses [258].

8.1.2 The Masri test case

A well-studied benchmark for vented obstructed gas explosions is the Masri test case [216, 217] depicted in Figure 8.1. The domain is a 3D rectangular chamber closed on all sides

except for an outlet face opening to a large plenum kept at atmospheric pressure. It is initially filled with a fully premixed stoichiometric propane-air fresh mixture at rest. The mixture is ignited at the closed end of the chamber, leading to the development of a hemispherical laminar flame. The propagation of the flame is disrupted by a succession of obstacles placed inside the chamber. The turbulence generated in the wake of the obstacles causes significant wrinkling of the flame front, which triggers flame acceleration and a pressure build-up inside the chamber. Experiments with this configuration have been carried out for a domain length of 25 cm (small scale or SS) by Prof. A. Masri and his team [216], and later repeated on two larger scales 1.5 m (medium scale MS), and 6 m (large scale) by GexCon [369]. From these experiments, overpressure time signals and high-speed video footage providing the propagation speed of the flame front are used to validate numerical simulations.

Variations on the Masri configuration involve varying the position and number of obstacles. In all cases covered here, a large central obstacle is placed as the last obstruction. Three positions upstream of the central obstacle can be occupied by removable grids. The configurations are referenced by a quadrigram XXXS, where X can be B (the grid is present) or O (the grid is removed). The more rows are present, the higher the peak overpressure is. Figure 8.1 shows the detailed measurements of the BBBS SS configuration which is the focus of this work.

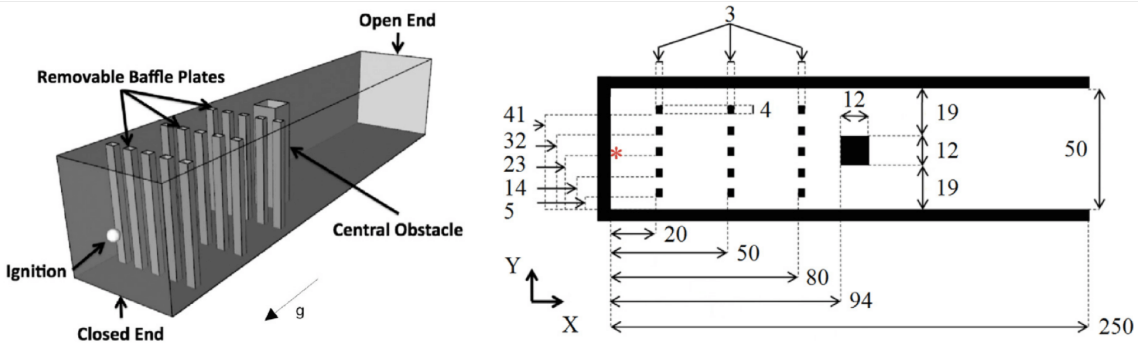


Figure 8.1: Masri test case in the BBBS SS configuration (3 rows of grids followed by a central obstacle). Left: 3D view, right: top view with domain measurements (in millimeters) and location of experimental overpressure measurements (red asterisk). Figures from Vermorel et al. [369] and Volpiani et al. [376].

The various stages of flame propagation are shown in Figure 8.2. After an initial laminar expansion phase, the flame kernel crosses the first obstacle grid at $t = 6$ ms, first through the two central channels and then through the outer channels. Moderately wrinkled finger-like structures are generated by the pinching and acceleration of the flame as it passes through the grid. The two central fingers drive the propagation of the leading edge of the flame. They merge before going through the second obstacle grid ($t = 8$ ms). After the second grid ($t = 9.6$ ms), the flame tips accelerate further in the streamwise direction. As a consequence, the finger structures are more elongated and do not have time to merge in the transverse direction before meeting the third row. All the central and outer fingers are also noticeably wrinkled by turbulence. During the passage of the third grid ($t = 10$ ms) and of the central obstacle ($t = 10.8$ ms), the flame tips experience a final phase of flame acceleration caused

by intense turbulence in the wake of the obstacles. The large discrepancy between the propagation speed of the tips and other regions of the flame front generates a vast amount of flame surface, including many pockets of unburnt gases trapped in the wake of the obstacles. Peak overpressure measured at the closed end of the chamber is generated by the sharp increase in flame surface as the flame tip crosses the central obstacle.

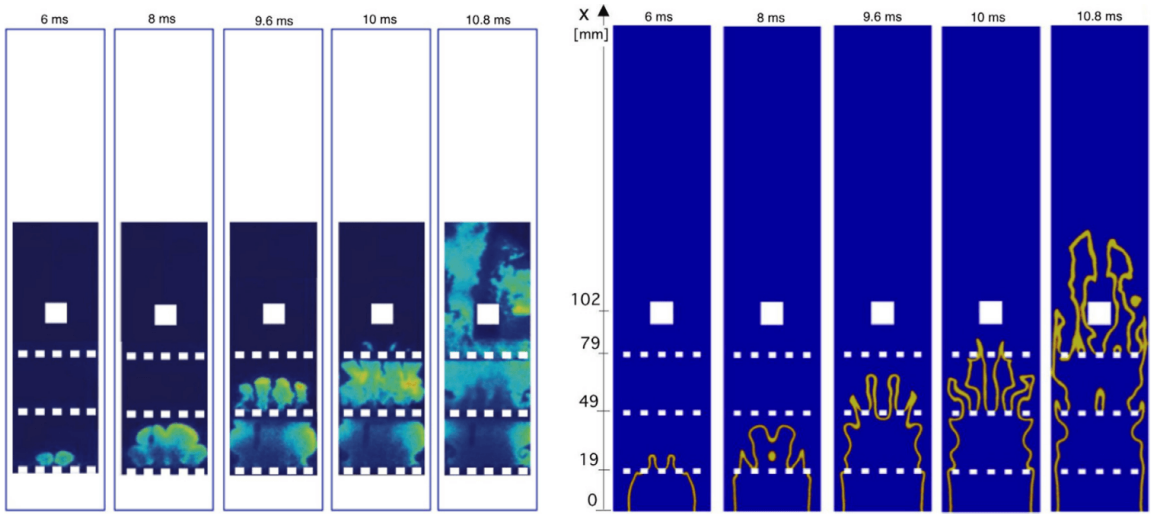


Figure 8.2: Visualizations of flame propagation in the Masri experiment (left, OH-PLIF) and LES (right, fuel consumption rate). Figure from Vermorel et al. [369].

Previous LES studies on the Masri configuration [291, 369, 376] have shown that TFLES with static efficiency models [45, 67] predicts an overpressure evolution that closely matches experimental results in the SS and MS configurations, and also correctly reproduces parametric variations at a given scale. As an example, Figure 8.3 shows the overpressure signal from an AVBP LES simulation using the Colin model [67] on the small-scale configuration with three, two, and one row(s) of obstacle grids. In all cases, the peak overpressure of the LES lands in the envelope of experimental results. However, these models rely on an arbitrary determination of the model constant which cannot be known prior to the computation. As a consequence, they struggle to account for variations in the scale of the domain. Fig. 8.4 shows the overpressure predicted using the Colin and Charlette efficiency functions with standard parameters $\beta_{Colin} = 0.3$ and $\beta_{Charlette} = 0.5$ for the SS and MS configurations. The Colin model fits the experimental overpressure very well on the SS but underpredicts the overpressure on the MS while the Charlette model has the opposite behavior. Fine tuning the model coefficients is possible but only when the experimental results are known, hence weakening the predictive ability of the turbulent combustion models.

Volpiani et al. [376] reported very good results using the dynamic saturated Charlette model on the Masri BBBS SS configuration. The dynamic model has the advantage of correctly capturing the laminar and turbulent phases of the flame development without arbitrary parameter tuning. In the laminar spherical phase, as long as the ignition radius is not too small [238], the filtered and test-filtered flame fronts are identical and the dynamic model predicts no SGS wrinkling. When the flame crosses the obstacle grids, it is significantly

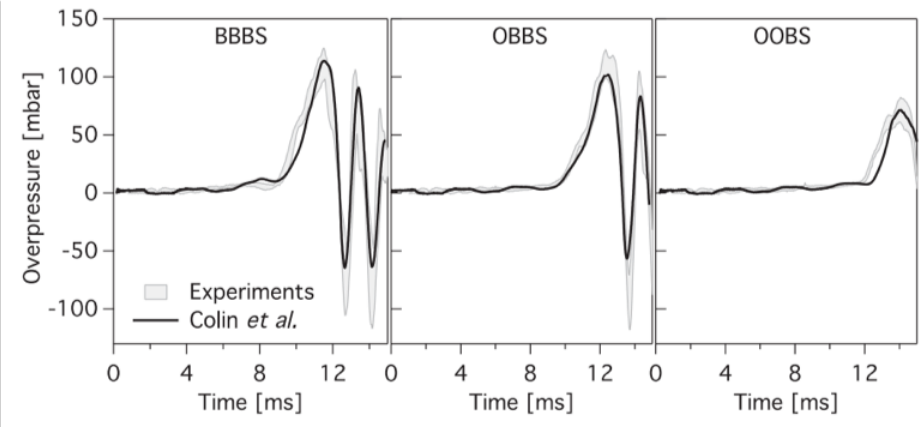


Figure 8.3: Overpressure time evolution for the SS case with three (left), two (middle), and one (right) row(s) of obstacle grids. Grey: experimental envelope. Black: TFLES simulation with the Colin efficiency model [67]. From [369].

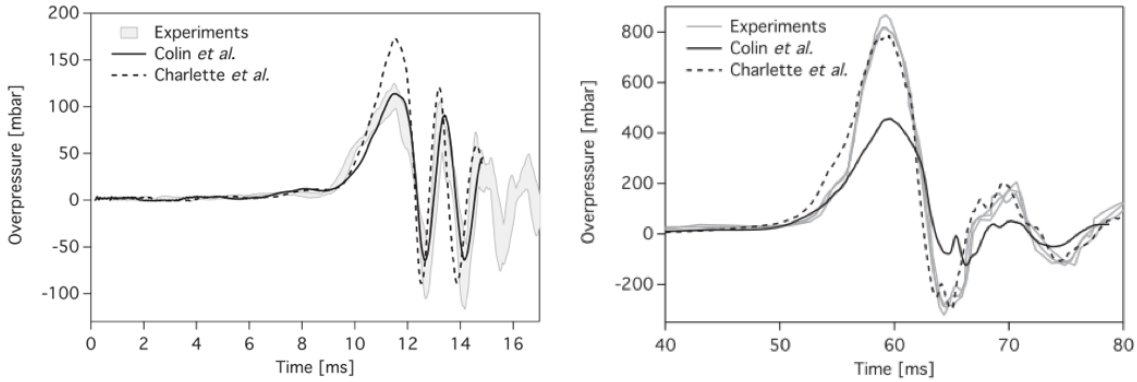


Figure 8.4: Overpressure time evolution for the SS case (left) and MS case (right) using Colin [67] and Charlette [45] efficiency models with standard parameters. From [369].

curved due to the pinching effect of the narrow gap between the grids. Coupled with the effect of the resolved turbulence in the obstacle wake, this triggers the activation of the dynamic model in the wake of the obstacles, thus correctly accelerating the flame. To highlight the variation of the model parameter over the course of the simulation, Figure 8.5 shows the evolution of the $\beta_{Charlette}$ parameter as a function of the flame tip position. In all the obstacle variations that are studied, β remains negligible in the laminar phase before the flame crosses the first obstacles, then increases as long as additional obstacles are present. The dynamic model correctly predicts the peak overpressure in the SS configurations and underlines the need to account for spatial and temporal variations of β based on the local flame geometry. However, no conclusive results have yet been reached on the MS and LS cases.

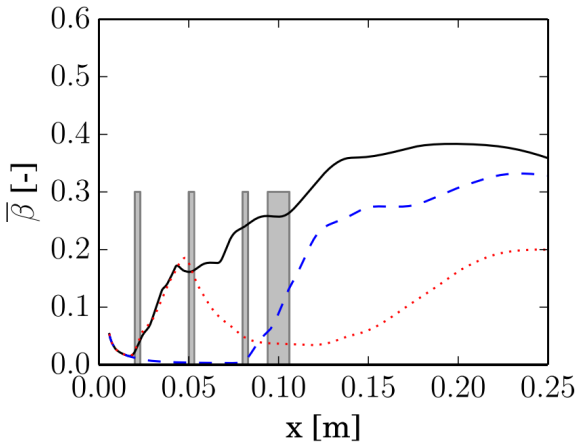


Figure 8.5: Evolution of the mean $\beta_{Charlette}$ parameter as a function of the flame tip position in the BBBS (solid black), OOBS (dashed blue), and BOOS (dotted red) configurations. From [376].

Overall, the Masri configuration is a challenging test case to benchmark turbulent combustion models as:

- it features laminar and turbulent flame propagation which must both be well reproduced,
- the peak overpressure is sensitive to numerical, physical, and geometrical parameters [369],
- existing models fail to scale to the medium and large scales if all model parameters from the small-scale are used.

This chapter investigates to what extent deep learning models can produce accurate overpressure results on the Masri configuration. Additionally, since no DNS data from Masri or another confined explosion configuration was used to train the model, this constitutes a challenging framework to study how CNNs can generalize to LES of large-scale realistic configurations like the Masri case.

8.2 SGS wrinkling modeling

For the Masri configuration, the goal is to train a SGS model that easily integrates in the TFLES formalism used by AVBP for *a posteriori* validation. Like in Chapter 5, a model for the unresolved flame surface is chosen. Unlike Chapter 5 however, the model learns to predict the SGS wrinkling factor Ξ and not the total FSD. The SGS wrinkling factor can be directly used as a TFLES efficiency function, whereas the total FSD must first be converted to a wrinkling factor by division with the resolved FSD (Equation 2.91)

From model predictions of the total FSD $\overline{\Sigma}^{+}_{\mathrm{CNN}}$ given by Equation 5.1, a wrinkling factor can be computed from the resolved FSD as:

$$\Xi_{\mathrm{CNN}} = 1 + S \times \left(\frac{\overline{\Sigma}^{+}_{\mathrm{CNN}} \sigma}{|\nabla \bar{c}| + \epsilon} - 1 \right) , \tag{8.2}$$

using $\epsilon << |\nabla \bar{c}|$ and a sensor S based on the the values of $|\nabla \bar{c}|$ to avoid divisions by zero. For instance,

$$S = \begin{cases} 1, & \text{if } |\nabla \bar{c}| > \max(|\nabla \bar{c}|)/10, \\ 0, & \text{otherwise.} \end{cases} \tag{8.3}$$

Figure 8.6 shows the behavior of this model on a 1D laminar flame. In this case, the resolved and total FSD are equal, and the model should predict a wrinkling profile uniformly equal to unity. The top plot shows that the CNN is able to capture nearly perfectly the evolution of $\overline{\Sigma}^+$ in the flame front. Tiny errors nevertheless have a noticeable impact on the final profile of Ξ_{CNN}. In the flame front, the CNN wrinkling factor Ξ_{CNN} is lower than one because of slight underpredictions of $\overline{\Sigma}^+$. This issue is amplified in *a posteriori* due to approximation errors in the computation of $|\nabla \bar{c}|$ on an unstructured mesh, especially for curved laminar flame fronts. The model is therefore unable to reliably propagate a laminar flame at the laminar flame speed.

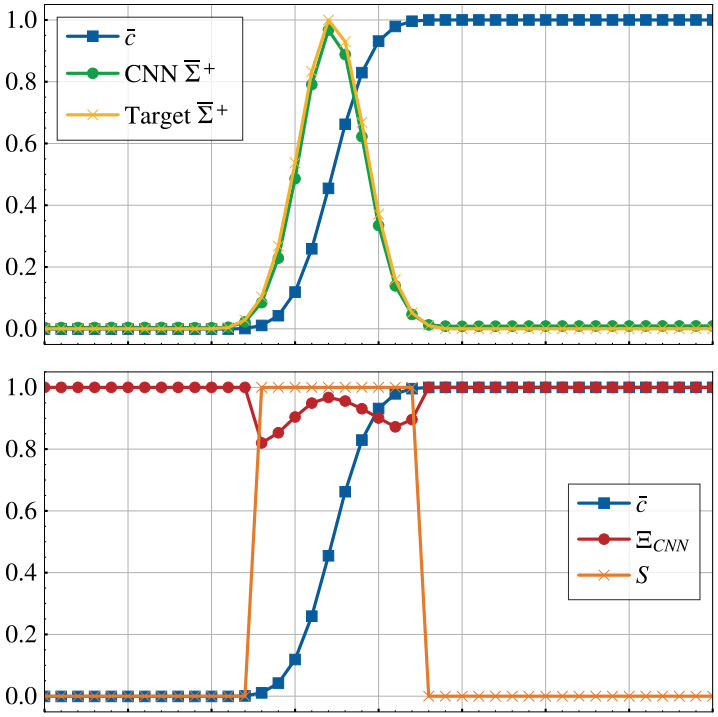


Figure 8.6: Illustration of non-unity wrinkling arising from a CNN model for $\overline{\Sigma}^+$. Ground truth and CNN $\overline{\Sigma}^+$ profiles (top), numerical sensor and wrinkling factor profiles (bottom).

To overcome this issue, this chapter focuses on the direct modeling of the wrinkling factor. This formulation was found to lead to more stable *a posteriori* efficiency values, as the output of the model suffers no numerical issues and is directly used in the solver.

8.3 Choice and relevance of the training dataset

The Masri configuration offers a unique opportunity to further study how far CNNs can generalize away from their training database. The approach is summarized as follows: a CNN is trained to predict SGS wrinkling on a canonical turbulent flame, and directly used in an LES of the Masri configuration. The canonical configuration is the statistically planar flame in decaying HIT presented in Section 6.2.1 which covers both laminar and turbulent wrinkling regimes. The CNN is therefore able to learn a diverse representation of Ξ which can be applied to the Masri LES. The premise is that the $\tilde{c} \mapsto \Xi$ mapping learnt on the HIT also applies to Masri. The relevance of training the CNN on this dataset to apply it on the Masri configuration is a crucial point that merits a careful discussion.

Strong constraints on filtering and turbulent combustion characteristic ratios were imposed in Chapter 6 to set strict prerequisites for generalization. These constraints need to be relaxed for a Masri simulation where these ratios are difficult to evaluate and significantly vary depending on the time and location of the flame. Estimates of the turbulent combustion ratios can nonetheless indicate whether the choice of the training dataset is appropriate. The integral lengthscale l_t is chosen as the intergrid y spacing, leading to $l_t/\delta_L = 15$. Experimental studies [123] have measured turbulent velocity fluctuations encountered by the leading edge of the flame and report corresponding u'/s_L values ranging from 0.1 to 40. Numerical estimates from LES simulations [290] fall into this range as well. To complement these estimates, the integral lengthscale and turbulent intensity were computed on a VRLES snapshot of the OOBS configuration (see Section 8.6.1) using the same method as on the HIT (Section 6.2.1) and only in the flame front aft of the third obstacle grid where the flame is fully wrinkled. This leads to an integral lengthscale value which is consistent with the estimate from the intergrid spacing.

Figure 8.7 reports the locations of HIT and Masri operating points in the Borghi-Peters turbulent combustion diagram. The key observation is that both simulations are mainly located in the thin reaction zone regime, except for the onset of the Masri simulation where the flame is laminar, and that the turbulence intensity and Karlovitz number seen by the HIT flame are contained in the range spanned by the Masri simulation. It can thus be expected that in both configurations, the flame front is moderately wrinkled by turbulence on multiple scales, while preserving a flamelet structure.

Nonetheless, the values of l_t/δ measured on the HIT flame are smaller than in the the Masri configuration. Compared to the filter size $\Delta/\delta = 15$, the integral lengthscale in the HIT is much smaller than Δ while the integral lengthscale in the Masri configuration is larger. Whether this difference impacts the relationship between the filtered progress variable field and the wrinkling factor is unknown. Understanding the marginal effect of the integral lengthscale on the wrinkling factor at a filter scale Δ could be an interesting avenue for future research. Additionally, although fully laminar flame fronts are present in the HIT, it does not contain any spherical laminar flame fronts. Wrinkling induced by flame acceleration around and between obstacles is not covered, and the range of turbulence intensities is narrower than in the Masri simulation. All these factors highlight the limitations induced by the use of a single training dataset on the modeling capacity of the CNN. Future work should focus around the design of more varied datasets, which come at a great computational premium

but could be specifically tailored to explosion configurations. Nevertheless, the applicability of the CNN is motivated by the characterization of the HIT and Masri wrinkled flame fronts in the same turbulent combustion regime, and the limitations raised in this discussion will help guide the *a posteriori* analysis of the CNN.

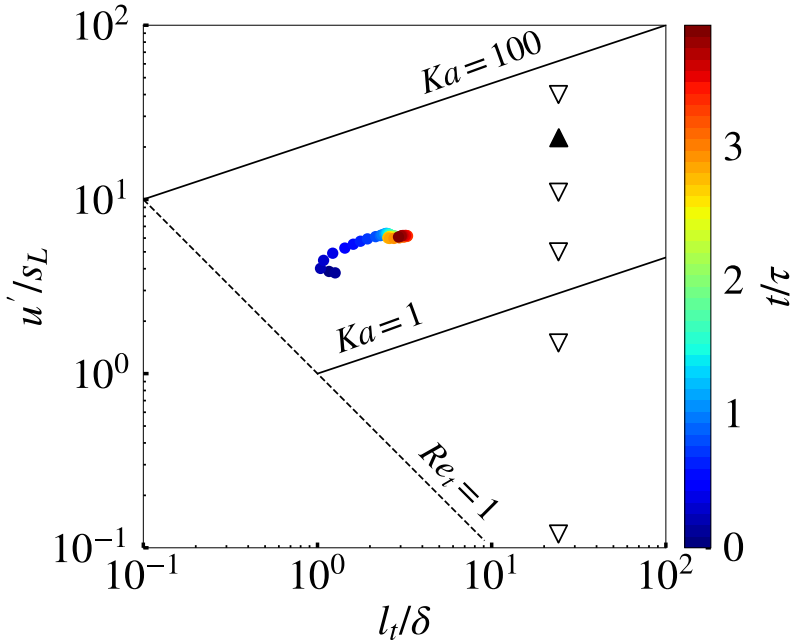


Figure 8.7: Borghi-Peters turbulent combustion diagram. ○: HIT points colored by time, ▽: Masri BBBS values reported in the literature [123, 290], ▲: Masri OOBS values measured on a VRLES snapshot (Section 8.6.1).

Figure 8.8 summarizes the methodology used to train and then evaluate *a posteriori* the CNN model. The CNN is trained on filtered snapshots of the HIT DNS. Its trainable weights are then frozen, and the resulting trained model is used in the LES of the Masri BBBS small-scale configuration. Local and integral comparisons are performed with a reference simulation that uses the CDSM model.

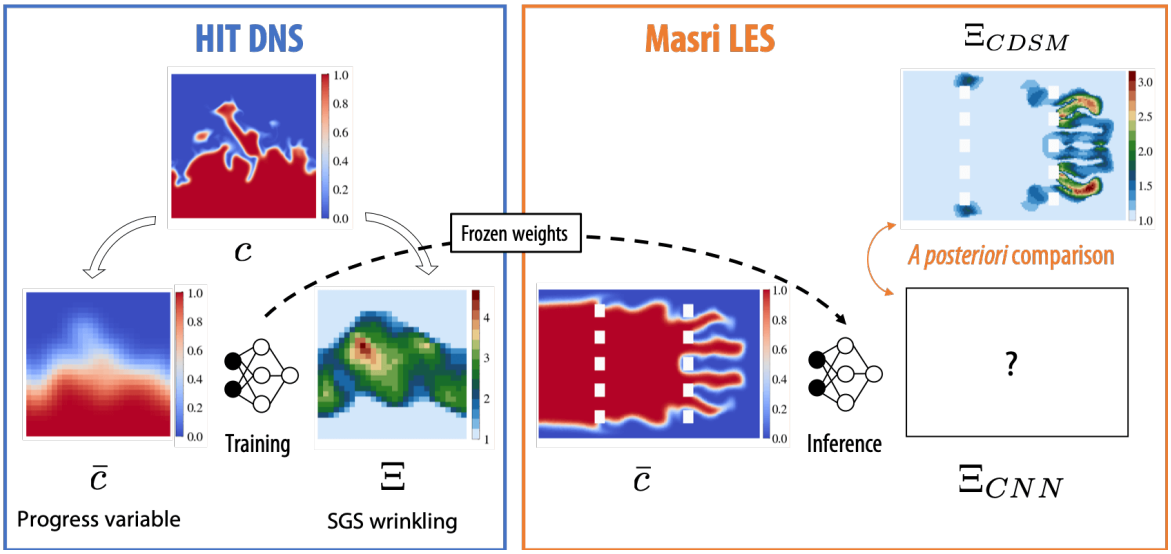


Figure 8.8: Summary of the training and *a posteriori* evaluation process.

8.4 Training data preparation

8.4.1 Determining the filtering parameters

Like in the previous chapters, filtering and downsampling is applied to the HIT DNS snapshots to generate $\bar{c}$ and Ξ fields. Ξ is computed as in Equation 8.2.

The HIT and Masri flames are both stoichiometric propane-air flames with the same unburnt temperature and pressure. The only difference is the chemical mechanism, with the HIT using a one-step Pfitzner source term while the Masri simulation uses a two-step Arrhenius chemistry (more details in Section 8.7). Both laminar flame thicknesses and profiles are therefore very similar, and the filtered flame thickness of the HIT $\bar{\delta}_L^H$ is close to the thickness of the Masri flame $\check{\delta}_L^M$.

The CNN is trained on a single filter size Δ^H and coarse mesh resolution h^H. Although a generic model should be able to handle varying filter sizes, the filter size and resolution of the thickened flame fronts in the Masri are constant, and it is therefore possible to restrain to study to a single filter size. Algorithm 2 is used to determine the filter size Δ^H and coarse mesh size h^H of the HIT. The filtering parameters for the HIT are reported in Table 8.2.

δ_L^H	h^H	Δ^H	$\bar{\delta}_L^H$
352 µm	500 µm	3289 µm	2494 µm

Table 8.2: Filtering parameters for the HIT.

8.4.2 Flame-flame interaction corrections

Figure 8.9 shows HIT samples of a quasi-laminar and a fully wrinkled flame front, as well as their corresponding $\bar{c}$ and Ξ fields. For the laminar flame front, the wrinkling factor is

uniformly very close to unity. In the wrinkled flame front, the large size of the filter relative to the wrinkled structures leads to a destruction of flame surface. The filtered flame front loses its flamelet structure and contains thick regions of moderate c values. Moreover, wrinkling values can reach spurious high values due to the behavior of $|\nabla \bar{c}|$ in flame-flame interactions that occur at scales below the filter scale. This is the same issue that can occur in the test-filtering operation of the dynamic Charlette model. Here, its amplitude is magnified by the large size of the filter compared to the wrinkled structures. The solution proposed by Mouriaux et al. [239] in the CDSM model and described in Section 2.7.5 is used to correct the values of wrinkling. Here, filtering at the scale Δ takes the role of test-filtering in the dynamic model, and the DNS flame front replaces the LES thickened flame front.

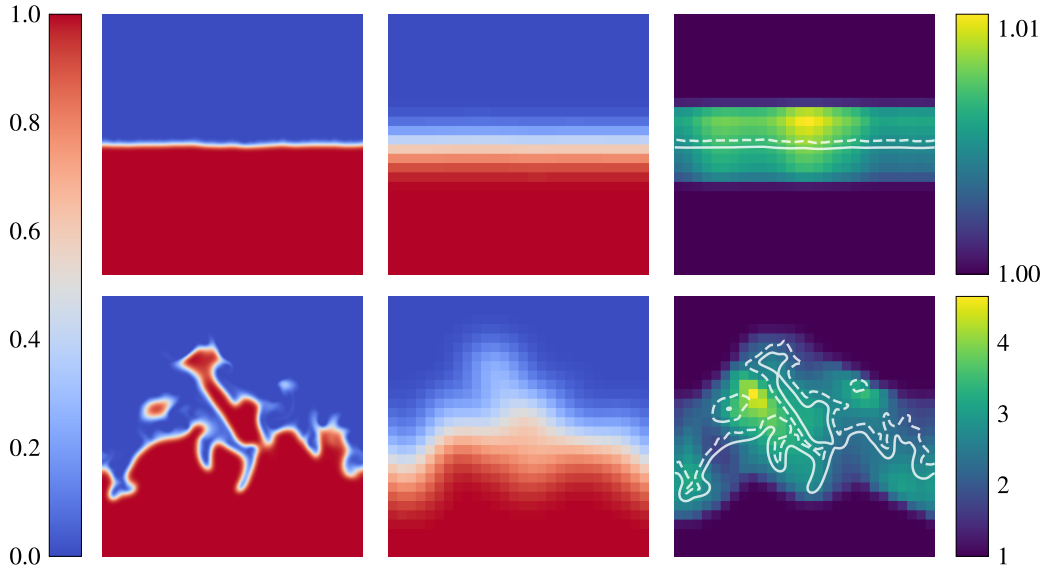


Figure 8.9: Sample slices from the training dataset at $t = 0.1\tau$ (top) and $t = 1.7\tau$ (bottom). Left: DNS c, middle: filtered $\bar{c}$, right: Ξ after flame-flame interaction corrections, with isolines $c = 0.1$ (dashed), $c = 0.9$ (solid).

Figure 8.10: Sample slices of wrinkling factor before (left) and after (middle) corrections for flame-flame interactions, with isolines of $c = 0.1$ (dashed), $c = 0.9$ (solid). The difference between the corrected and uncorrected fields is shown on the right.

Figure 8.10 compares the wrinkling field before and after corrections of flame-flame-interactions. Before the corrections, values up to 13 are found in regions between opposing DNS flame fronts. Mouriaux corrections for this behavior effectively target only these regions, while leaving wrinkling in the rest of the flame front untouched. One difference compared to the corrections for the dynamic model lies in the threshold value for the interaction sensor ζ (Equation 2.105). In the CDSM model, the test-filter scale is typically 1.5 to 2 times the thickened flame filter scale, so minute misalignments of the filtered and test-filtered normals can reveal flame-flame interactions. The threshold value used in Mouriaux et al. [239] is therefore high, *i.e.* $\boldsymbol{n} \cdot \boldsymbol{N} = 0.9$. Here, since filtering is performed at a larger scale relative to the flame thickness ($\Delta/\delta_L = 9.3$, see Table 8.2), a greater amount of wrinkling is lost, and the filtered flame front is more prone to misalignment with the DNS flame front. A threshold $\boldsymbol{n} \cdot \boldsymbol{N} = 0.9$ activates ζ in the entire DNS flame front and renders the corrections ineffective, as illustrated in Figure 8.11. A value of $\boldsymbol{n} \cdot \boldsymbol{N} = 0.2$ was found to better isolate the problematic regions and correctly mitigate wrinkling there.

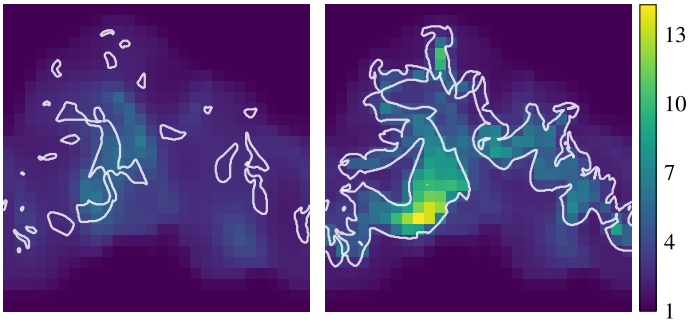


Figure 8.11: Effect of flame-flame interaction corrections with a threshold $\boldsymbol{n} \cdot \boldsymbol{N} = 0.2$ (left) and $\boldsymbol{n} \cdot \boldsymbol{N} = 0.9$ (right), with isolines of $\zeta = 1$, on the same slices as Figure 8.10.

Given the non-flamelet structure of some filtered flame elements and the use of flame-flame interaction corrections, the definition of the wrinkling field and its relevance as a SGS modeling target need to be justified. The corrected wrinkling factor field must allow the DNS flame surface to be recovered from filtered quantities. The generalized flame surface density, defined as $\overline{|\nabla c|}$, is representative of the subgrid surface density of any c isosurface in thin flamelets, where all the c isosurfaces are parallel and have the same surface area. In a thin flame front like in the DNS, volume integration of the generalized total FSD leads to a flame surface equal to the area of any c isosurface. In a non-thin filtered flamelet like in Figure 8.9, the corrected wrinkling field must recover the DNS flame surface:

$$\int |\nabla c| \, \mathrm{d}V = \int \Xi |\nabla \bar{c}| \, \mathrm{d}V \tag{8.4}$$

Figure 8.12 plots the DNS flame surface $\int |\nabla c| \, \mathrm{d}V$, the filtered flame surface $\int |\nabla \bar{c}| \, \mathrm{d}V$ and the filtered flame surface with the corrected wrinkling factor $\int \Xi |\nabla \bar{c}| \, \mathrm{d}V$ for all snapshots in the HIT dataset. The corrected wrinkling factor perfectly recovers the DNS flame surface in the early phase of the DNS, when the flame is weakly wrinkled. Later on, a deviation from the DNS flame surface is observed and could be explained by the non-flamelet nature of the

filtered flame front and by the effects of flame-flame interaction corrections. The relative error remains smaller than 10%, and the corrected wrinkling field is therefore deemed a reliable measure of the unresolved flame surface density.

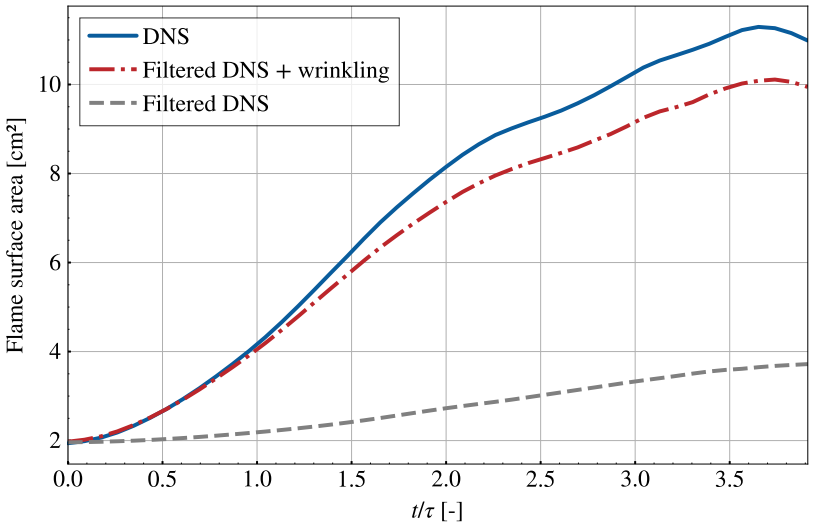


Figure 8.12: Flame surface from the DNS, filtered DNS and filtered DNS with the corrected wrinkling factor over the course of the HIT simulation.

8.4.3 Dataset characteristics

The training dataset contains the first 41 snapshots of the HIT. The last 5 snapshots of the HIT are kept as a validation set. Downsampling the filtered fields from the DNS mesh size $h_{\text{DNS}} = 36\,\mu\text{m}$ to the coarse mesh size $h^H = 500\,\mu\text{m}$ (Table 8.2) drastically reduces their spatial dimensions from 384^3 to 28^3. Like in Section 5.5.2, periodic padding is applied in the y and z periodic directions to increase the spatial dimensions to $28 \times 52 \times 52$. Table 8.3 summarizes the composition of the training, validation, and test datasets.

Training	**Validation**	**Test (Section 8.6.1)**
First 41 snapshots of HIT	Last 5 snapshots of HIT	1 snapshot of Masri VRLES
$41 \times 28 \times 52 \times 52$	$5 \times 28 \times 52 \times 52$	$1 \times 116 \times 104 \times 104$

Table 8.3: Composition of the datasets used for training and *a priori* testing.

8.5 Model training

The U-Net architecture presented in Section 5.5.1 is trained with the procedure detailed in Section 5.5.2. Training hyperparameters are reported in Table 8.4. The size d_c and number n_c of the random crops are smaller than in Section 5.5.2 because of the reduced spatial dimensions of the filtered snapshots. This is partly compensated by a larger number of snapshots sampled per batch n_s. Training and validation losses are plotted in Figure 8.13.

Training is stopped when an adequate accuracy is reached on the validation set. Figures 8.14 and 8.15 show that the trained CNN predicts Ξ on planar and wrinkled flame fronts from the training dataset very accurately.

n_c	n_s	d_c	η	Batches per epoch	Epochs
4	16	24	0.01	50	1000

Table 8.4: Training hyperparameters for the HIT dataset.

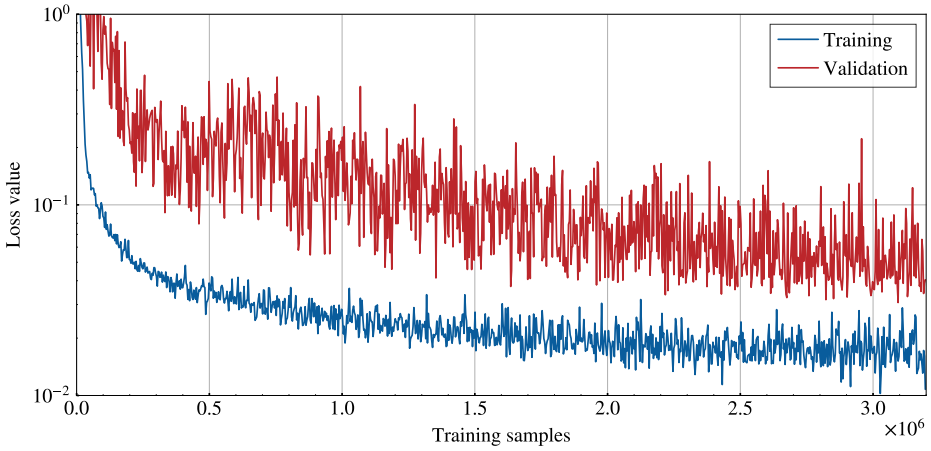


Figure 8.13: Training and validation loss curves on the HIT dataset.

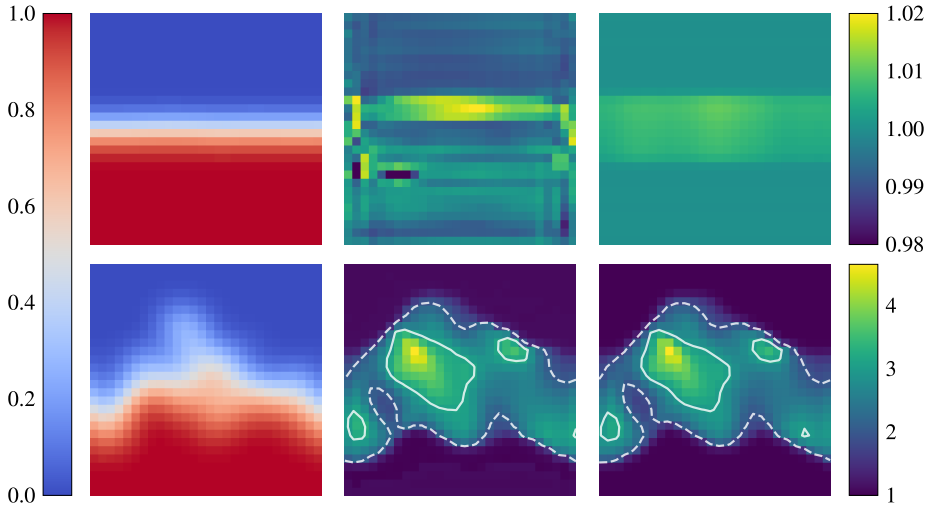


Figure 8.14: Visual comparison of the predictions of the trained CNN on snapshots at $t = 0.1\tau$ (top) and $t = 1.7\tau$ (bottom). Left: $\bar{c}$, middle: ground truth Ξ, right: Ξ predicted by the CNN. Isolines of $\Xi = 2$ (dashed), $\Xi = 3$ (solid) are shown for the bottom Ξ fields.

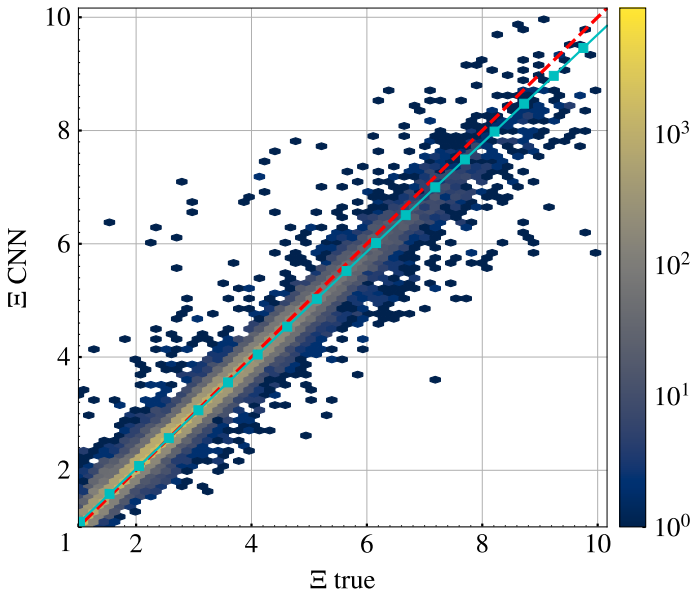


Figure 8.15: Hexbin plot of model predictions against the ground truth on the HIT validation set, colored by bin count. The red line $y = x$ represents a perfect model. The cyan line is the conditional mean of model predictions.

8.6 *A priori* testing on a reference LES simulation

8.6.1 Presentation of the Masri OOBS VRLES

The test set used for *a priori* validation of the CNN is a very refined LES (VRLES) of the Masri small-scale OOBS configuration, performed by Pierre Quillatre during his PhD [290, 369]. In this configuration, the chamber only contains one obstacle grid placed close to the central obstacle. The laminar propagation phase therefore extends further than in the BBBS configuration, as the flame front only wrinkles when it passes through the grid. The peak overpressure and turbulence intensity are thus lower than in the BBBS configuration. The domain is discretized with a 973 million cell mesh of tetrahedral elements, with a characteristic mesh size in the chamber $h = 136\,\mu m$. Table 8.5 compares the mesh size to the laminar flame thickness, Kolmogorov lengthscale, and integral lengthscale. With $h/\eta = 3.9$, the smallest eddies are close to being fully resolved. A laminar flame front would be resolved on $\delta_L/h = 2.5$ cells, and as a result, to resolve the thickened flame front on $N_c = 5$ cells, a thickening factor of only $F = Nh/\delta_L = 2$ is needed.

h	δ_L/h	h/η	l_t/h
136 µm	2.5	3.9	43

Table 8.5: Comparison of the VRLES mesh size with characteristic combustion and turbulence lengthscales (from [369]).

Details on the numerical setup of the simulation are available in the original papers [290,

369]. The thickened flame model with a target thickened flame resolution of $N_c = 5$ cells in the flame front is used, leading to a thickening factor $F \approx 2$ in the flame front inside the chamber. SGS flame wrinkling is modeled by the Colin efficiency model (Equation 2.94) with $\beta_{Colin} = 0.3$. Initially, the domain is filled with a stoichiometric premixed propane-air mixture at $T_u = 300\,\mathrm{K}$. A half-sphere of burnt mixture with a 1 cm radius is deposited at the closed end of the chamber to initialize the spherical laminar propagation phase. A reduced 2-step chemical mechanism ensures that the computational cost of the simulation stays as low as possible, while correctly reproducing the main features of flame acceleration and flame wrinkling [291].

The CNN is tested on a snapshot of the VRLES at $t = 11.5\,\mathrm{ms}$, when the tip of the flame front propagates past the central obstacle. This coincides with the peak generation of overpressure as the leading edge of the flame enters the last blocked section of the chamber and meets the turbulent wake generated downstream of the central obstacle. Visualizations of the vorticity magnitude and a $\check{c}$ iso-surface colored by efficiency are shown in Figures 8.16 and 8.17.

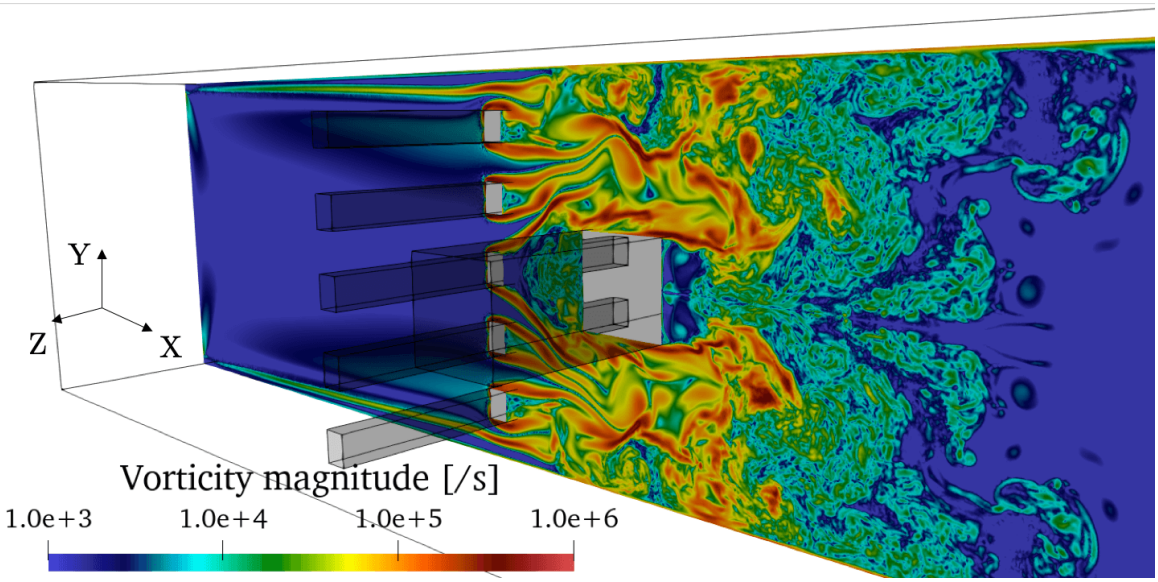


Figure 8.16: Centerplane slice of vorticity magnitude in the VRLES.

8.6.2 *A priori* results

The fine resolution of the turbulence and low value of the thickening factor in the VRLES indicate that a large part of the flame surface is resolved. The efficiency function was shown to play a limited part in the total fuel consumption rate (Figure 20 in [369]), which is confirmed by the moderate values taken by the efficiency in Figure 8.17. Moreover, the VRLES correctly predicts the peak overpressure and evolution of the flame tip speed (Figure 19 in [369]). As a consequence, modeling errors in the efficiency function are expected to be small, and $E|\nabla\check{c}|$ is considered to reliably measure the total flame surface density at the scale of the VRLES.

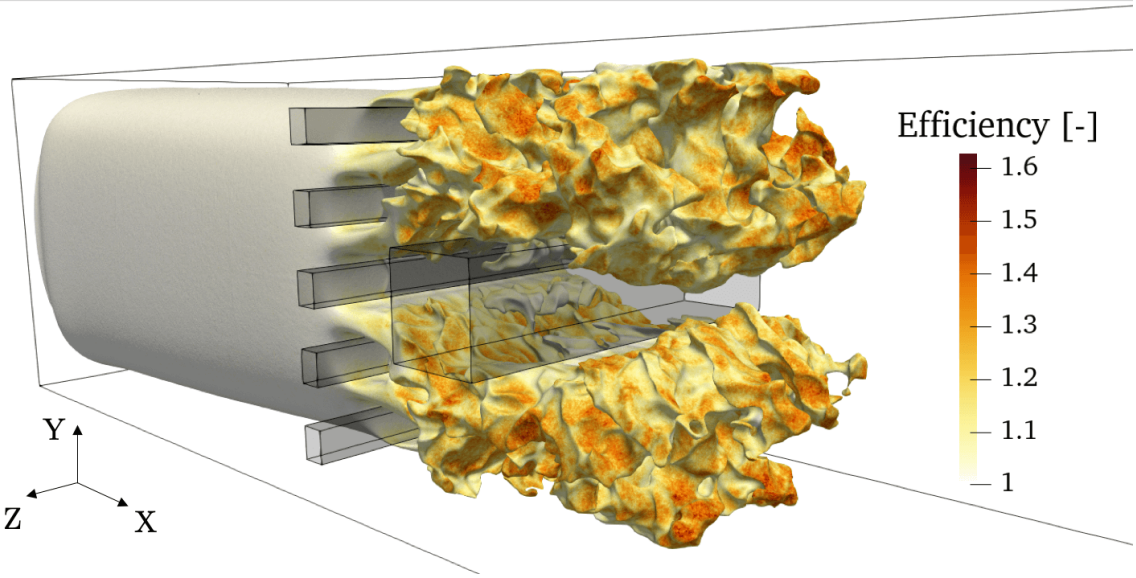


Figure 8.17: Iso-surface of $\check{c} = 0.86$ colored by efficiency in the VRLES. $\check{c} = 0.86$ is the progress variable value of peak heat release rate in a laminar flame.

The VRLES snapshot is interpolated by inverse distance weighting onto a structured mesh of uniform hexahedral cells with the same mesh size h as the unstructured mesh. After this step, structured mesh points inside the obstacle are filled with an arbitrary value. Next, filtering $\bar{\cdot}$ at the scale Δ^M is performed through masked Gaussian convolutions which ignore the values inside the obstacles and appropriately normalize the convolution kernel. Since the CNN operates on a structured grid, values of $\bar{\check{c}}$ must be prescribed inside the obstacles. Nearest neighbor interpolation is therefore applied inside the obstacles to fill in missing values of $\bar{\check{c}}$. At the scale Δ^M, the ground truth total FSD is given by:

$$\overline{\Sigma}_{\mathrm{VRLES}} = \overline{E|\nabla \check{c}|}\,. \tag{8.5}$$

From the filtered $\bar{\check{c}}$ field, the CNN predicts a wrinkling field Ξ_{CNN}, leading to the corresponding total FSD:

$$\overline{\Sigma}_{\mathrm{CNN}} = \Xi_{\mathrm{CNN}}|\nabla \bar{\check{c}}|\,. \tag{8.6}$$

Note that the total FSD is used as the ground truth quantity instead of the wrinkling factor. Numerical noise in the wrinkling factor computed as $\overline{|\nabla \check{c}|}/|\nabla \bar{\check{c}}|$ makes it unwieldy for pointwise comparisons with the smooth predictions produced by the CNN.

Figure 8.18 compares the ground truth and CNN predictions for the total FSDs in the region around the central obstacle where the flame is most wrinkled. While the ground truth takes uniformly high values in the flame front, CNN predictions are less homogeneous. The CNN appears to predict high values where the filtered flame front is wrinkled. In Figure 8.19, a pointwise comparison of the total FSD values shows a wide scatter with relatively low bias. When isolating only bins containing more than 100 data points, scatter is considerably reduced, indicating that the majority of points are in fact distributed in a narrower distribution around the perfect model line. The total flame surface computed by

volume integration of the total FSD is only 10% higher for the CNN than the ground truth. The CNN is thus able to precisely predict the total flame surface of this filtered VRLES snapshot. This positive outcome of the *a priori* investigation of the generalization of the CNN to the OOBS configuration is promising for the final *a posteriori* application.

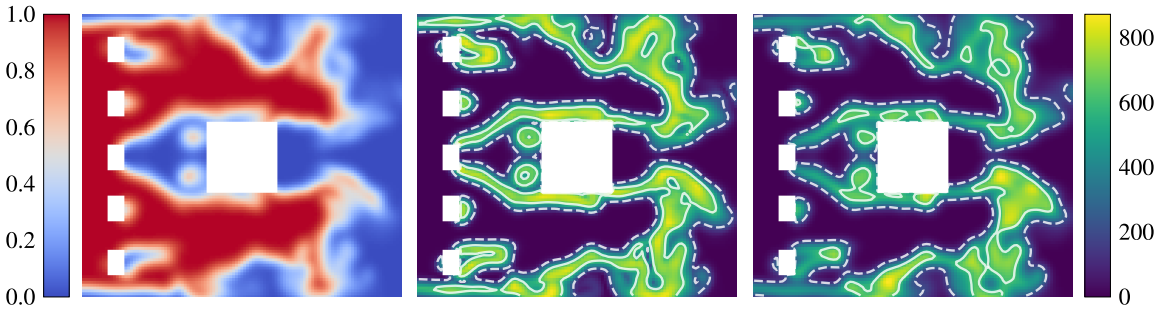


Figure 8.18: Slices of $\breve{c}$ (left), $\overline{\Sigma}_{\mathrm{VRLES}}$ (middle), and $\overline{\Sigma}_{\mathrm{CNN}}$ (right) on the VRLES snapshot, cropped around the central obstacle, with iso-lines of $\overline{|\nabla \breve{c}|} = 200$ (dashed), $\overline{|\nabla \breve{c}|} = 600$ (solid).

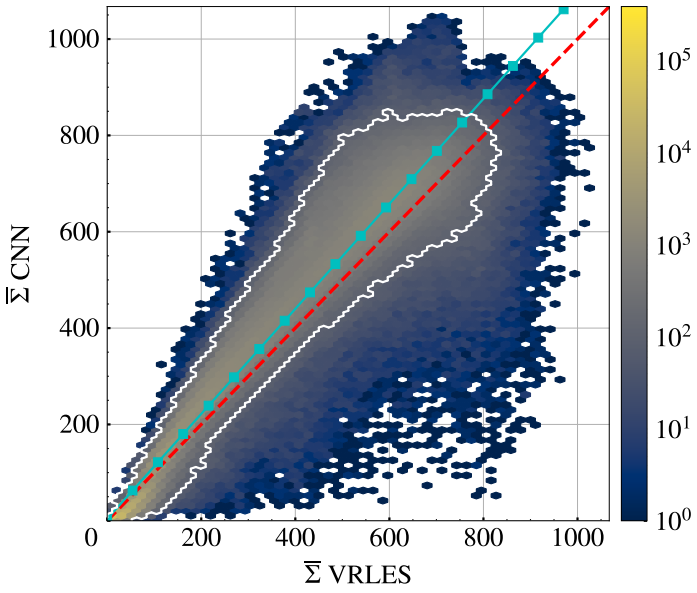


Figure 8.19: Hexbin plot of CNN predictions against ground truth total FSD. The white contour delineates bins containing more than 100 data points.

8.7 AVBP-DL LES numerical setup

The *a posteriori* validation of the CNN is carried out on the BBBS configuration which is the most challenging, as the presence of three obstacle grids, instead of a single one in the OOBS case, leads to greater flame acceleration and peak overpressure.

The domain is meshed with an unstructured mesh of 35 million cells, with a uniform mesh size $h = 500\,\mu\mathrm{m}$ in the main chamber, as illustrated in Figure 8.20. At the obstacle and

chamber walls, adiabatic no-slip boundary conditions are imposed. NSCBC outlet conditions at atmospheric pressure are prescribed at the exit of the plenum. SGS turbulent stresses are computed by the WALE model (Equation 2.108). The same propane-air 2-step chemical mechanism as in the VRLES is used, with unity Lewis numbers for all the species. The simulation is initialized with a 1 cm radius half-sphere of burnt mixture deposited at the closed end of the chamber, with a laminar flame profile imposed at the transition with the premixed mixture that fills the rest of the chamber.

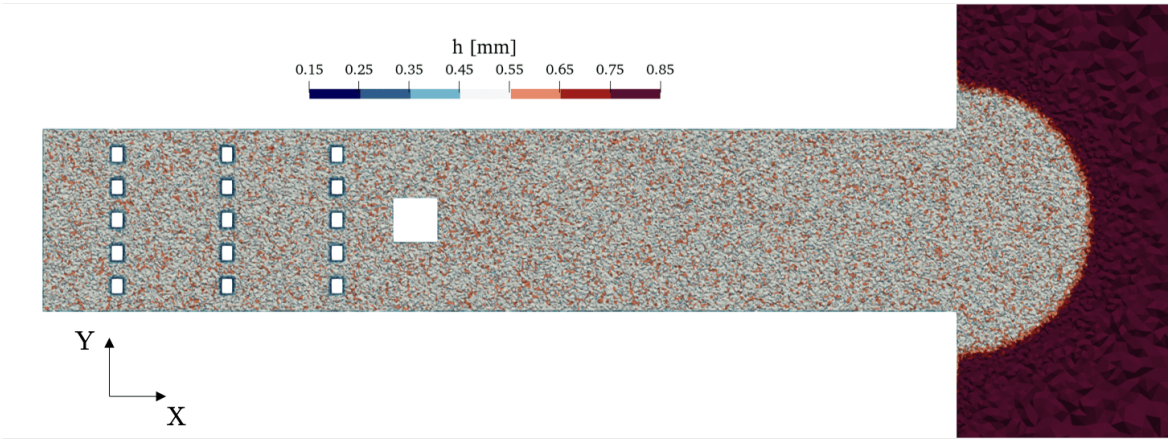


Figure 8.20: Cell size distribution in the Masri chamber. Most of the plenum was truncated for this visualization.

The thickened flame model is used with a target resolution of the thickened flame front $N_c = 5$, corresponding to a thickening factor $F = 6.8$ in the chamber. The dynamic thickening flame sensor defined in Equation 2.113 is based on the reaction rate of the $C_3H_8 + 3.5\,O_2 \longrightarrow 3\,CO + 4\,H_2O$ reaction. It is filtered by 5 successive gather-scatter operations to extend its coverage and ensure that density gradients are well captured.

In the AVBP-DL simulation, the wrinkling factor predicted by the CNN is directly used as an efficiency function in the chamber. A structured mesh of size $500 \times 100 \times 100$ ($h = 500\,\mu m$) coinciding with the chamber is partitioned into 16 equal hexahedral regions. As detailed in Chapter 7, a separate CNN instance is run on each partition and queried to predict Ξ at runtime, with overlaps of size $s = 12$. A simulation is also performed using the CDSM model ($\hat{\Delta} = 1.8\Delta$, $\Delta_m = 2.2\Delta$) and serves as a basis for comparison against the CNN.

The AVBP-DL simulation is run on hybrid CPU/GPU computing nodes of the Jean-Zay supercomputer. A total of 1280 cores are used, split between 1264 cores for AVBP processes and 16 cores for DL processes. This corresponds to the 32 node configuration of distribution D0.5 in the scalability study of Section 7.3. The DL solver is queried every $N = 100$ iterations, so according to Table 7.3, the DL cost is 0.3% of the computational time. The simulation is stopped when the flame reaches the plenum after 8.5 ms of physical time are simulated, amounting to 15k CPUh. Detailed performance reports, including the computational overhead of the DL coupling and comparisons to the overhead of the CDSM model, are found in Chapter 7. Table 8.6 reports all the Masri LES runs which are discussed in the *a posteriori* results.

Name	Model	Description	Reference
CDSM	CDSM	Reference simulation	Section 8.8
AVBP-DL	CNN	Original AVBP-DL run	Section 8.8
AVBP-DL FGI	CNN	Initialization from a CDSM solution at the first grid	Section 8.8.4
AVBP-DL LAM	CNN	Spherical laminar flames in the training set	Section 8.9.1
AVBP-DL VORT	CNN	Resolved vorticity input channel	Section 8.9.2.2
AVBP-DL TEST-FILT	CNN	Test-filtered progress variable input channel	Section 8.9.3

Table 8.6: List of Masri LES simulations.

8.8 *A posteriori* results

8.8.1 Flame propagation

The evolution of the flame front is visualized through five z-normal centerplane slices of efficiency and progress variable for the AVBP-DL (Figure 8.21) and CDSM (Figure 8.22) simulations. Starting from the laminar propagation phase (snapshot 1), it appears that the CNN predicts efficiency values larger than unity in the spherical flame front. Figure 8.23 shows that the highest values of the efficiency are located in the preheat zone, for c values between 0.2 and 0.4. In the reaction zone, typical efficiency values range from 1.5 to 2. This is explained by the absence of laminar curved flame fronts in the training dataset. In the HIT, the flame is initially planar, and curvature of the filtered flame front originates from partially unresolved turbulence-induced curvature of the DNS flame front. Curved filtered flame fronts in the training dataset are therefore systematically associated with non-unity wrinkling factors. As the CNN is aware of the geometry of the flame front, this pattern is replicated in the spherical laminar flame front of the Masri initialization.

Note that the CDSM model can exhibit a similar behavior in small laminar spherical kernels [238]. Because the test-filtering operation destroys highly curved structures that correspond to high wrinkling frequencies, when the curvature radius is of the order of the test-filter size, the β coefficient of the model can be non-zero in a laminar flame. The dynamic model does not inherently distinguish between laminar and turbulent flame fronts: any variation of the flame curvature at the scale of the test-filter size is captured by the model. In this simulation, the radius of the initial burnt kernel is large enough so that β is zero in the laminar flame front. The sensitivity of the CDSM model to highly curved flame fronts explains the slight overshoot in efficiency at the center of the initial CDSM flame kernel, where gradients of c are not quite 0. However, these values are deep into the burnt gases and do not affect the propagation of the flame front.

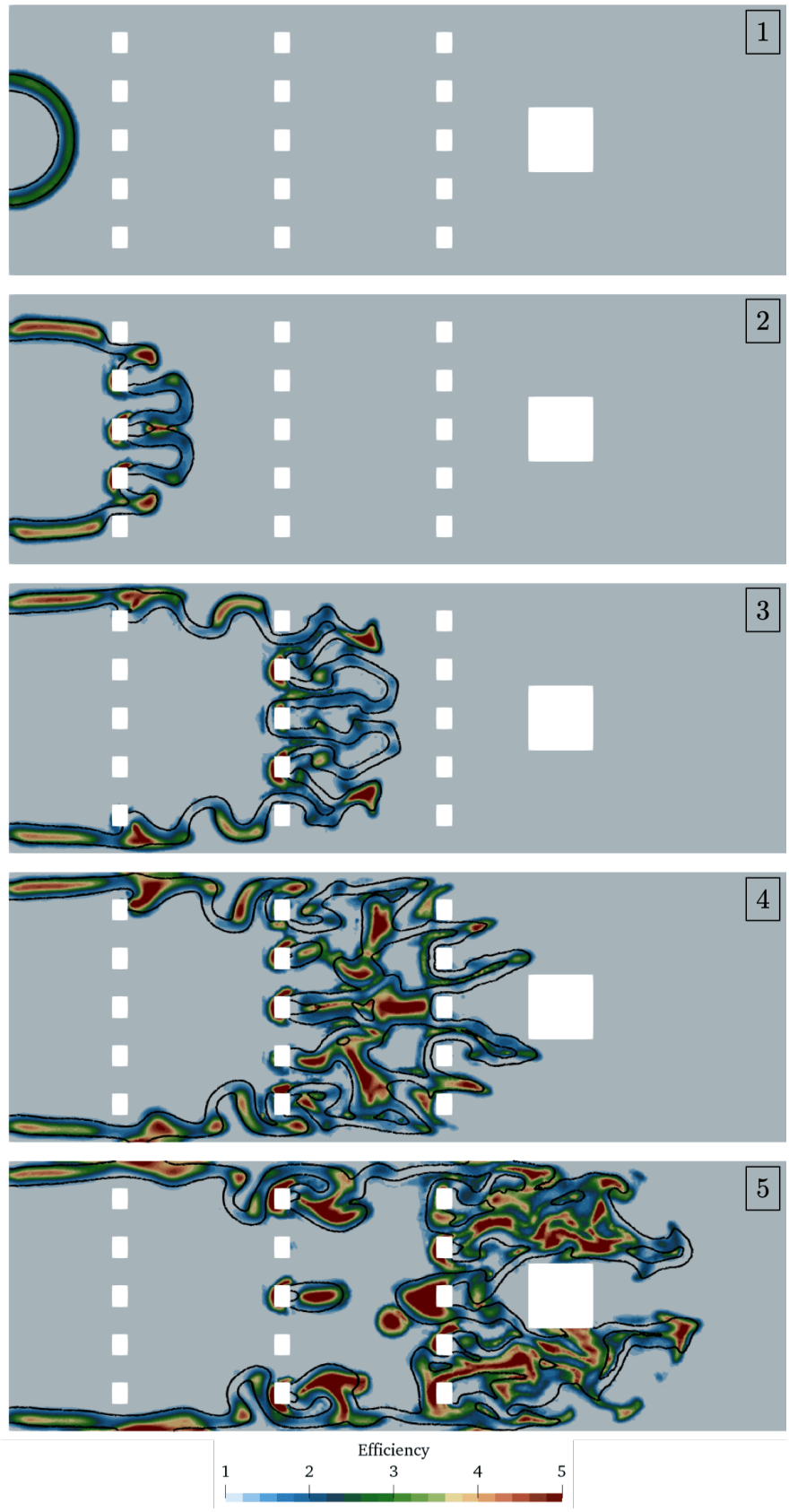


Figure 8.21: Efficiency slices from the AVBP-DL simulation at $t = 0.1, 3, 4.7, 5.2, 5.6$ ms, with iso-lines $c = 0.1$, $c = 0.9$. Snapshots are numbered 1 through 5.

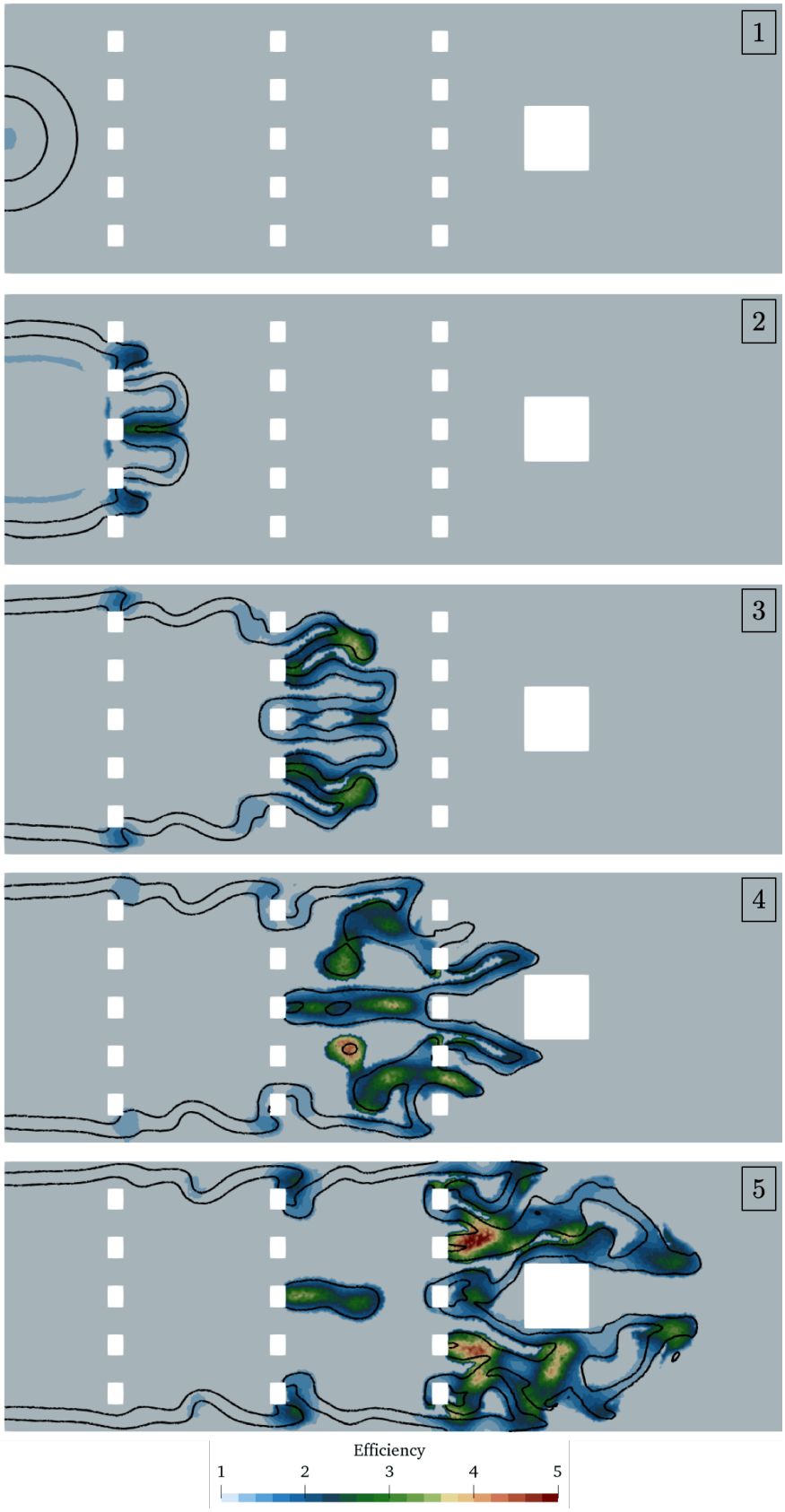


Figure 8.22: Efficiency slices from the CDSM simulation at $t = 0.1, 5.5, 8.1, 8.8, 9.3$ ms, with iso-lines $c = 0.1$, $c = 0.9$. Snapshots are numbered 1 through 5.

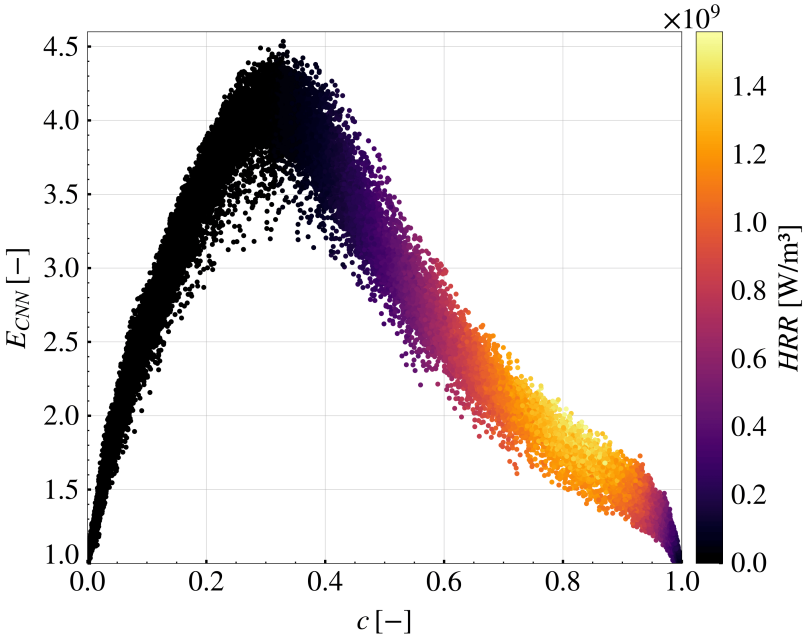


Figure 8.23: Scatter plot of efficiency vs. progress variable colored by heat release rate in the AVBP-DL snapshot 1.

After the flame passes through the first obstacle grid (snapshot 2), similar flame structures are found in the CDSM and AVBP-DL simulations. Efficiency values are higher in the AVBP-DL simulation, especially in the outer flame tips. Merging of the two central finger-like structures happens near the central obstacle of the first grid for the CDSM flame, as the efficiency is uniformly high in the region where the two upper and lower flame fronts interact. For the AVBP-DL flame, the efficiency predicted by the CNN is close to one in the wake of the obstacle, and merging happens further downstream. Artifacts in the AVBP-DL efficiency field are found in the burnt mixture around the three central grid obstacles. They are caused by the imperfect treatment of the progress variable field inside the obstacles for the CNN. Figure 8.24b shows a close-up view around the middle of the first grid of the progress variable field seen by the CNN and the associated efficiency field. As detailed in Section 7.2.2, the progress variable is interpolated by a nearest neighbor procedure inside the obstacles. The transition between the edge of the obstacle and the fluid domain is smooth, but sharp gradients appear inside the obstacle. They trigger high wrinkling predictions, which are interestingly located only in the burnt gases. Although the highest values are localized inside the domain and do not affect the flame, the large receptive field of the CNN makes predictions in the fluid domain near the obstacle sensitive to the discontinuities. Spurious non-unity efficiency values are therefore observed inside the domain, but they are limited to regions with no flame front. Figure 8.24a highlights the absence of any artifact as the flame is propagating through the grid, since the progress variable field on all sides of the obstacles is mostly homogeneous. All in all, spurious artifacts only affect fully burnt regions upstream of the obstacles after the passage of the flame front, and thus have a negligible effect on flame propagation.

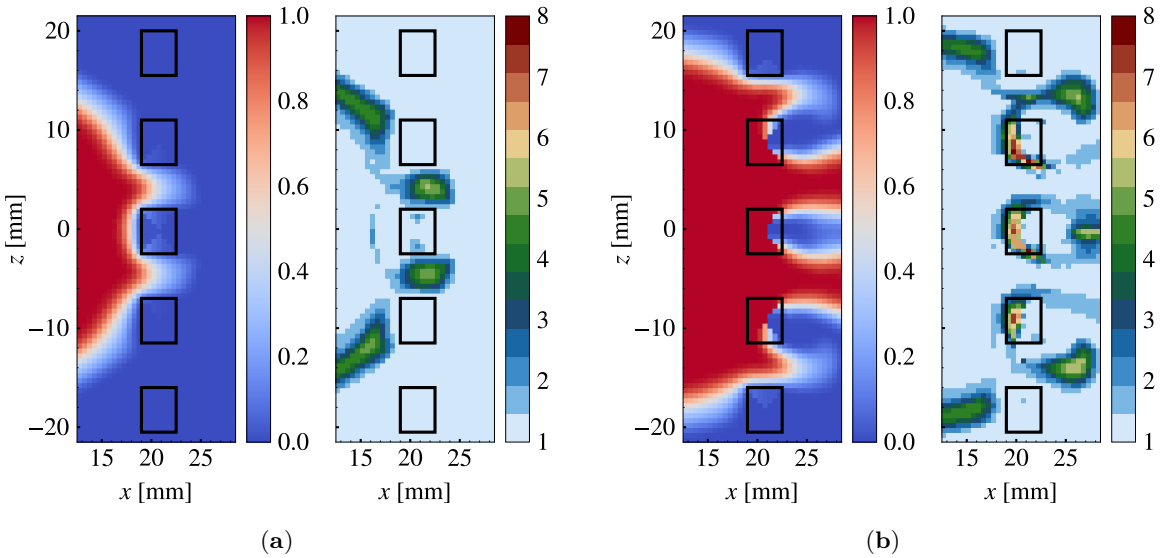


Figure 8.24: Effect of obstacle-induced artifacts on the CNN efficiency field as the flame front is crossing the grid (**a**) and after the flame front has crossed the grid (**b**). Progress variable (left) and efficiency (right) slices on the CNN mesh near the first obstacle grid.

After the second grid (snapshot 3), both flames experience similar resolved wrinkled structures. Outer fingers are curved outwards in the wake of the second grid, and their tip veers back inwards. Efficiency values are particularly high at the tip in both simulations, although more so in the AVBP-DL case. This leads to a smearing of the tip of the outer fingers. In the AVBP-DL flame, high efficiency values are still seen in broadened flame fronts upstream of the second grid. In these regions, only small pockets of fresh gases near the chamber walls remained to be slowly consumed, so efficiency values there are mostly irrelevant to the generation of overpressure. In the CDSM flame, high efficiency values seem correlated to flame-flame interactions behind the obstacles and between the two inner fingers.

After the third grid and the central obstacle (snapshots 4 and 5), notable differences emerge between the AVBP-DL and the CDSM flames. They are further illustrated by comparing progress variable slices in Figure 8.25 and Figure 8.26. In the AVBP-DL flame, large heterogeneities in the efficiency field lead to uneven burning rates and fragmentation throughout the flame front. Unlike in the CDSM simulation, large pockets of unburnt gas are still present between the second and third grids. This is explained by the accumulation of large CNN efficiency values in broadened preheat zones, further broadening these regions without affecting the propagation of the flame which is driven by the efficiency in the reaction zone.

Figure 8.27 shows the distribution of AVBP-DL efficiency values with respect to $|\nabla\check{c}|\check{\delta}_L$. It highlights the correlation of high efficiency values with low gradients of the progress variable. Looking at the same data in the training dataset (Figure 8.28) reveals a similar pattern. The highest wrinkling values are associated with low values of $|\nabla\bar{c}|\bar{\delta}_L$, and the conditional mean peaks at $|\nabla\bar{c}|\bar{\delta}_L \approx 0.3$ like in the Masri simulation. However, Masri $\check{c}$ values there are significantly lower than HIT $\bar{c}$ values. This implies that high CNN efficiencies are more sensitive to gradients of the progress variable rather than its local value.

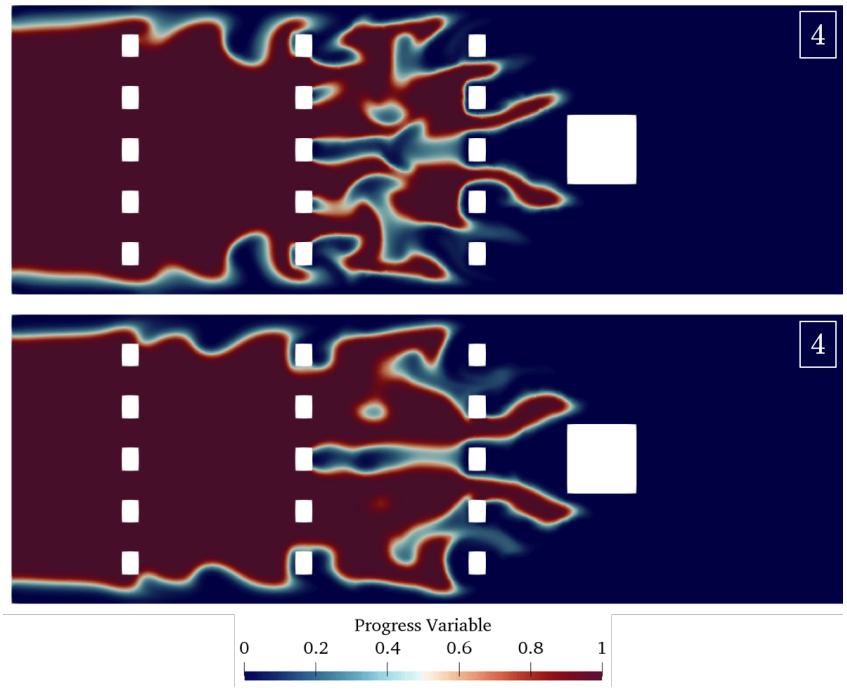


Figure 8.25: Progress variable slices from snapshots 4 of the AVBP-DL (top) and CDSM (bottom) simulations.

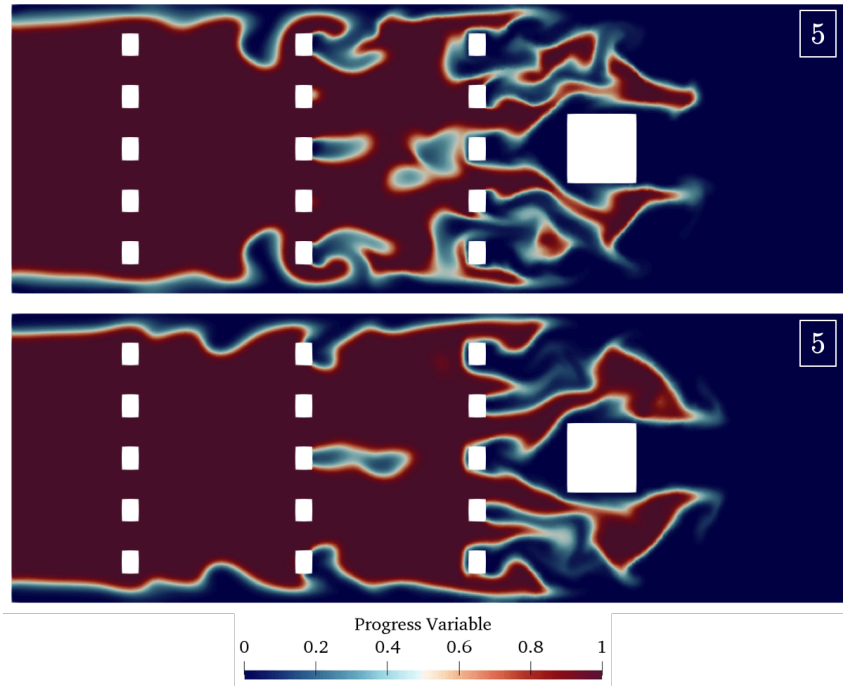


Figure 8.26: Progress variable slices from snapshots 5 of the AVBP-DL (top) and CDSM (bottom) simulations.

For both simulations, the two inner flame fingers reach the central obstacle as the outer fingers are crossing the third grid. AVBP-DL efficiencies are close to unity in the inner fingers, while they are uniformly of the order of 2 for the CDSM model. CDSM efficiencies are also uniformly high in the fresh gas pockets trapped between the second and third grid. Due to the self-interaction of the flame front surrounding these pockets, the dynamic model predicts elevated wrinkling values and the pockets are consumed faster than in the AVBP-DL simulation.

Overall, the CDSM and AVBP-DL flames go through the same stages of flame propagation, with similar resolved structures observed in both simulations. The main differences seem to be induced by excessive AVBP-DL efficiency values in the laminar phase, leading to increased flame acceleration throughout the simulation. Figure 8.29 provides evidence for this with the temporal evolution of the flame tip position, defined as the furthest streamwise location of the $\check{c} = 0.86$ iso-surface corresponding to the progress variable value of peak heat release rate in a laminar flame. This iso-surface is the reference flame surface for all the following post-processings. The AVBP-DL flame reaches the central obstacle 4 ms ahead of the CDSM flame. Increased flame acceleration is further underlined by tracking the velocity of the flame tip in Figure 8.30. The AVBP-DL flame tip is consistently faster than the CDSM and experimental flames until it reaches the wake of the central obstacle, which coincides with the production of peak overpressure. In addition to flame acceleration, greater heterogeneity in the CNN efficiency field results in increased fragmentation of the resolved flame front after the third grid due to differential propagation across the flame front.

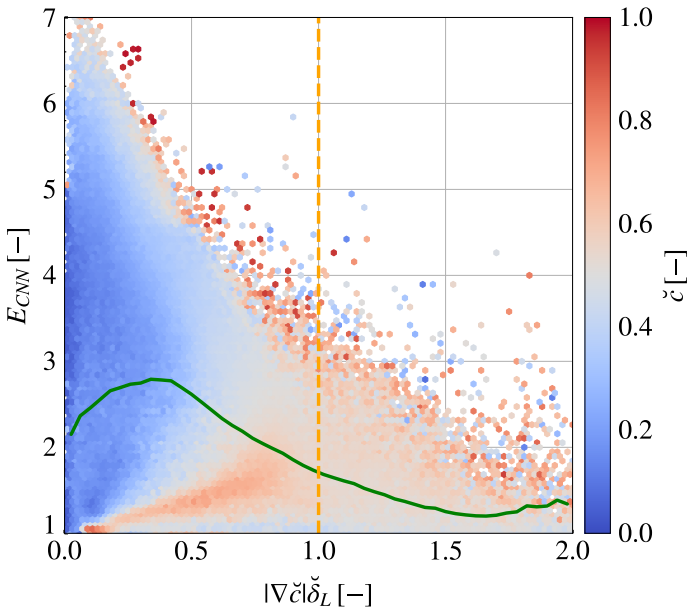


Figure 8.27: Hexbin plot of efficiency vs. $|\nabla\check{c}|\check{\delta}_L$ in the AVBP-DL snapshot 5, colored by mean $\check{c}$ in each bin. Only points where $0.01 < \check{c} < 0.99$ are collected. Green solid line: conditional mean. Yellow dotted line: $|\nabla\check{c}|\check{\delta}_L = 1$.

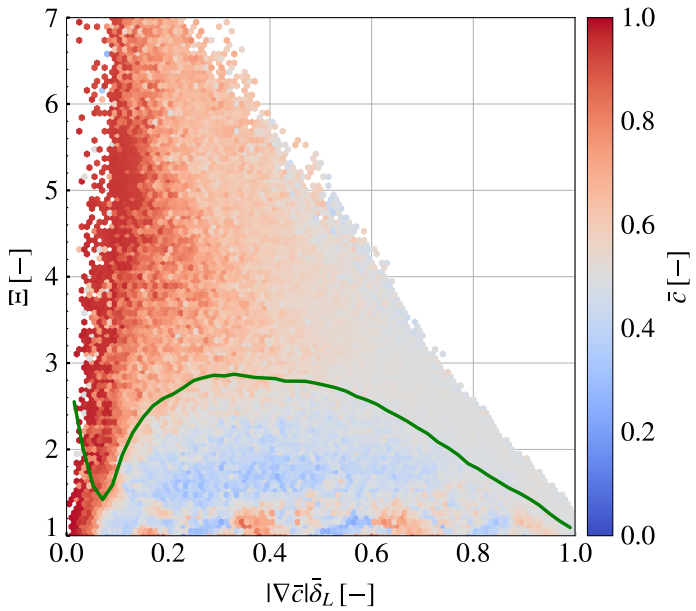


Figure 8.28: Hexbin plot of wrinkling vs. $|\nabla\bar{c}|\bar{\delta}_L$ in the training dataset, colored by mean $\bar{c}$ in each bin. Only points where $0.01 < \bar{c} < 0.99$ are collected. Green solid line: conditional mean.

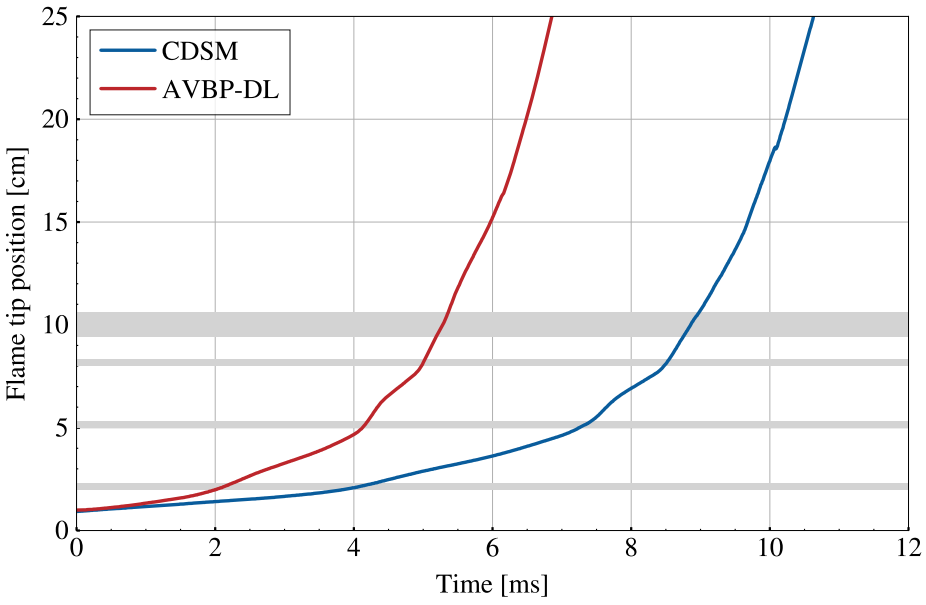


Figure 8.29: Temporal evolution of the flame tip position. The obstacles are represented as grey horizontal areas.

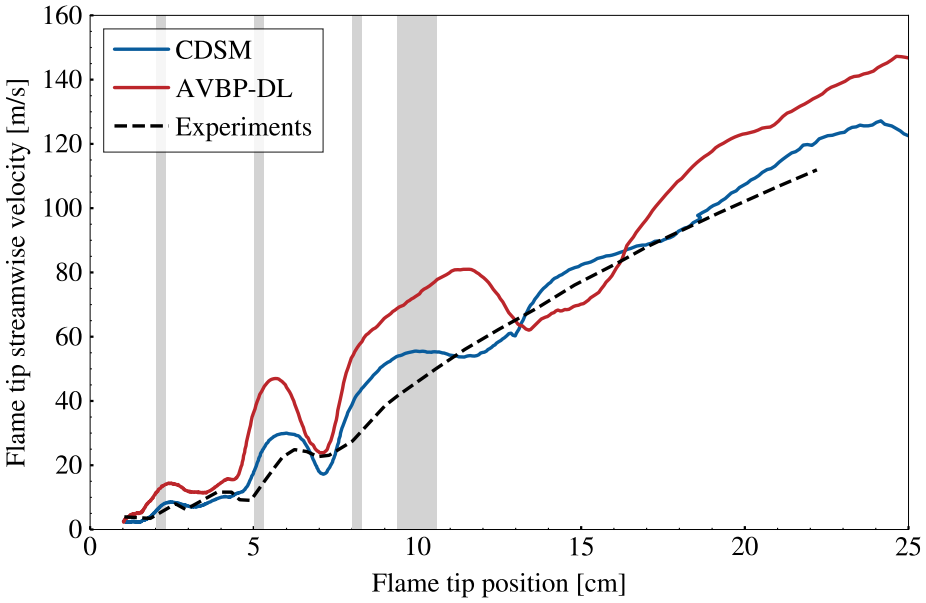


Figure 8.30: Evolution of the flame tip streamwise velocity with respect to the flame tip position. The obstacles are represented as grey vertical areas.

8.8.2 Overpressure

The peak overpressure is the main quantity of interest that must be correctly predicted by the LES simulation. Due to variations in the experimental ignition process, experimental and LES overpressure temporal signals are typically shifted to match the time when the peak overpressure is reached [291, 369]. Figure 8.31 reports the temporal evolution of the time-shifted CDSM and AVBP-DL overpressures. The AVBP-DL simulation predicts a peak overpressure value inside the experimental envelope. Given the increased flame acceleration in the spherical phase, this result can be surprising and may suggest an error compensation in the form of weaker fuel consumption in the later stages of flame propagation. The following sections seek to understand the behavior of the CNN model and compare it with the reference CDSM simulation.

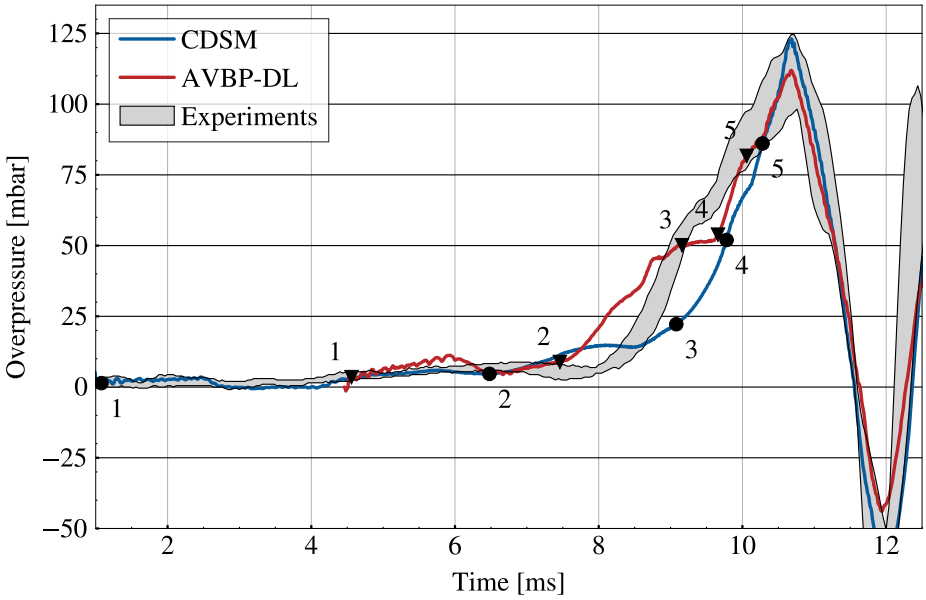


Figure 8.31: Temporal evolution of the overpressure. LES times are shifted to match the time of peak overpressure in the experiment envelope. Snapshots shown in Figures 8.21 (▼) and 8.22 (•) are numbered $1-5$.

8.8.3 Analysis of the turbulent combustion models

Equation 8.1 shows that the overpressure is controlled by the mean heat release rate in the chamber. This quantity is proportional to the total flame surface if the flame front has a flamelet structure, as discussed in Section 2.5. This is verified by plotting the JPDF of the heat release rate and the progress variable for snapshot 5 in Figure 8.32. Despite the large amount of turbulence-induced scatter in the distribution, the conditional mean heat release rate $\langle \dot{\omega}_T | c \rangle$ follows the profile of a laminar flamelet. The interpretation is that the state relations of turbulent flame elements do not deviate from flamelet state relations on average, hence the validity of the flamelet assumption for this flame [77]. Turbulence therefore primarily affects the flame by wrinkling the flame front, and the mean heat release rate is mainly driven by the increase of the flame surface as the flame propagates through the

domain. The rest of this section investigates the effect of the turbulent combustion models on the total flame surface.

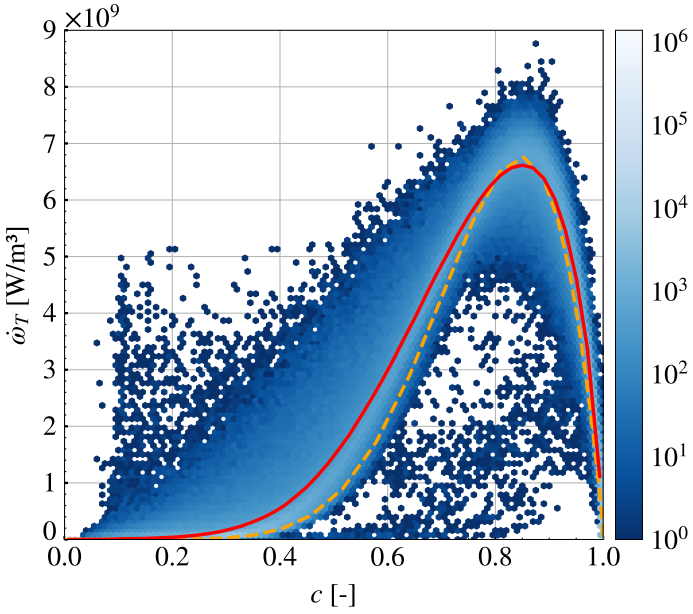


Figure 8.32: Hexbin plot of heat release rate vs. progress variable in the AVBP-DL snapshot 5. Red solid line: conditional mean. Orange dashed line: flame structure of the corresponding laminar flame. In the turbulent heat release rate, the efficiency and the thickening are removed to enable the comparison to the laminar data, *i.e.* $\dot{\omega}_T = \dot{\omega}_T^{AVBP} \times F/E$.

Figure 8.33 plots the evolution of the resolved and total flame surfaces with respect to the flame tip position. The resolved flame surface is evaluated as the surface area of the reference $\check{c} = 0.86$ iso-surface. The total flame surface is the integral of the total flame surface density $E|\nabla\check{c}|$. Before the central obstacle, resolved flame surfaces in the CDSM and AVBP-DL simulations are nearly identical. Increased efficiency values in the initial phases of the AVBP-DL run lead to a moderately greater flame surface until the third grid. After the central obstacle, both resolved and total AVBP-DL flame surfaces reach a plateau, while the CDSM values continue to rise. This may explain why the two peak overpressures are the same despite the early acceleration of the AVBP-DL flame. Peak overpressure is reached as the flame tip goes past the central obstacle, and at this moment the abated rise of the AVBP-DL total flame surface compensates the effects of early flame acceleration.

The contribution of the efficiency models to the total flame surface is further investigated through the evolution of the proportion of SGS flame surface in the total flame surface, defined as $\int(E-1)|\nabla\check{c}|\,\mathrm{d}V$ and plotted in Figure 8.34. The contribution of the CDSM model is low in the laminar phase as the resolved flame front is weakly curved. It increases over the course of flame propagation until it reaches the same level as the CNN after the central obstacle. Meanwhile, the CNN does not discriminate between laminar and turbulent regimes and is consistently responsible for 30% to 40% of the total flame surface. Crucially, this contribution does not follow the increased turbulence levels seen by the propagating flame tip.

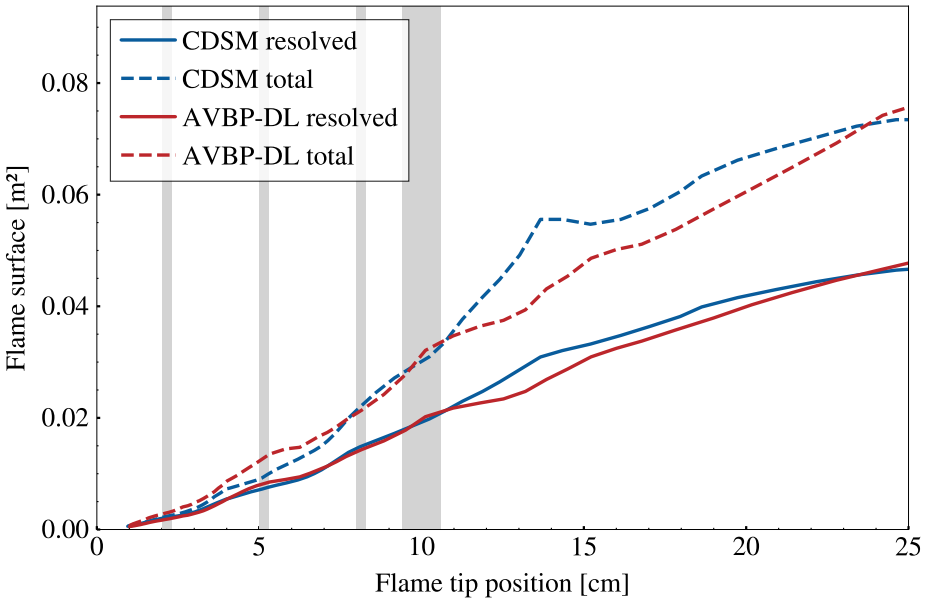


Figure 8.33: Evolution of the resolved and total flame surfaces with the flame tip position.

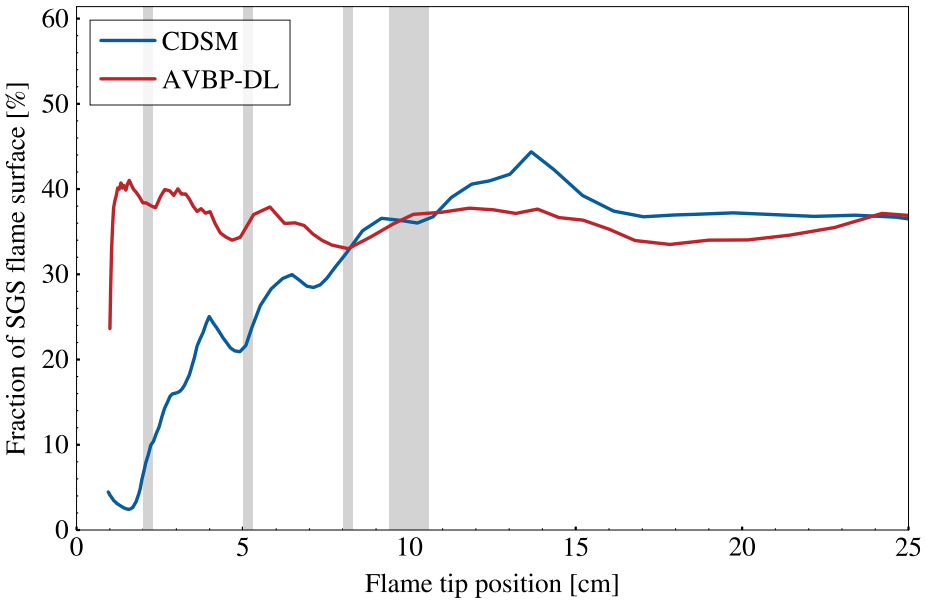


Figure 8.34: Evolution of the fraction of SGS flame surface with the flame tip position.

Despite the large efficiency values observed throughout the flame in Figure 8.21, the comparable resolved and total flame surfaces observed in Figure 8.33 indicate that efficiency values in the reaction zone are not much higher than in the CDSM simulation. This is supported by visualizations of the efficiency field on the reference isosurface in Figures 8.35 and 8.36. In addition, the efficiency distribution over the iso-surface at the same timestamps is plotted in Figure 8.37. Over the entire flame, the CDSM distribution has a clear mode

at unity efficiency corresponding to laminar flame fronts in the far end of the chamber. For the AVBP-DL flame, these regions correspond to a second mode around $E = 1.5$, once again outlining the excessive predictions of the CNN there. When limiting the distributions to the portion of the domain after the second grid to eliminate these laminar regions, it is striking to note that a significant portion of the turbulent flame front is assigned unity efficiency values by the CNN. On the other hand, sections with an efficiency larger than 2 constitute a greater proportion of the turbulent streamwise propagating flame front in the CDSM simulation than in the AVBP-DL case. This confirms that the CDSM model is predicting a greater amount of SGS flame surface at the leading point of the flame, which compensates for the slower flame tip velocity and leads to the same peak overpressure as the AVBP-DL simulation.

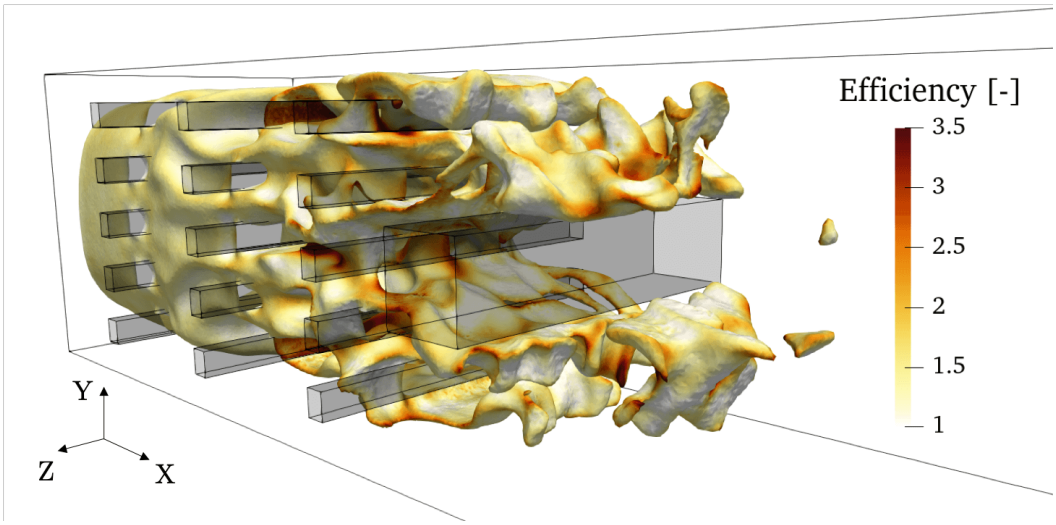


Figure 8.35: Iso-surface of $c = 0.86$ colored by efficiency in the AVBP-DL snapshot 5.

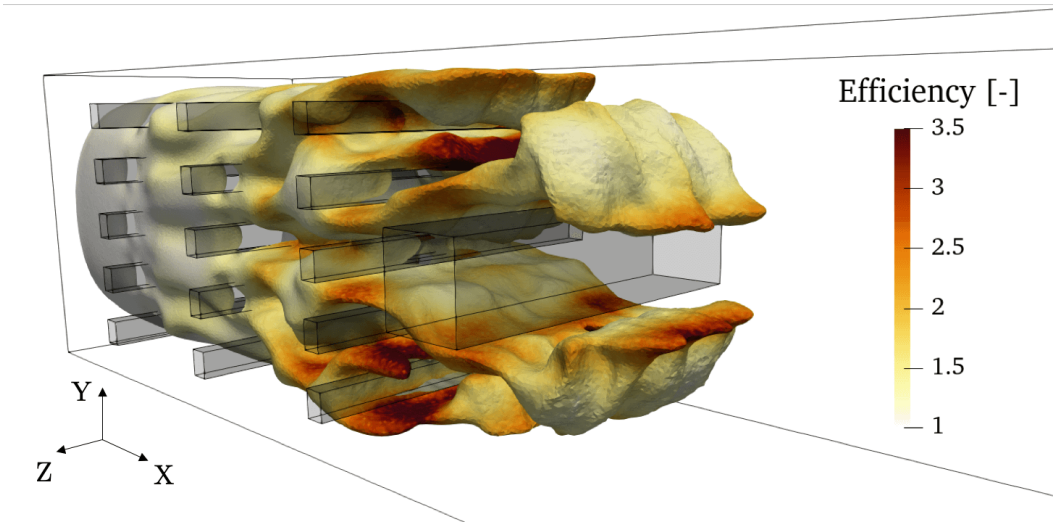


Figure 8.36: Iso-surface of $c = 0.86$ colored by efficiency in the CDSM snapshot 5.

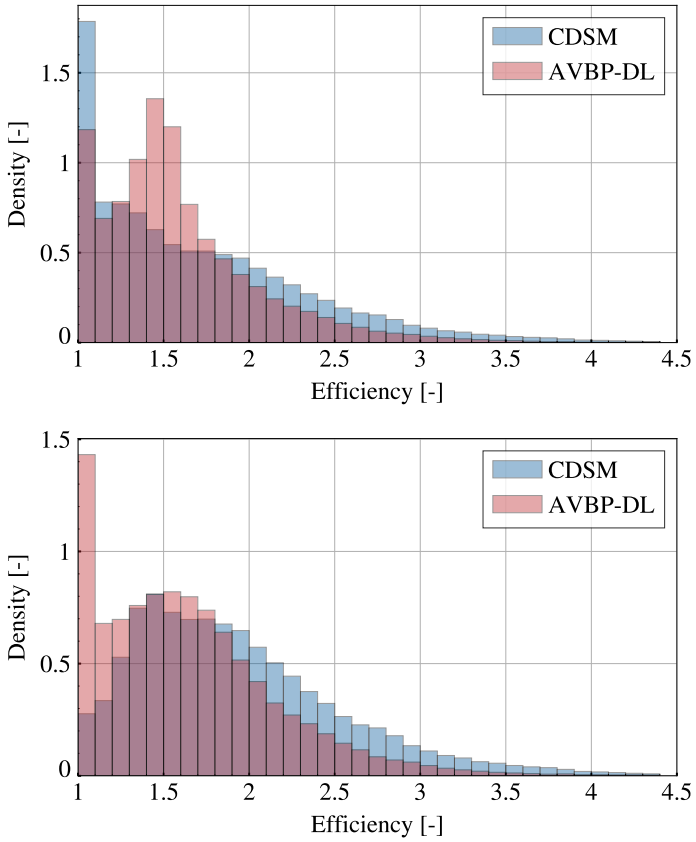
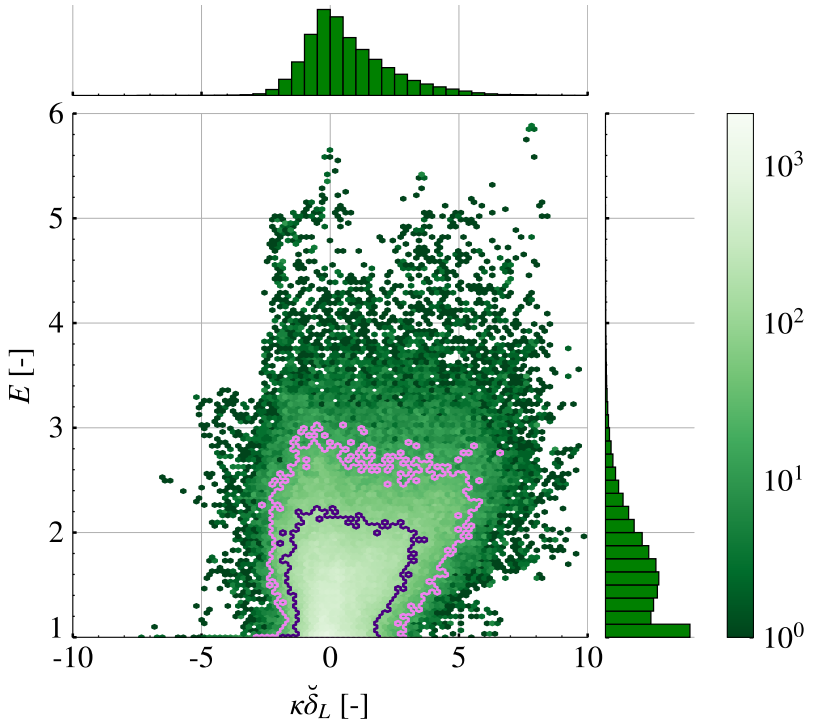


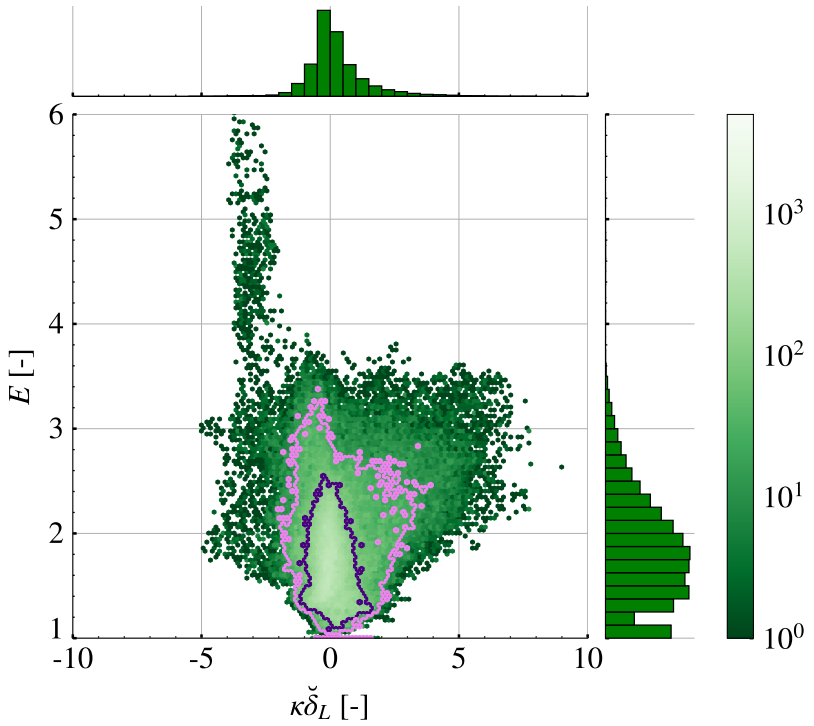
Figure 8.37: Density histogram of efficiency in snapshots 5 on the reference iso-surface at the same flame tip position (after the passage of the central obstacle) for the whole flame (top) and after the second grid only (bottom).

Finally, the correlation between efficiency and flame curvature is investigated in Figure 8.38 for the portion of the reference iso-surface located after the second grid. The JPDF shows that in both AVBP-DL and CDSM flames, the variance of the curvature conditioned on E grows with the value of E. This means that large efficiency values are distributed over a larger range of curvatures than small efficiency values. It is indeed fair to assume that highly curved resolved flame fronts should be assigned high efficiency values, as they are wrinkled by turbulent eddies of the size of the thickened flame front which may also generate unresolved wrinkling.

While unity efficiencies are concentrated only in low curvature regions for the CDSM flame, they are spread across a relatively larger extent of curvatures for the AVBP-DL flame. For the CDSM flame, curvature in the resolved flame front will lead to different resolved and test-filtered flame surface densities, and thus an efficiency value higher than one. This does not apply to the CNN model which can predict unity efficiency values even in highly curved flame elements. It is unclear whether this behavior is strictly beneficial or detrimental to the accuracy of the model, but it is a clear difference that separates the two models.



(**a**) AVBP-DL



(**b**) CDSM

Figure 8.38: Hexbin plot of efficiency against normalized curvature colored by bin count in the AVBP-DL (**a**) and CDSM (**b**) snapshot 5 on the reference iso-surface after the second grid. Marginal histograms are plotted on the side. Contours delineate bins containing more than 20 (light pink) and 100 (dark purple) samples.

It appears that AVBP-DL efficiencies are moderately correlated with the curvature, whereas such a correlation is less clear for the CDSM model from the vertical orientation of its purple and pink density isocontours. This correlation is self-enhancing, as illustrated in Figure 8.39. Positively curved flame elements receive higher efficiencies and propagate faster towards the fresh mixture, thus further increasing their curvature. In Figure 8.38a, this phenomenon is evidenced by the large extent of the marginal curvature distribution in the positive values compared to the same distribution for the CDSM flame in Figure 8.38b. It could explain the fragmented nature of the AVBP-DL flame front observed in Figure 8.35.

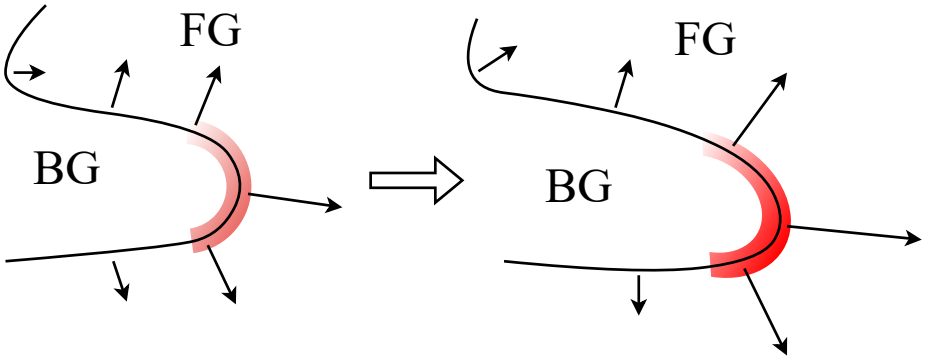


Figure 8.39: Self-enhancement of the high efficiency/positive curvature correlation. Illustrative efficiency magnitudes are shaded in red. BG: burnt gases, FG: fresh gases. Black arrows indicate the direction of local flame propagation.

8.8.4 Initializing the AVBP-DL simulation from the first grid

Results from the AVBP-DL simulation highlight that although the peak overpressure lands inside the experimental envelope, one key area of concern is the increased initial flame acceleration due to non-unity efficiencies in the laminar phase. In the following, a new AVBP-DL simulation is initialized from a CDSM snapshot right before the flame crosses the first obstacle grid as illustrated in Figure 8.40. This ensures that the laminar propagation phase has already been correctly represented, and the CNN is used as a SGS combustion model only when the flame is known to become turbulent. This new simulation is denoted AVBP-DL FGI (First Grid Initialization).

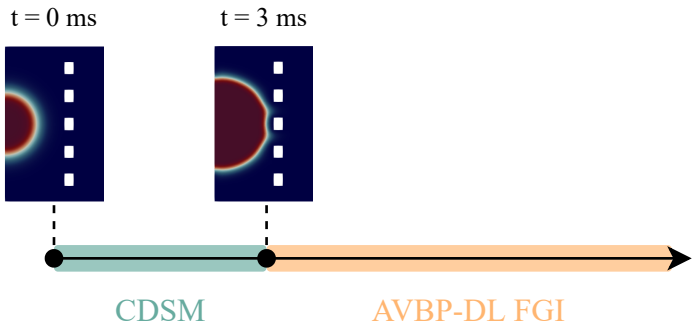


Figure 8.40: Initialization of the AVBP-DL FGI run.

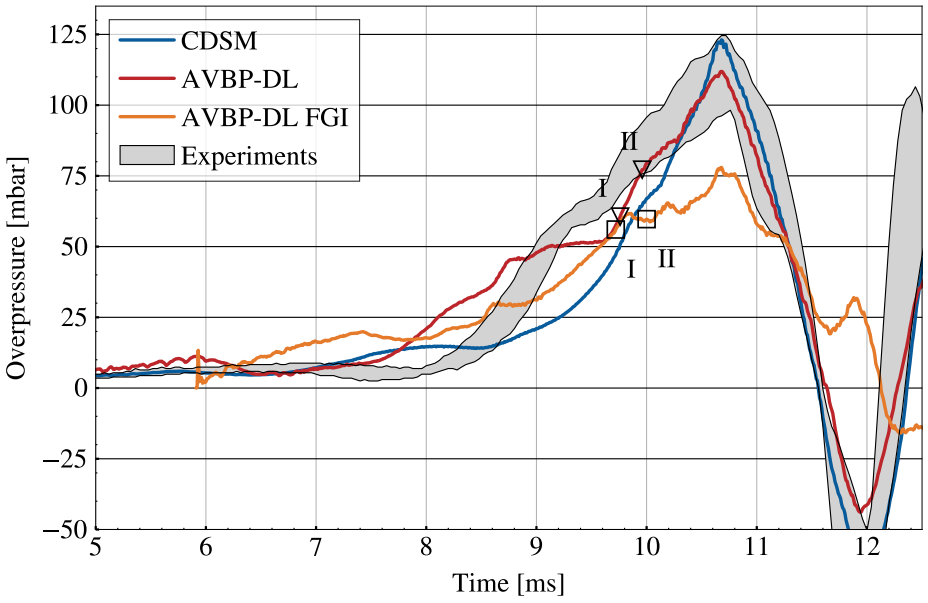


Figure 8.41: Temporal evolution of the overpressure, including the AVBP-DL FGI run. LES times are shifted to match the time of peak overpressure in the experiment envelope. Snapshots shown in Figures 8.45 (▽) and 8.46 (□) are marked I and II.

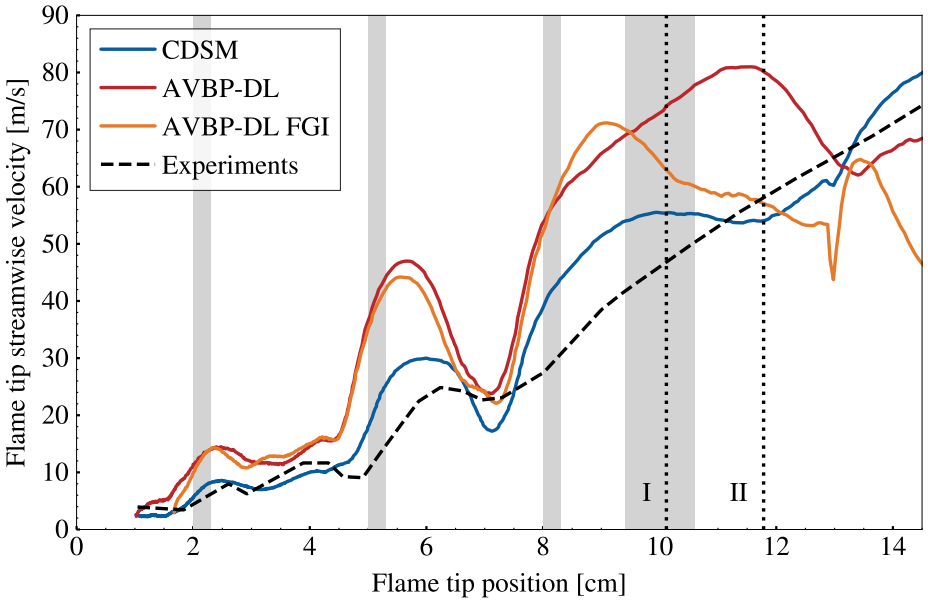


Figure 8.42: Evolution of the flame tip streamwise velocity with respect to the flame tip position, including the AVBP-DL FGI run. Dotted lines mark the flame tip position for the snapshots of Figures 8.45 and 8.46.

Figure 8.41 shows the new overpressure time signal along with the two previous ones. This time, the overpressure does not rise sharply, and instead reaches a plateau before hitting its maximum value. This leads to a peak overpressure value which is 30% lower than the

previous AVBP-DL simulation, and is outside of the experimental range.

One candidate explanation for the lower peak overpressure value is that the flame front does not propagate as fast as in the original AVBP-DL simulation before reaching the central obstacle, since it does not benefit from the acceleration in the initial spherical phase. Plotting the flame tip streamwise velocity (Figure 8.42) reveals that this is not the case. In fact, the AVBP-DL flame front initialized after the laminar phase catches up with the propagation speed of the original AVBP-DL flame soon after the first grid. However, major differences emerge between the third grid and the central obstacle. After reaching the central obstacle, the flame tip velocity significantly drops below the trend of the original AVBP-DL flame.

To investigate the cause of this phenomenon, the spatial evolution of the flame surface is plotted in Figure 8.43. In the AVBP-DL FGI run, the resolved and total flame surfaces stabilize when the flame is located between the third grid and the central obstacle. As a comparison, a plateau is also observed in the original AVBP-DL simulation but only after the passage of the central obstacle, and the CDSM flame surfaces steadily rise until well after the central obstacle.

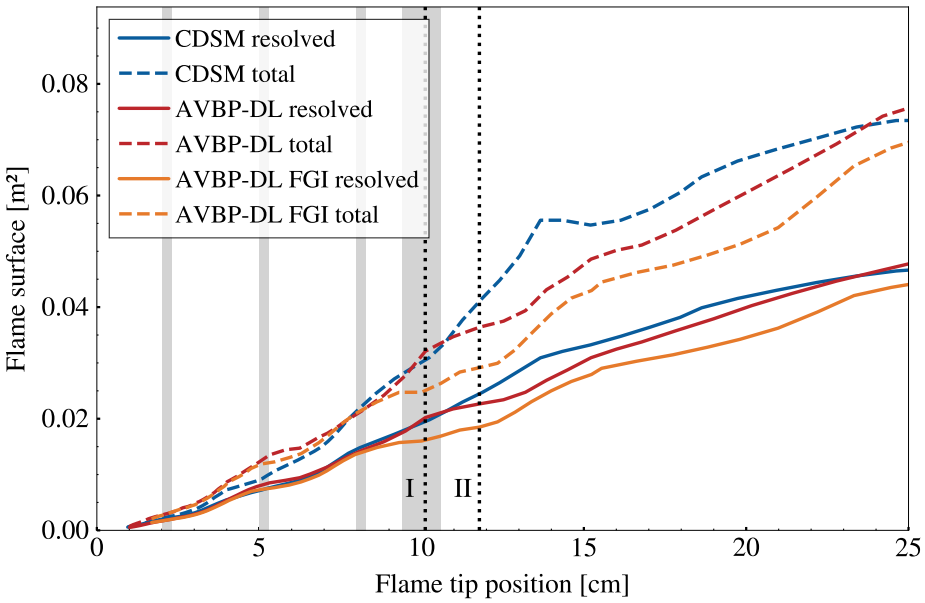


Figure 8.43: Evolution of the resolved and total flame surfaces with respect to the flame tip position, including the AVBP-DL FGI run. Dotted lines mark the flame tip position for the snapshots of Figures 8.45 and 8.46.

The impact of the diminished increase in flame surface on the overpressure generation is quantified by the temporal growth rate of the mean heat release rate in the chamber $\partial\langle\dot{\omega}_T\rangle/\partial t$. From Equation 8.1, it is evident that this quantity drives the rise and fall of the overpressure, higher values that are not immediately compensated by the venting term being responsible for an increase in overpressure. Figure 8.44 shows that this term significantly drops in the AVBP-DL FGI simulation as the flame tip reaches the central obstacle. In particular, two distinct troughs appear and are marked by snapshots noted I and II corresponding to two distinct flame tip positions for each simulation. They correspond to times when the two

overpressure signals start to diverge (Figure 8.41).

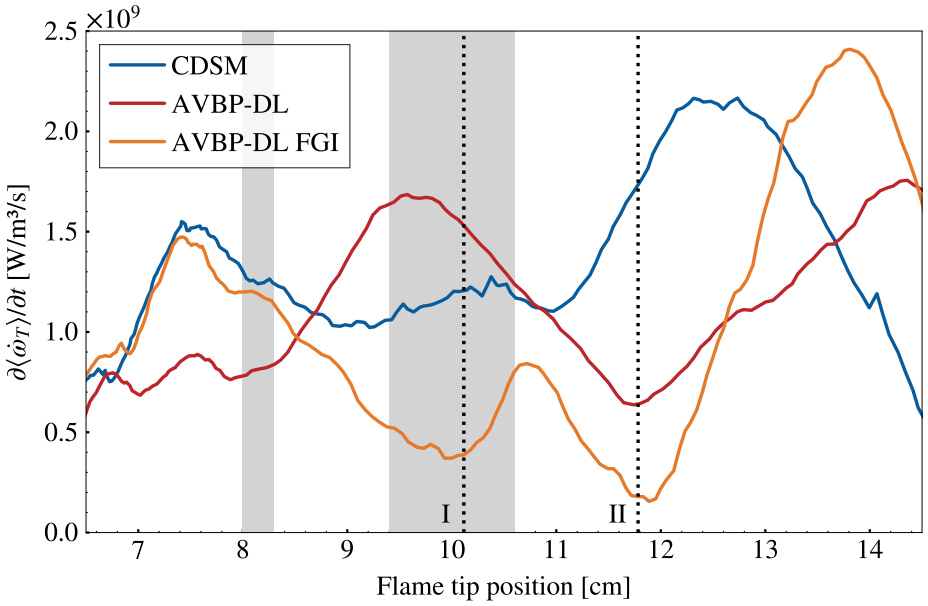


Figure 8.44: Temporal growth rate of the mean heat release rate in the chamber as a function of the flame tip position. Dotted lines mark the flame tip position for the snapshots of Figures 8.45 and 8.46.

Comparative visualizations of the progress variable field for snapshots I and II are shown in Figures 8.45 and 8.46. In snapshot I, compared to the original AVBP-DL flame, the flame front initialized from the first grid features smaller unburnt pockets between the second and third grid. In addition, the propagation of the outer flame fingers is delayed as they have not yet crossed the third obstacle grid. Finally, the tip of the inner flame fingers has detached from the main body of the flame. This last point has lasting consequences on flame propagation. In snapshot II, the flame tips have nearly completely vanished, leaving a lagging reaction front located around the start of the central obstacle along with partially burnt areas propagating ahead of it. This explains the stagnation of the resolved flame surface observed in Figure 8.43 and the resulting overpressure plateau. In the CDSM simulation and past LES results (Figure 8.2), a key driver in the burning rate after the central obstacle is the uneven propagation of the central and outer flame fingers, which creates outer pockets of fresh gases that are swiftly consumed as the outer fingers catch up to the center. This cannot occur here in the absence of well-defined central flame fingers.

Results from the AVBP-DL FGI simulation therefore seem to be explained by variations in the resolved flame structure as it interacts with the obstacles. These variations are not compensated by the turbulent combustion model which predicts insufficient wrinkling values during the critical stages of turbulent flame propagation, *i.e.* after the flame tip crosses the central obstacle. In the rest of this chapter, improvements to the CNN are sought in order to fix the issues outlined in this analysis.

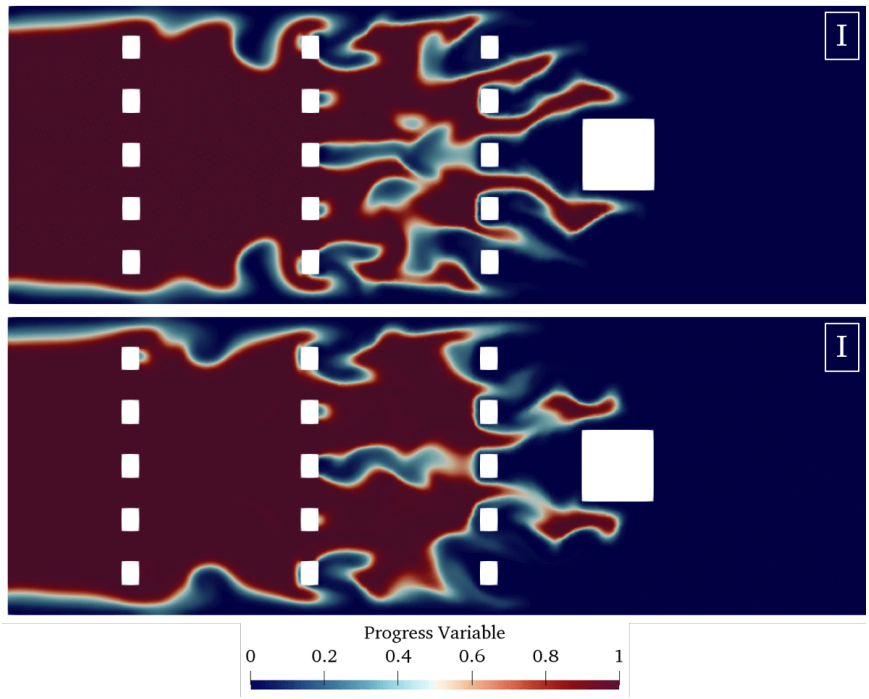


Figure 8.45: Progress variable slices from the AVBP-DL (top) and AVBP-DL FGI (bottom) simulations for snapshot I.

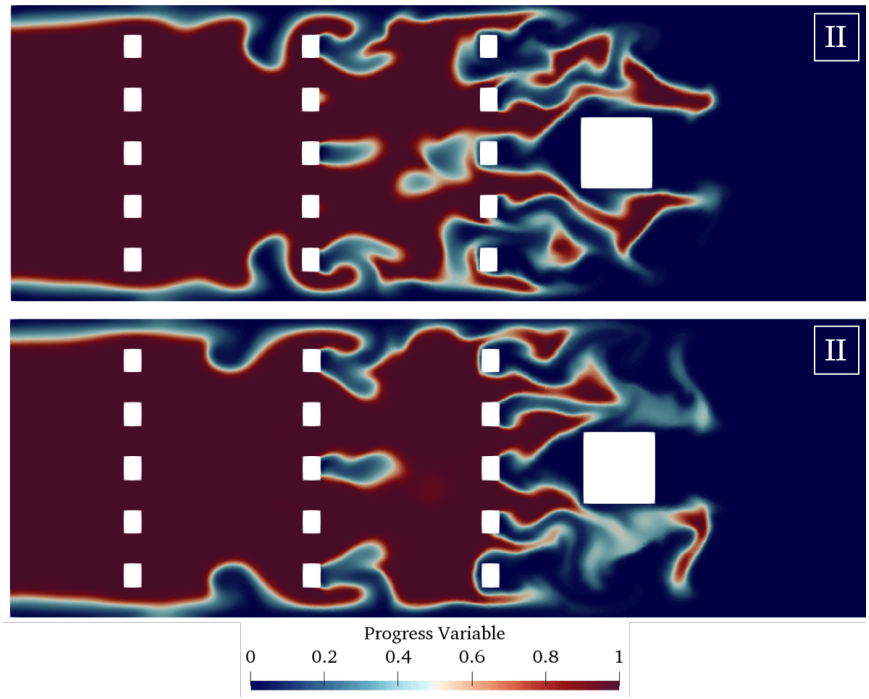


Figure 8.46: Progress variable slices from the AVBP-DL (top) and AVBP-DL FGI (bottom) simulations for snapshot II.

8.9 Search for an improved CNN model

8.9.1 Addition of laminar flames to the training set

A specificity of the Masri configuration is the initial laminar regime of the expanding ignited sphere, until the first grid is reached. To encourage the network to predict a unity SGS wrinkling factor in this phase, 7 snapshots of thickened laminar spherical fronts are added to the training dataset, with a target wrinkling field that is uniformly unity. They are obtained by running the TFLES of an expanding burnt gas kernel using the same chemistry and transport as the HIT, and the thickening parameters of the reference Masri TFLES. The radius of the 7 kernels ranges from 8 mm to 22 mm, which covers the whole range of radii of the laminar phase in Masri. Similarly, a snapshot of a planar flame front with unity target wrinkling is added to the training dataset.

The CNN trained on the HIT dataset is fine-tuned on this new composite dataset until convergence for the laminar flames is reached, after about 20 epochs. Results on the training set contained in Appendix C show that this does not affect the validation accuracy of the model. The consequences in the Masri simulation are shown in Figure 8.47. While the CNN now predicts the correct wrinkling in the laminar phase, it underpredicts wrinkling in weakly curved flame fronts after the first obstacle compared to the original AVBP-DL simulation (compare with Figure 8.21). This may be the reason for the lower peak overpressure reached in this simulation (Figure 8.48) as flame acceleration is not as strong. The peak overpressure falls short from the experimental range by a factor 2. This may again be due to the inability of the CNN to distinguish between laminar and turbulent flame fronts.

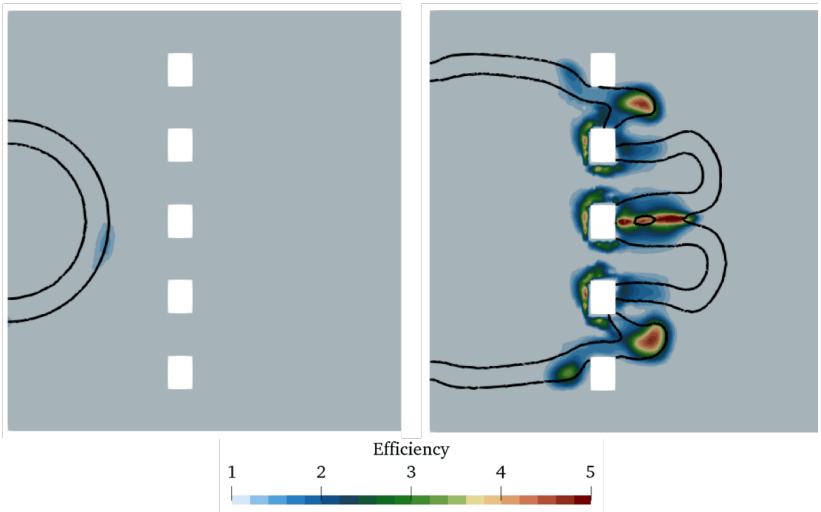


Figure 8.47: Efficiency slices from the AVBP-DL LAM simulation at $t = 0.1$ ms (left) and $t = 4.6$ ms (right), with iso-lines $c = 0.1$, $c = 0.9$.

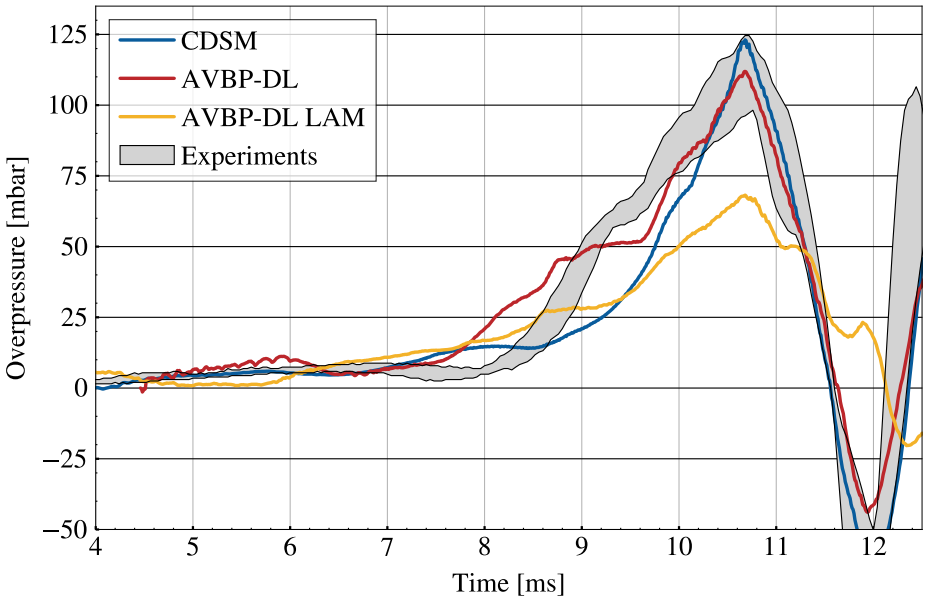


Figure 8.48: Overpressure vs. time for the AVBP-DL LAM simulation. LES times are shifted to match time of peak overpressure in the experiment.

8.9.2 Turbulence input channel

To address this issue, a natural avenue for improvement is to supply the CNN additional information on the flow turbulence. This can be achieved in two ways:

- providing an estimate of the subgrid-scale velocity fluctuation intensity u',
- providing a resolved turbulence field such as the vorticity $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$.

8.9.2.1 SGS turbulence intensity input

The first option relies on an accurate SGS model for u', like in the Colin and non-saturated Charlette wrinkling models. Estimating u' is known to be challenging in a practical LES. Langella et al. [181] benchmark u' models against filtered DNS data. Scale-similarity models for the turbulent velocity [282] and turbulent kinetic energy [21], an eddy viscosity model [194], the Colin et al. model [67], and a new dissipation-diffusion model are compared on their ability to recover the mean SGS kinetic energy on a DNS of a statistically planar turbulent premixed flame. All the models rely on constants which can be fitted to best match the target SGS kinetic energy. The authors note that the ideal model constant depends on the Reynolds number, filter size, and combustion state (products, reactants, or inside the flame front). Importantly, all these values differ from the base model constants provided in the original papers and which were fitted on a different configuration. This work therefore highlights the lack of robust u' models for arbitrary reacting flows.

Furthermore, a good u' model for SGS wrinkling models must solely capture fluctuations that generate an increase of flame surface. In reacting flows, SGS velocity fluctuations contain a dilatational component due to thermal expansion in the flame front, even in a

laminar flame. Colin et al. purposefully discard this component to build an irrotational formulation for the turbulence intensity between the Kolmogorov scale and a cutoff scale $\Delta_e = 10h$ which is their estimate for a typical thickened flame thickness $F\delta_L$. The model writes [67]:

$$u' = c_2 h^3 \nabla^2 (\nabla \times \boldsymbol{u}) \,, \tag{8.7}$$

with $c_2 = 2$ fitted to recover the kinetic energy between the Kolmogorov scale and Δ_e on a non-reacting homogeneous isotropic turbulence DNS. The rotational operator ensures that u' is zero through a laminar flame front, as only the solenoidal component of the velocity field is retained. The authors observe little variation of the optimal c_2 with Reynolds numbers ranging from 10^2 to 10^5, which seems to contradict the findings of Langella et al. discussed above. This may be explained by the generation of potential velocities by the flame front which would be present in the reacting case of Langella et al. and not in the non-reacting HIT of Colin et al. Recently, weakly turbulent premixed flames have been observed to generate non-negligible anisotropic potential velocity fluctuations upstream of the flame front [317]. These fluctuations perturb the incoming turbulence seen by the leading edge of the flame and may therefore affect wrinkling. Additionally, DNS analysis of low and high Karlovitz jet flames indicates that heat release effects on incoming turbulence decrease as the Karlovitz number increases [215]. This further complicates the task of finding a u' model that correctly accounts for turbulence-flame interactions.

In light of these difficulties, some wrinkling models like the CDSM model eschew a dependency of u' and were shown to capture the temporal and spatial evolution of SGS wrinkling on a turbulent swirled flame in *a priori* and *a posteriori* testing [371]. The strong performance of the model suggests that an accurate SGS wrinkling model can be found without including u' in its formulation. This work therefore adopts the same modeling strategy and avoid using unreliable u' models as input to the CNN. Instead, information on the resolved turbulence in the form of the resolved vorticity field is fed to the CNN.

8.9.2.2 Resolved vorticity input

A normalized vorticity magnitude field $\omega^+ = ||\nabla \times \boldsymbol{u}||\delta_L/s_L$ is here used as a second input channel for the CNN. The normalization factor δ_L/s_L is a characteristic flame time that provides some degree of generalizability regarding the type of flame. Masking the ω^+ field to remove vorticity values outside the flame front was empirically found to slightly improve performance on the validation set. Masking is applied using the same flame sensor S as for the wrinkling factor (Equation 8.3).

The performance metrics in Appendix C indicate that it performs well on the validation dataset. However, the AVBP-DL VORT simulation shows that this solution is not able to solve the problems of the original AVBP-DL simulation. It appears that the model is unable to predict unity wrinkling in the laminar phase (Figure 8.49), and worse, a sudden increase in overpressure after the third row of obstacles leads to a numerical crash (Figure 8.50).

Investigating the efficiency and ω^+ fields at this instant (Figure 8.51) reveals that ω^+ values are much higher than in the training dataset, inducing very high efficiency predictions by the CNN. Vorticity values in the Masri configuration are naturally much higher than in the HIT since the velocity of the flow reaches higher values, around $100\,\mathrm{m\,s^{-1}}$ at peak flame

acceleration. Turbulence metrics that generalize better should therefore be sought in further investigations to robustly integrate turbulence information in the CNN.

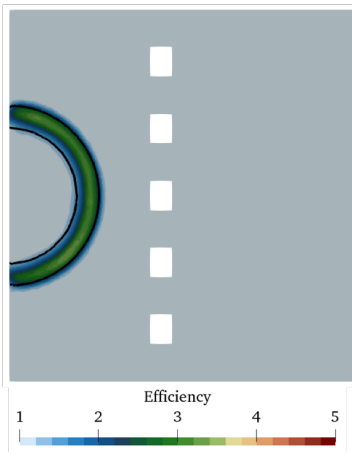


Figure 8.49: Efficiency slice from the AVBP-DL VORT simulation at $t = 0.1$ ms, with iso-lines $c = 0.1$, $c = 0.9$.

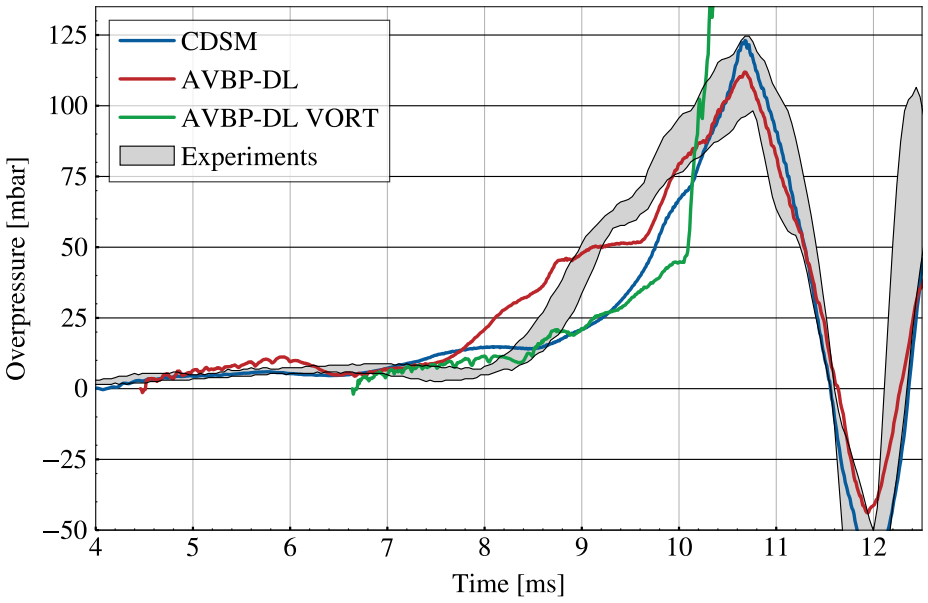


Figure 8.50: Overpressure vs. time for the AVBP-DL VORT simulation. LES times are shifted to match time of peak overpressure in the experiment.

8.9.3 Test-filtered progress variable input channel

Finally, since the CDSM model uses the test-filtered progress variable as a key additional input to derive the wrinkling, an idea could be to introduce this variable as an input to the CNN. It is interesting to dwell upon some differences in the modeling heuristics of the CNN and CDSM models.

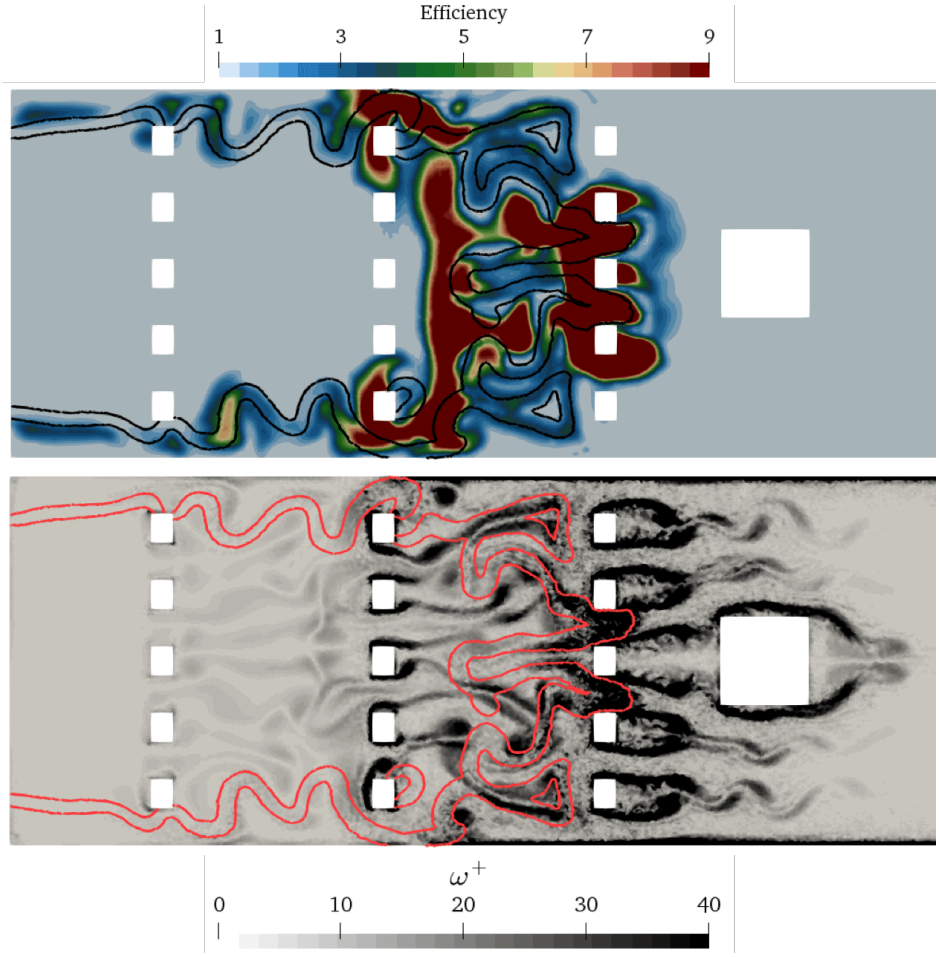


Figure 8.51: Efficiency and ω^+ slices from the AVBP-DL VORT simulation at $t = 3.8\,\text{ms}$, with iso-lines $c = 0.1$, $c = 0.9$.

The CDSM model relies on a strong physical prior: the relation between the SGS flame surface and the filter scale is logarithmic up to a test-filter scale (Figure 2.10). The comparison of the filtered and test-filtered FSDs is then sufficient to deduce the SGS flame surface. One may note that fractal theory predicts a logarithmic relationship between turbulent surfaces and a measurement scale [348], but the spatial scale used in the CDSM model is a combustion filter size associated with the thickness of the LES flame. Although these two scales play similar roles, seemingly no study has verified whether the power-law exponent derived from the evolution of the surface of a DNS flame front filtered at increasing filter scales is the same as the fractal dimension of a caliper-stepping or box-counting method applied to the unfiltered flame front. If these two quantities were to be equal, this should offer an insightful justification to the power-law relationship of the CDSM model with respect to the combustion filter size.

The CDSM model leans towards the high bias/low variance side of the bias-variance tradeoff at the center of conventional ML principles (Section 3.1). On the other hand, the CNN is rather a low bias/high variance model, leveraging vast amounts of data without any

specific physical hypothesis. Without a prior regarding the existence of a specific relationship between the filtered and test-filtered FSDs, it is unclear whether the CNN should be able extract any useful information from the $\hat{\tilde{c}}$ field which does not contain any new information by virtue of the data processing inequality.

Nevertheless, a final version of the CNN is trained with the test-filtered progress variable $\hat{\tilde{c}}$ as a second input channel. The test-filter size is chosen as $\hat{\Delta} = 1.8\Delta$. Results in Appendix C still show good performance on the validation dataset. However, this version does not lead to an improvement in the behavior of the model. Wrinkling in the laminar phase reaches the same levels as the original AVBP-DL simulation, and the peak overpressure is slightly lower (Figure 8.52), underlining the sensitivity of this metric to the specific instance of the CNN model.

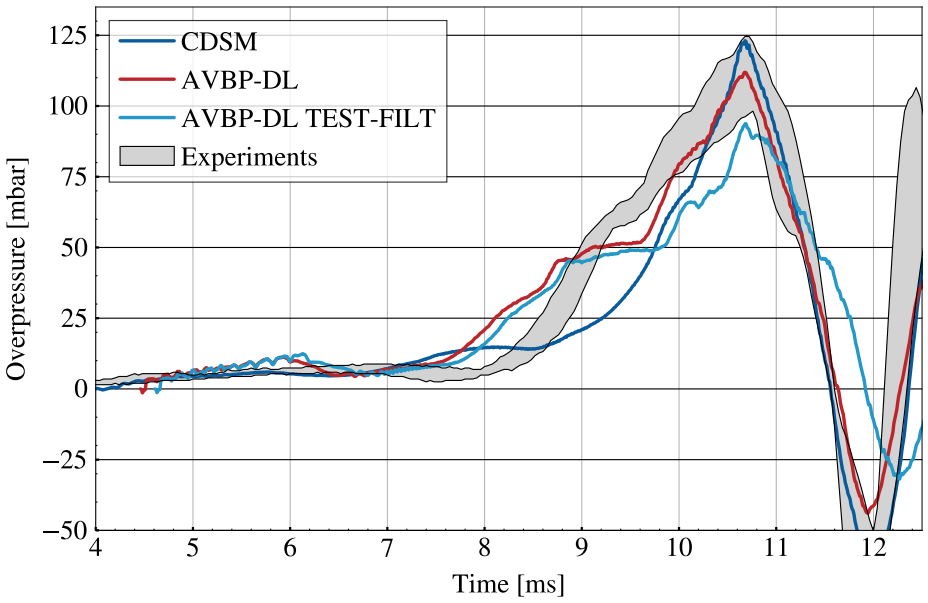


Figure 8.52: Overpressure vs. time for the AVBP-DL TEST-FILT simulation. LES times are shifted to match time of peak overpressure in the experiment.

8.10 Conclusion and perspectives

CNN models for the SGS wrinkling factor were trained on the HIT dataset and evaluated *a posteriori* on the small-scale Masri explosion configuration. This case is known to be sensitive to the choice of the turbulent combustion model. The AVBP-DL coupling strategy was used with Table 8.7 sums up the main results of the AVBP-DL simulations. The original CNN reached a peak overpressure inside the experimental range, but displayed increased flame acceleration in the early stages of flame propagation compared to the reference CDSM simulation. This is induced by high wrinkling values predicted by the CNN in the initial laminar phase. Differences between the CNN and CDSM models were underlined. The CNN model tends to predict more low values of wrinkling at the forward end of the propagating flame front, which may balance the greater initial flame acceleration in the resulting peak

overpressure. A moderate correlation of the CNN predictions with the flame front curvature could explain the greater fragmentation of the AVBP-DL turbulent flame front.

Attempts to reconcile the accurate prediction of the peak overpressure and unity wrinkling values in the laminar phase were proposed. Transitioning from the CDSM model to the CNN after the flame has crossed the first flame front or adding laminar spherical flames in the training dataset both lead to a lower peak overpressure than the original simulation. This suggests that the wrinkling levels predicted by the CNN in the turbulent phase are not large enough. This issue is not solved by adding the test-filtered progress variable as input, and could motivate to reconsider the choice of the training configuration. Including turbulence information via the resolved vorticity field led to an unstable simulation to a distribution shift in the values of the vorticity given to the model as input.

Clearly, some work still remains before DL SGS models are widely used in industrial-scale LES computations. Beyond addressing the shortcomings of the CNN seen on the small-scale configuration, applications to the medium-scale or large-scale cases, where the chamber length grows by a factor 6 and then 4, should be pursued in future works. Currently, even the CDSM model struggles to recover the correct overpressure at these scales, so an adequately trained DL model has the potential to unlock unprecedented use cases for industrial safety. Increasing the scale of the configuration at a fixed computational cost would however bring about new challenges for the CNN training methodology. The combustion filter size would increase, and the strict application of Algorithm 2 would require the resolution of the training set DNS to grow by the same scale factor. For the large-scale configuration, this would require a DNS with unattainable computational demands. New methodologies could be developed to overcome this difficulty.

Name	Correct peak overpressure	Correct laminar behavior
AVBP-DL	✓	✗
AVBP-DL FGI	✗	–
AVBP-DL LAM	✗	✓
AVBP-DL VORT	✗	✗
AVBP-DL TEST-FILT	✗	✗

Table 8.7: Summary of the AVBP-DL simulation results.

Chapter 9

Conclusion and perspectives

Contents

9.1 Conclusion on the contributions of this thesis

This thesis has investigated developments for the applicability of deep learning SGS models to large-scale LES of turbulent premixed flames. There is now a sizable body of literature on SGS modeling with deep neural networks, yet little to no *a posteriori* validation of these models on practical configurations. To achieve this objective, three elements were identified as insufficiently studied: evaluation of deep learning models on challenging high Reynolds test cases, assessment of their ability to generalize by selecting different training and test configurations, and the development of efficient coupling strategies with high-performance LES solvers. These elements were incrementally put in practice in the three DL models based on U-Net convolutional neural networks.

First, an existing model for the total flame surface density was trained and evaluated *a priori* on the R2 high Reynolds turbulent premixed jet flame (Chapter 5). It showed an improvement on the state-of-the-art CDSM model, including when tested on LES snapshots. These results were robust to variations in the nature of the filter and the progress variable formulation, and insights into the effective receptive field and the evolution of parameter distributions during training were provided.

Then, a model was trained to predict the SGS progress variable variance on the HIT statistically planar flame DNS (Chapter 6). Based on this model, the PB-CNN analytical closure for the filtered reaction rate was proposed and tested *a priori* on the R2 flame, which exhibits significant differences with the training configuration. Yet, with a proper choice of the filtering parameters for the training dataset, excellent generalization performance was observed for the SGS progress variable variance and the filtered reaction rate.

A posteriori evaluation of DL SGS models were made possible by the AVBP-DL coupling strategy which seamlessly integrates DL models in the temporal loop of the AVBP solver for virtually no computational overhead (Chapter 7). The final application was the LES of a gaseous explosion in the vented obstructed chamber of the Masri configuration (Chapter 8). A DL model trained on the HIT dataset to predict the SGS wrinkling factor led to the

correct peak overpressure. However, this result was shown to result from the compensation of increased flame acceleration in the early stages of flame propagation, especially in the laminar phase where super-unity wrinkling values are predicted by the model, and of insufficient wrinkling in the most critical phase of turbulent propagation. Many variations were tested to attempt to reconcile the expected physical behavior of the model and the correct peak overpressure, but none delivered fully convicing results.

All in all, progress was made in the right direction, culminating in the Masri AVBP-DL simulation which is, to our knowledge, the first *a posteriori* test of a DL SGS model in a non-academic configuration with a massively-parallel LES solver. Yet the conclusions of this simulation prove that some work still remains before deep learning reliably replaces SGS models in numerical combustion.

9.2 Perspectives for future research

Ensuring consistent laminar/turbulent behaviors Chapter 8 suggests that a pure data-driven learning approach on the HIT dataset may not be sufficient to capture both the laminar and turbulent flame propagation regimes in the Masri simulation. Future work could investigate modifications to the model inputs, architecture, or training procedure to pursue consistent laminar/turbulent behaviors in a generalizable way. For example, turbulence sensors used in mesh adaptation could be used as input features to the model.

Turbulent combustion regime adaptation More generally, ensuring that the model can generalize to a wide set of turbulent conditions would increase the range of applications that could be covered with deep learning. In this thesis, dependency on the combustion regime was softly enforced by conserving u'/s_L and Δ/δ_L in Chapter 6. Failure to do so resulted in degraded results, and may also explain the shortcomings of the wrinkling model on the Masri simulation. A simple solution would be to use quantities that quantify the turbulent combustion regime such as u'/s_L as input features to the model, or as side information via feature-wise transformations [79]. At first glance, this runs the risk of failing to generalize outside of the range of values seen in training, but it could prove to be sufficient in target specific applications where the combustion regime is known *a priori*. Such a solution could also require an extensive training database that efficiently samples a subset of the turbulent combustion diagram. New dataset sharing initiatives [62] may accelerate this process in the near future.

Filter size dependency To increase the flexibility of DL SGS models, their dependency on the filter size must be addressed, as it controls the magnitude of the SGS quantity to be modeled. Empirical rules (Propositions 4.1, 4.2) were proposed in this thesis to adapt the training set filter size to a known evaluation configuration. This dependency is addressed in some works by including multiple filter sizes in the training dataset [158] or using Δ as an input feature [152, 158, 335, 337]. In the future, more robust, generalizable methods should be sought to incorporate this information in the model.

Mesh dependency A related topic is the dependency of DL models on the data discretization. The models trained in this thesis were CNNs that operate on voxel grids for 3D spatial data. CNNs rely on a consistent grid size for generalization, as it determines the magnitude of gradients in the data. In this thesis, only uniform voxel meshes were used, and Proposition 4.3 proposed a simple criterion to determine the discretization size of the training set depending on the evaluation case. Extending this method to non-uniform voxel meshes could be implemented by adding the cell size to the input features. The issues related to generalization that were mentioned for the filter size and the turbulent combustion regime would then also apply here.

For irregular geometries and complex meshes, working directly on the unstructured grid of the LES solver is a promising solution. This would simplify the coupling scheme as no back-and-forth interpolation would be required, although this step was not found to induce any sensible overhead in Chapter 7. As discussed in the conclusion of Chapter 7, GNNs like MeshGraphNets [268] are well-adapted to this form of discretization as their node and edge embeddings contain absolute distance values, allowing them to capture the spatial structure of the mesh. Point cloud representations could also be investigated, as it is unclear whether the connectivity of the unstructured mesh must be strictly respected by the network [157, 288].

In fact, the grid discretization defines two separate quantities: the resolution of the spatial structures and their scale (absolute size). Both can vary independently from each other, and ideal models should be expected to respond correctly to changes in resolution (where the underlying physics does not change) and in scale (where the physics changes but not the discretized fields seen by the model). To circumvent this difficult problem, methods that learn continuous mappings such as Fourier neural operators [193] could be explored. To date however, GNNs and neural operators still struggle to operate on large 3D meshes.

Encoding physical invariance Leveraging physical invariance has proven to effectively help generalization in some CFD applications of deep learning [196, 284, 388]. Many fluid quantities are invariant or equivariant to operations such as translation, rotation, scaling, or Galilean transformations. Invariance is one of the most promising research topics in the general deep learning community [65, 66, 359, 391, 396]. The intersection between model invariance and numerical combustion modeling warrants some attention to build innovative models that can generalize better. This thesis has only scratched the surface of this topic through the promotion of CNNs that are naturally translation equivariant, and systematic data augmentation with 90° rotations and reflections.

Partially-premixed and non-premixed combustion This thesis has focused on modeling approaches for fully premixed combustion. In practical combustion systems, flames usually burn in a partially-premixed regime, where reactants are not fully premixed and mixture fractions fluctuate between zero to unity [76, 328]. In this regime, modeling may combine models derived for fully premixed and non-premixed combustion, or require a new separate treatment. In the first case, a simple assignment of DL SGS models to purely premixed and non-premixed regions could be performed with a flame index [401] or classification by a machine learning model [63, 383].

Decarbonized fuels Future combustion systems will increasingly involve decarbonized fuels such as hydrogen and ammonia. In lean premixed mixtures, these fuels are particularly prone to enhanced combustion triggered by thermodiffusive instabilities. The interplay of differential diffusion and stretch effects leads to increased burning rates compared to thermodiffusively-stable flames [28] which is not directly correlated to increased wrinkling [27]. Flame surface-based models like the ones developed in Chapters 5 and 8 can therefore not account for the turbulent reaction rate of these mixtures. Note that this also applies to equidiffusive mixtures subject to extreme turbulence that causes reaction layer thickening [15]. DL SGS models applied to lean hydrogen or ammonia combustion should perhaps focus directly on the prediction of the filtered reaction rate, trading off genericity for the accuracy of a specialized model.

Beyond supervised training The conventional workflow to train an SGS model, whether physically- or data-driven, relies on the generation of high-fidelity DNS data that is filtered to generate ground truth labels. This is a supervised training framework that constrains training datasets to DNS simulations that are readily available. Reliable DNS simulations are a rare commodity since they involve great computational expenses, and often focus on academic test cases that are not representative of practical combustion configurations. Finding a solution for semi-supervised, self-supervised, or unsupervised training of SGS models would make great inroads in deep learning applications for LES. Inspiration can be taken from the vast quantity of deep learning research on unsupervised training for style transfer [201, 202, 394] which was already applied to turbulence super-resolution [162], or contrastive and non-contrastive self-supervised learning [41, 53, 120, 136, 407].

9.3 List of publications

This thesis has led to a first author journal publication

- Xing, V.; Lapeyre, C.; Jaravel, T.; Poinsot, T. Generalization Capability of Convolutional Neural Networks for Progress Variable Variance and Reaction Rate Subgrid-Scale Modeling. Energies 2021, 14, 5096. https://doi.org/10.3390/en14165096,

a presentation as a main speaker in an international conference

- Lapeyre, C.; Misdariis, A.: Cazard, N.; Xing, V.; Veynante, D.; Poinsot, T. (2019, May 6-8). A convolutional neural network-based efficiency function for sub-grid flame-turbulence interaction in LES. 17th International Conference on Numerical Combustion, Aachen, Germany,

and a related presentation in an international conference

- Serhani, A.; Xing, V.; Dupuy, D.; Lapeyre, C.; Staffelbach, G. (2022, May 25-27). High-performance hybrid coupling of a CFD solver to deep neural networks. 33rd Parallel CFD International Conference, Alba, Italy.

Other journal publications are in preparation, including:

- Xing, V.; Lapeyre, C. (2023). Deep Convolutional Neural Networks for Subgrid-Scale Flame Wrinkling Modeling. In N. Swaminathan and A. Parente (eds.), Machine Learning and Its Application to Reacting Flows, Lecture Notes in Energy 44, https://doi.org/10.1007/978-3-031-16248-0_6
- Shin, J.; Xing, V.; Pfitzner, M.; Lapeyre, C. Probabilistic deep learning of turbulent premixed combustion. Submitted to Applications in Energy and Combustion Science.
- Coulon, V.; Gaucherand, J.; Xing, V.; Laera, D.; Lapeyre, C.; Poinsot, T. Comparison of methane, ammonia/hydrogen and hydrogen turbulent premixed flames. In preparation.
- Serhani, A.; Xing, V.; Dupuy, D.; Lapeyre, C.; Staffelbach, G. High-performance hybrid solver for computational fluid dynamics coupled to data-driven models: AVBP-DL. In preparation.

Appendices

Appendix A

Kolmogorov-Petrovsky–Piskunov analysis of the Pfitzner beta PDF source term

This appendix uses the Kolmogorov-Petrovsky–Piskunov theorem [173] to derive a closed-form formula for the turbulent flame speed of a 1D premixed steady flame computed with the beta PDF source term of Equation 6.5. The KPP theorem is a classical tool in the study of propagating waves in nonlinear partial differential equations [112, 323, 353] and has been used in the past to derive analytical turbulent flame speeds based on specific reaction source terms formulations [97, 171, 198, 409].

A.1 KPP analysis for an arbitrary source term

Consider a 1D premixed steady flame with a single-step irreversible chemical mechanism. The Lewis number is set to unity. Turbulence is assumed to be incorporated in a total diffusivity D containing both laminar and turbulent diffusivities. Conservation of mass implies that $\rho u = \rho_u s_T$ where s_T is the flame leading edge velocity which is equated to the turbulent flame speed. The balance equation for the progress variable $c = 1 - Y_F/Y_F^u$ writes:

$$\rho_u s_T \frac{\partial c}{\partial x} = \frac{\partial}{\partial x}\left(\rho D \frac{\partial c}{\partial x}\right) - \frac{\dot{\omega}_F}{Y_F^u}\,. \tag{A.1}$$

Like in Hakberg and Gosman [129], assume that s_T is governed by the behavior at cold boundary, where $\rho D \approx \rho_u D_u$. Equation A.1 is then a second-order autonomous differential equation of the form

$$c'' - \alpha c' + f(c) = 0 \tag{A.2}$$

with:

$$\alpha = s_T/D_u\,, \tag{A.3}$$

$$f(c) = -\dot{\omega}_F(c)/(\rho_u D_u Y_F^u)\,. \tag{A.4}$$

The usual method to study this type of equation is to introduce a phase space (c, p) with $p = c'$:

$$\begin{cases} c' = p \\ p' = \alpha p - f(c) \end{cases} \tag{A.5}$$

The equation has two singular points $(0,0)$ and $(1,0)$. Linearizing around $(0,0)$ gives:

$$\begin{pmatrix} c' \\ p' \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -f'(0) & \alpha \end{pmatrix} \begin{pmatrix} c \\ p \end{pmatrix} \tag{A.6}$$

The nature of the singular point is determined by the characteristic polynomial $\lambda^2 - \alpha\lambda + f'(0)$. Its roots are $\lambda_{1,2} = \frac{c \pm \sqrt{\alpha^2 - 4f'(0)}}{2}$. If $\alpha^2 - 4f'(0) < 0$, the singular point $(0,0)$ is an attractor, and phase trajectories approach it in a converging spiral. This is impossible because c must remain positive, hence:

$$\alpha^2 \geq 4f'(0)\,. \tag{A.7}$$

Note that this is the same equation as Equation (5.132) of Poinsot and Veynante [275], where the KPP analysis is presented for an eddy break-up source term which is quadratic in c.

The KPP theorem [323, 353] states that the propagation speed of the traveling flame front is found when Equation A.7 is an equality:

$$\alpha^2 = 4f'(0)\,. \tag{A.8}$$

Using Equations A.3 and A.4, the final expression for the turbulent flame speed is then given by:

$$\boxed{s_T = 2\sqrt{\frac{-\dot{\omega}'_F(0) D_u}{\rho_u Y_F^u}}\,.} \tag{A.9}$$

Two hypotheses are necessary for the KPP theorem to rigorously apply:

- $f'(0) > 0$
- $f'(c) \leq f'(0),\ c \in [0,1]$

Since $f = \dot{\omega}_F / (\rho_u D_u Y_F^u)$, these two inequalities must also apply to the source term.

A.2 Application to the Pfitzner source term

The non-dimensional Pfitzner turbulent source term for a beta PDF $p_\beta(a;b)$ writes:

$$\omega_\beta = (m+1)\frac{\Gamma(a+b)}{\Gamma(a)}\left(\frac{\Gamma(a+m+1)}{\Gamma(a+b+m+1)} - \frac{\Gamma(a+2m+1)}{\Gamma(a+b+2m+1)}\right)\,, \tag{A.10}$$

with

$$a = \bar{c}\left[\frac{\bar{c}(1-\bar{c})}{\overline{c'^2}} - 1\right]\,, \tag{A.11}$$

$$b = (1-\bar{c})\left[\frac{\bar{c}(1-\bar{c})}{\overline{c'^2}} - 1\right]\,. \tag{A.12}$$

For a well-chosen expression of the SGS progress variable variance $\overline{c'^2} = k\bar{c}(1-\bar{c})$ with k a constant, a closed-form formula for the source term derivative at $c = 0$ can be found.

In the following, $k = 0.1$. Introducing $\alpha = 1/k - 1 = 9$, a and b have simple expressions (overbars denoting filtering are omitted for clarity):

$$a = c\left[\frac{c(1-c)}{kc(1-c)} - 1\right] = \alpha c\,, \tag{A.13}$$

$$b = (1-c)\left[\frac{c(1-c)}{kc(1-c)} - 1\right] = \alpha(1-c)\,. \tag{A.14}$$

For $m = 1$, the non-dimensional reaction rate can be simplified using the properties of the Γ function:

$$\omega_\beta = (m+1)\frac{\Gamma(\alpha)}{\Gamma(\alpha c)}\left(\frac{\Gamma(\alpha c + m + 1)}{\Gamma(\alpha + m + 1)} - \frac{\Gamma(\alpha c + 2m + 1)}{\Gamma(\alpha + 2m + 1)}\right) \tag{A.15}$$

$$= 2\Gamma(\alpha)\left(\frac{[(\alpha c+1)\alpha c]\cancel{\Gamma(\alpha c)}}{\cancel{\Gamma(\alpha c)}\Gamma(\alpha+2)} - \frac{[(\alpha c+2)(\alpha c+1)\alpha c]\cancel{\Gamma(\alpha c)}}{\cancel{\Gamma(\alpha c)}\Gamma(\alpha+3)}\right) \tag{A.16}$$

$$= 2\Gamma(\alpha)\left(\frac{\alpha^2c^2 + \alpha c}{\Gamma(\alpha+2)} - \frac{\alpha^3c^3 + 3\alpha^2c^2 + 2\alpha c}{\Gamma(\alpha+3)}\right)\,. \tag{A.17}$$

Its derivative at $c = 0$ is:

$$\omega'_\beta(0) = 2\Gamma(\alpha)\alpha\left(\frac{1}{\Gamma(\alpha+2)} - \frac{2}{\Gamma(\alpha+3)}\right) \tag{A.18}$$

$$= 2 \times 9!\left(\frac{1}{10!} - \frac{2}{11!}\right) \tag{A.19}$$

$$= \frac{9}{55} \tag{A.20}$$

$$\approx 0.164\,. \tag{A.21}$$

Going back to the dimensional reaction rate:

$$\dot{\omega}'_F(0) = -\frac{Y_F^u(\rho_u s_L)^2}{\rho_u D_u^{lam}}\omega'_\beta(0)\,. \tag{A.22}$$

The KPP Equation A.9 finally writes:

$$\boxed{s_T = 2s_L\sqrt{\frac{D_u}{D_u^{lam}}\omega'_\beta(0)}\,.} \tag{A.23}$$

This formula was validated by implementing the Pfitzner beta PDF reaction rate in AVBP and computing the turbulent flame speed with Equation 2.30. For the parameters $s_L = 0.383\,\mathrm{m\,s^{-1}}$, $\mu_t = 5 \times 10^{-4}\,\mathrm{N\,s\,m^{-2}}$, $\mu_{lam} = 1.8 \times 10^{-5}\,\mathrm{N\,s\,m^{-2}}$, Equation A.23 and AVBP give the same turbulent flame speed $s_T = 1.664\,\mathrm{m\,s^{-1}}$.

Some comments on this derivation:

- If k is increased to 1/3, Equation A.18 correctly accounts for the higher SGS variance by increasing $\omega'_\beta(0)$ to 1/3.

- Fichot et al. [97] performed a KPP analysis on a FSD-based source term and find a similar expression $s_T \propto \sqrt{k}$ with k the turbulent kinetic energy.

A.3 Some general formulas

A general formula can be derived for any integer value of m:

$$\omega'_\beta(0) = (m+1)\Gamma(\alpha)\alpha\left(\frac{m!}{\Gamma(\alpha+m+1)} - \frac{(2m)!}{\Gamma(\alpha+2m+1)}\right). \tag{A.24}$$

For instance with $m = 2$, $\omega'_\beta(0) = 0.0503$. If m is a half-integer, closed-form formulas can also be found.

A.3.1 $m = 1/2$

$$\omega_\beta = \frac{3\Gamma(\alpha)}{2\Gamma(\alpha c)}\left(\frac{\Gamma(\alpha c+3/2)}{\Gamma(\alpha+3/2)} - \frac{\Gamma(\alpha c+2)}{\Gamma(\alpha+2)}\right) = A(c) - B(c)\,. \tag{A.25}$$

First compute the derivative of $A(c)$:

$$A'(c) = \frac{3\Gamma(\alpha)}{2\Gamma(\alpha+3/2)}\left(\frac{\alpha\Gamma'(\alpha c+3/2)\Gamma(\alpha c) - \alpha\Gamma'(\alpha c)\Gamma(\alpha c+3/2)}{\Gamma(\alpha c)^2}\right) \tag{A.26}$$

$$= \frac{3\alpha\Gamma(\alpha)}{2\Gamma(\alpha+3/2)}\left(\frac{\psi^{(0)}(\alpha c+3/2)\Gamma(\alpha c+3/2)\Gamma(\alpha c) - \psi^{(0)}(\alpha c)\Gamma(\alpha c)\Gamma(\alpha c+3/2)}{\Gamma(\alpha c)^2}\right) \tag{A.27}$$

$$= \frac{3\alpha\Gamma(\alpha)\Gamma(\alpha c+3/2)}{2\Gamma(\alpha+3/2)}\left(\frac{\psi^{(0)}(\alpha c+3/2)}{\Gamma(\alpha c)} - \frac{\psi^{(0)}(\alpha c)}{\Gamma(\alpha c)}\right). \tag{A.28}$$

Since $\Gamma \xrightarrow{0+} +\infty$ and $\psi^{(0)}/\Gamma \xrightarrow{0} -1$, this leads to:

$$A'(0) = \frac{3\alpha\Gamma(\alpha)\Gamma(3/2)}{2\Gamma(\alpha+3/2)}\,. \tag{A.29}$$

For the second term $B(c)$:

$$B'(0) = \frac{3\alpha\Gamma(\alpha)}{2\Gamma(\alpha+2)}\,. \tag{A.30}$$

Finally:

$$\omega'_\beta(0) = \frac{3\alpha\Gamma(\alpha)}{2}\left(\frac{\Gamma(3/2)}{\Gamma(\alpha+3/2)} - \frac{1}{\Gamma(\alpha+2)}\right) \approx 0.28\,. \tag{A.31}$$

A.3.2 $m = 3/2$

$$\omega'_\beta(0) = \frac{5\alpha\Gamma(\alpha)}{2}\left(\frac{\Gamma(5/2)}{\Gamma(\alpha+5/2)} - \frac{6}{\Gamma(\alpha+4)}\right) \approx 0.090 \tag{A.32}$$

A.4 Turbulent flame thickness

Equation A.2 can be linearized around $c = 0$:

$$c'' - mc' + f'(0)c = 0\,, \tag{A.33}$$

which means $c(x)$ is an exponential form. The KPP theorem states that the discriminant of the characteristic equation is zero, leading to:

$$c(x) = e^{mx/2}(a_1 + a_2 x)\,. \tag{A.34}$$

However this formula does not easily lead to an expression for the turbulent flame thickness, because the KPP analysis is valid at the leading edge of the flame, and Equation A.34 is only valid there (obviously c does not go to infinity). Equation A.34 suggests the formula:

$$\delta_L^t = \frac{2}{m} = \frac{2D_u}{s_T}\,. \tag{A.35}$$

AVBP simulations indicate that this formula is off by a factor 4 exactly: $\delta_L^t = (8D_u)/s_T$. Future work could be dedicated to looking for an exact formula for δ_L^t which is consistent with simulation results.

Appendix B

AVBP-DL performance metrics

This appendix contains AVBP-DL performance metrics that supplement Section 7.3.

B.1 DL overhead components

Figure B.1 shows the time per operation of the AVBP send operation. Figure B.2 shows the time per iteration of the AVBP process Ξ operation. Both of these timers scale correctly with the number of nodes for distributions D4 and D0.5.

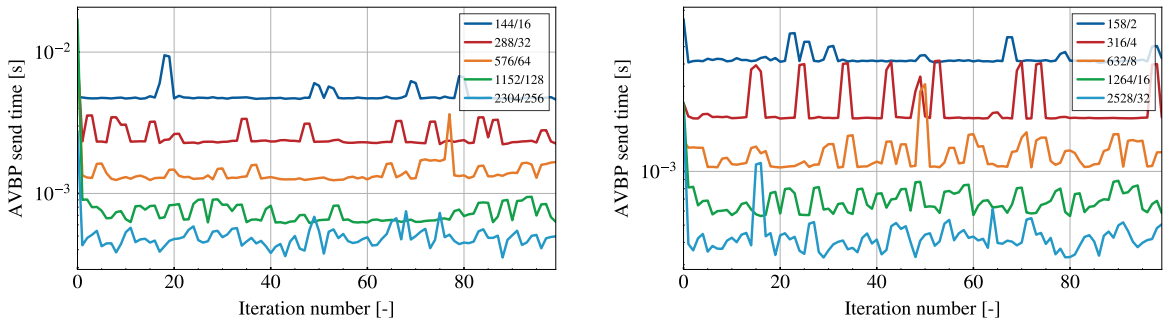


Figure B.1: AVBP send time per iteration for D4 (left) and D0.5 (right) (average over five executions of the maximum among all DL processes). Each configuration is labeled by its AVBP/DL process distribution.

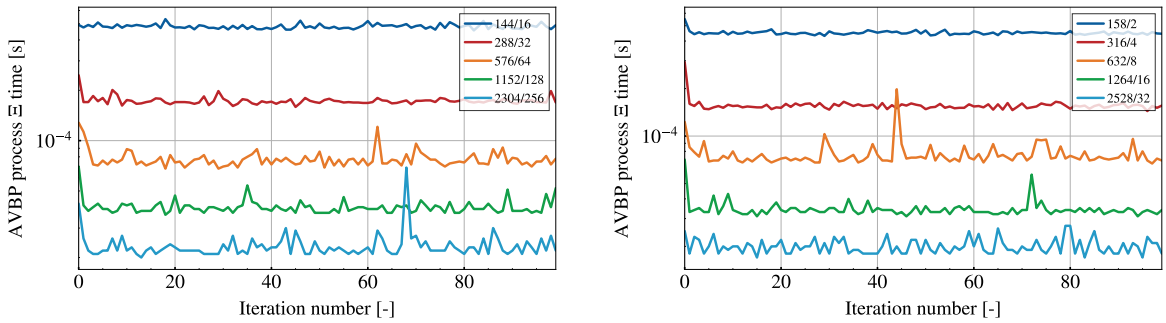


Figure B.2: AVBP process Ξ time per iteration for D4 (left) and D0.5 (right) (average over five executions of the maximum among all DL processes). Each configuration is labeled by its AVBP/DL process distribution.

B.2 DL time breakdown

Figures B.3 and B.4 report the breakdown of the DL iteration time averaged over all DL processes and all iterations for distributions D4 and D0.5. The AVBP-DL temporal iteration timeline of Figure 7.3 outlines 4 steps that compose the DL iteration between the exchanges of the c and Ξ fields. The fraction of each of these steps in the DL iteration time are shown in pie charts.

For D4, as the number of node increases, the proportion of the CNN prediction in the DL iteration grows. This is caused by the bad scalability of the prediction time due to the small dimensionality of the CNN inputs. With the lowest number of nodes, the duration of the prediction is already very low, around 20 ms. Improvements may come from Tensorflow installation optimizations but were not deemed necessary given the low prediction times. By contrast, the steps involving processing and exchanging Ξ take up smaller fractions of the DL iteration, and the share of the processing of c is stable.

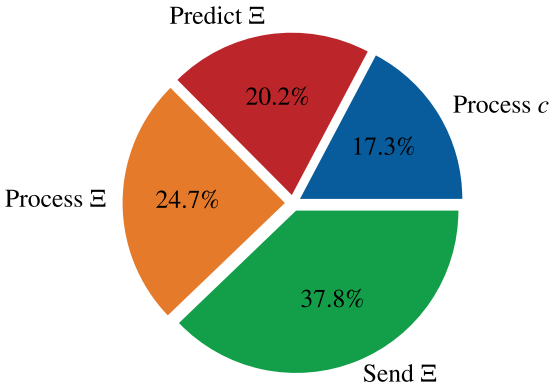
For D0.5, the largest component of the DL iteration time is the exchange of Ξ, except for the largest number of nodes where it is overtaken by the prediction. The exchange is dominated by the interpolation of Ξ from the DL mesh to the AVBP mesh. Compared to D4, there are less DL processes per AVBP process in D0.5, so each DL process must handle interpolation with a greater number of AVBP processes. This could explain the greater importance of this step in D0.5. In spite of the differences in the initial breakdown, each timer follows the same trend as in D4. The fraction of prediction time increases sixfold from 4 to 64 nodes, and the fraction of Ξ processing shrinks by a factor two.

B.3 DL overhead breakdown

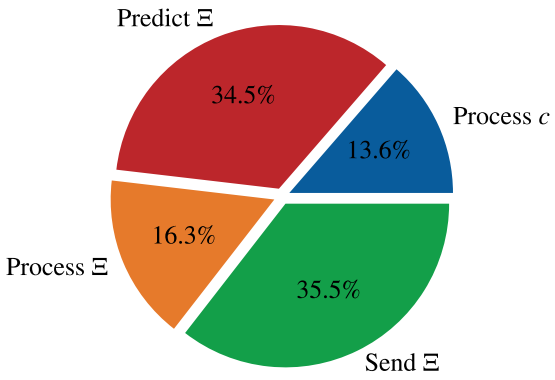
Figures B.5 and B.6 report the breakdown of the DL overhead averaged over all AVBP processes and all iterations for distributions D4 and D0.5.

For D4, as the number of nodes increases, the DL overhead is dominated by the send operation (involving interpolation to the DL mesh), then by the wait receive operation indicating unfinished computations by the DL solver, and finally by the wait send operation due to communication bottlenecks.

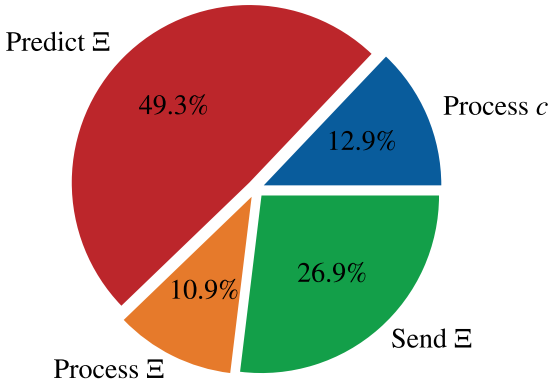
For D0.5, the reduced number of DL processes solves the issue of communication bottlenecks but increases the load placed on the DL solver, and the DL overhead is thus almost entirely due to the wait receive operation.



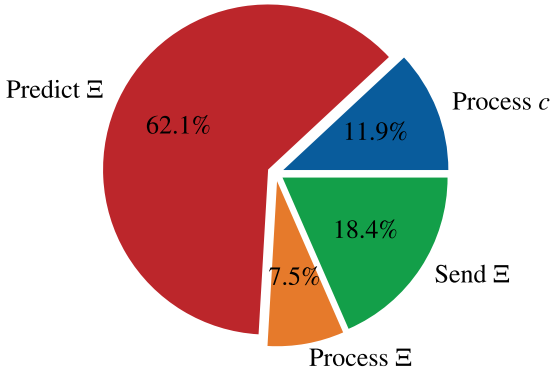
(**a**) 4 nodes (144 AVBP/16 DL processes)



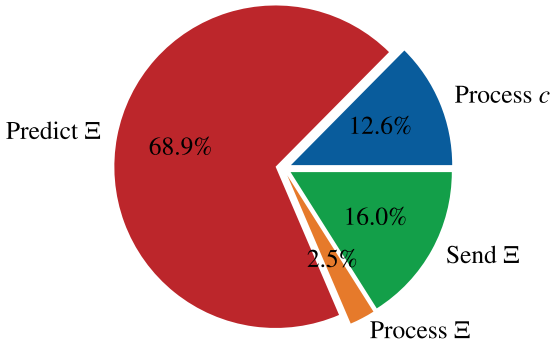
(**b**) 8 nodes (288 AVBP/32 DL processes)



(**c**) 16 nodes (576 AVBP/64 DL processes)

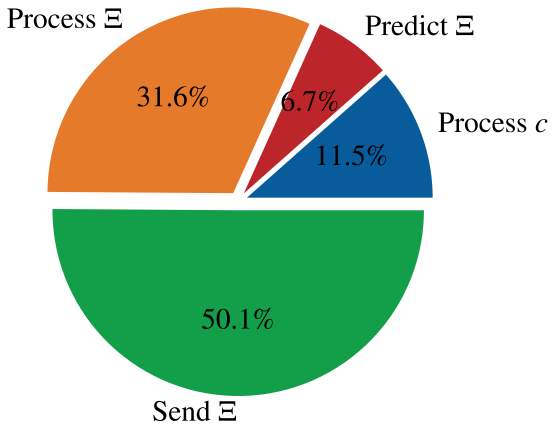


(**d**) 32 nodes (1152 AVBP/128 DL processes)

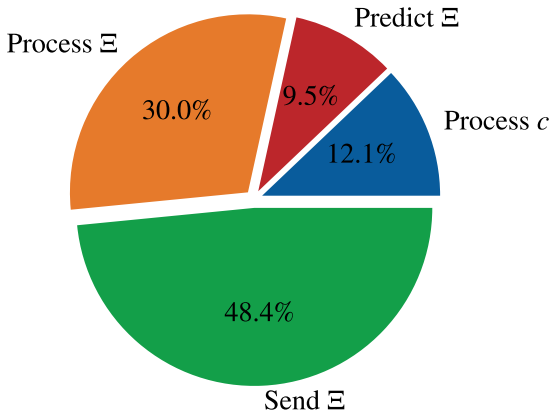


(**e**) 64 nodes (2304 AVBP/256 DL processes)

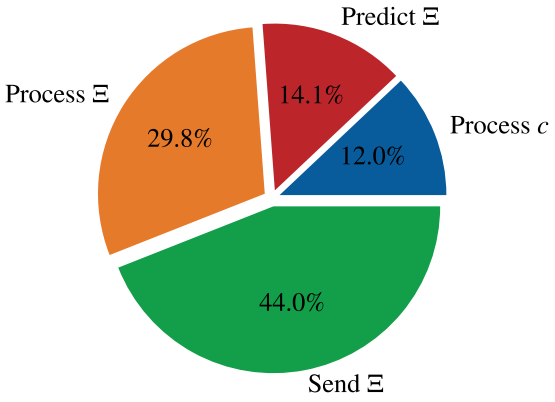
Figure B.3: Breakdown of the mean DL time per iteration for D4 with 4 (**a**), 8 (**b**), 16 (**c**), 32 (**d**), and 64 (**e**) nodes.



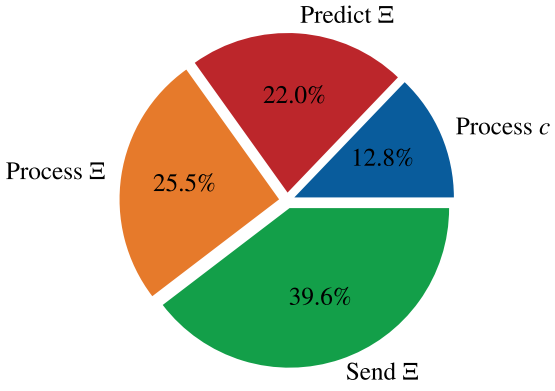
(**a**) 4 nodes (158 AVBP/2 DL processes)



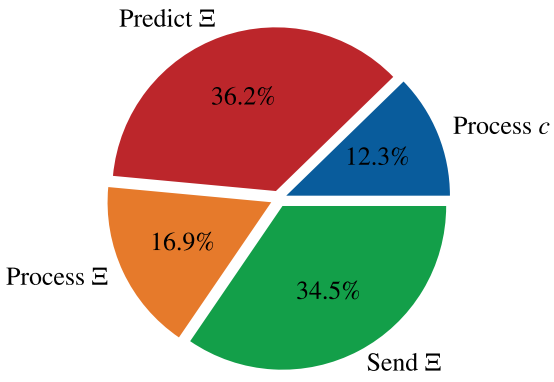
(**b**) 8 nodes (316 AVBP/4 DL processes)



(**c**) 16 nodes (632 AVBP/8 DL processes)

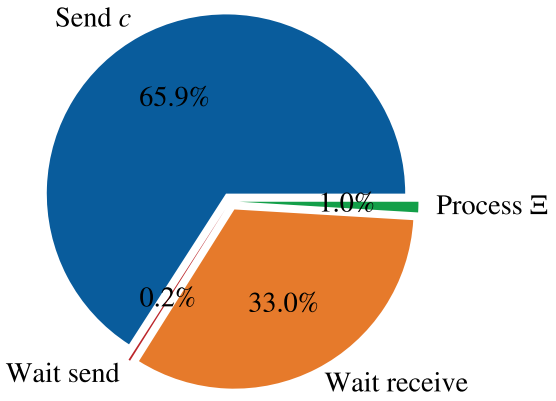


(**d**) 32 nodes (1264 AVBP/16 DL processes)

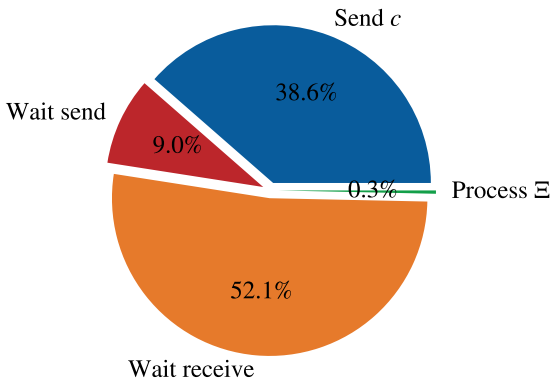


(**e**) 64 nodes (2528 AVBP/32 DL processes)

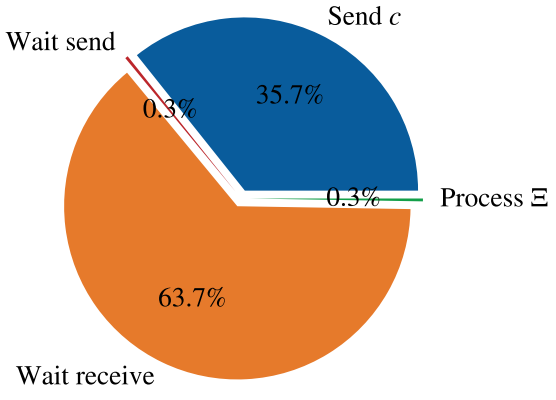
Figure B.4: Breakdown of the mean DL time per iteration for D0.5 with 4 (**a**), 8 (**b**), 16 (**c**), 32 (**d**), and 64 (**e**) nodes.



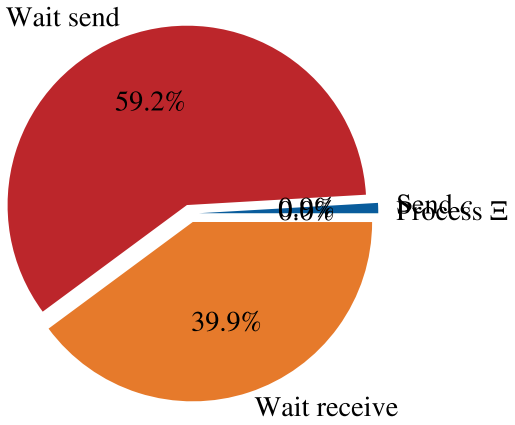
(**a**) 4 nodes (144 AVBP/16 DL processes)



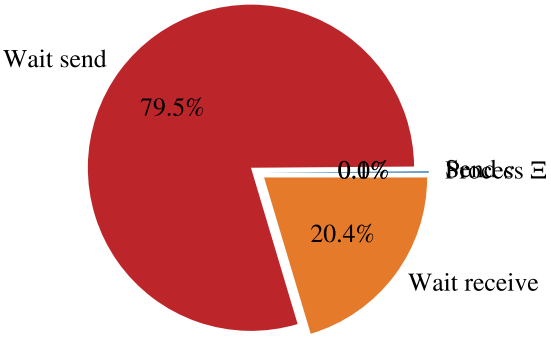
(**b**) 8 nodes (288 AVBP/32 DL processes)



(**c**) 16 nodes (576 AVBP/64 DL processes)

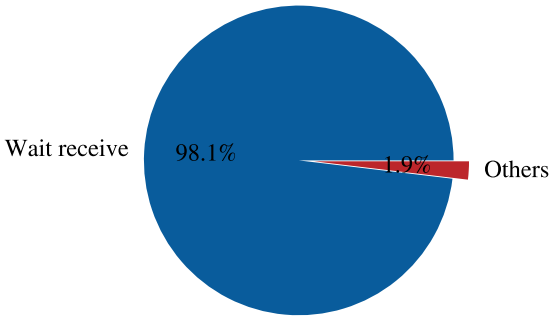


(**d**) 32 nodes (1152 AVBP/128 DL processes)

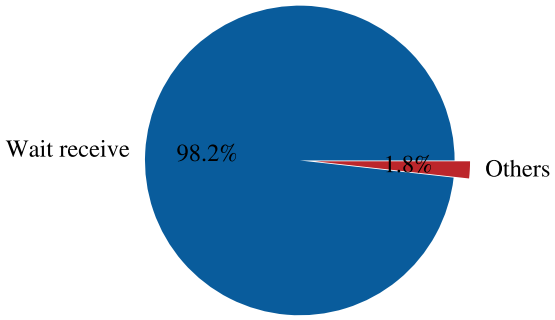


(**e**) 64 nodes (2304 AVBP/256 DL processes)

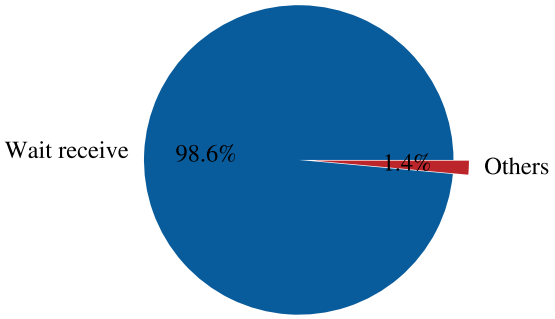
Figure B.5: Breakdown of the mean DL overhead per iteration for D4 with 4 (**a**), 8 (**b**), 16 (**c**), 32 (**d**), and 64 (**e**) nodes.



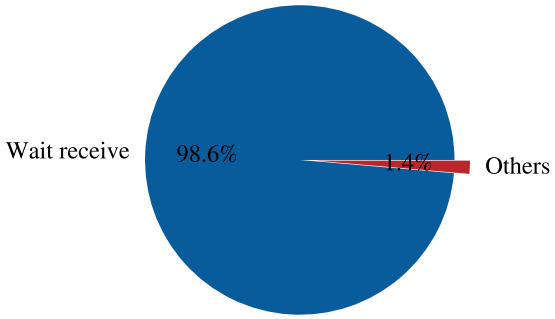
(**a**) 4 nodes (158 AVBP/2 DL processes)



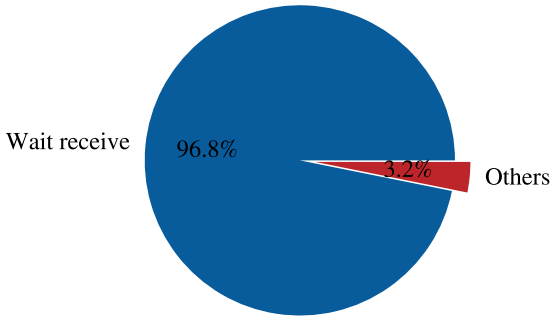
(**b**) 8 nodes (316 AVBP/4 DL processes)



(**c**) 16 nodes (632 AVBP/8 DL processes)



(**d**) 32 nodes (1264 AVBP/16 DL processes)



(**e**) 64 nodes (2528 AVBP/32 DL processes)

Figure B.6: Breakdown of the mean DL overhead per iteration for D0.5 with 4 (**a**), 8 (**b**), 16 (**c**), 32 (**d**), and 64 (**e**) nodes.

Appendix C

Performance metrics of wrinkling CNNs on the HIT dataset

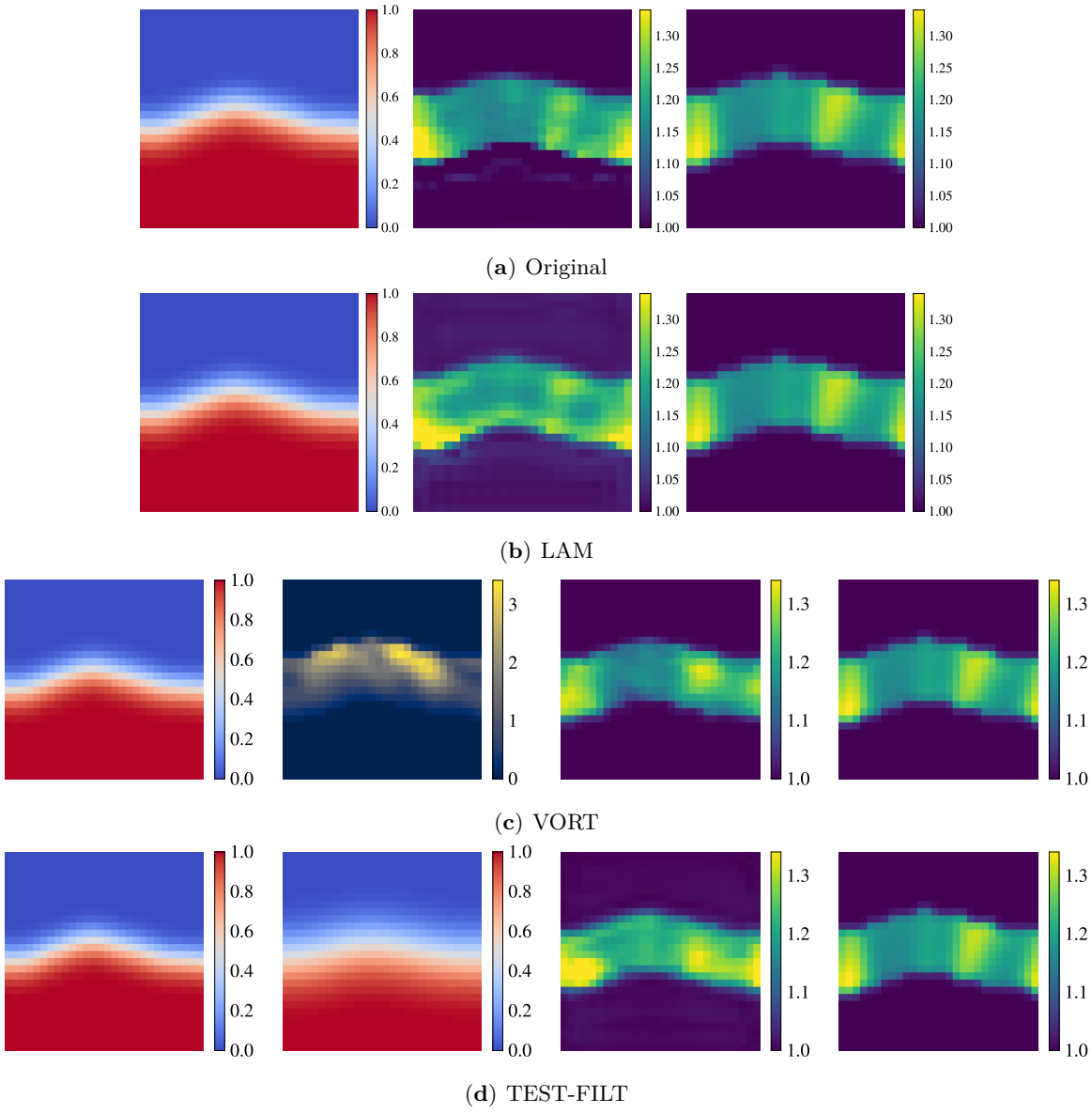
This appendix gathers performance metrics on the HIT dataset of the wrinkling CNN variants trained in Chapter 8. They are:

- the original CNN, taking the filtered progress variable $\overline{c}$ as input,
- the LAM CNN, which takes $\overline{c}$ as input and was trained on additional spherical laminar flame fronts,
- the VORT CNN, taking $\overline{c}$ and the masked normalized vorticity magnitude ω^+ as inputs,
- and the TEST-FILT CNN, taking $\overline{c}$ and the test-filtered progress variable $\hat{\overline{c}}$ as inputs.

Visualizations of the CNN inputs, CNN output, and ground truth are shown for two representative flame fronts of the dataset in Section C.1 and C.2. Sample 1 is extracted from the early stages of the HIT, at $t = 0.4\tau$, where the flame front begins to wrinkled. It is part of the training dataset of all the CNNs. Sample 2 is taken from the later stages of the simulation, at $t = 3.7\tau$. The flame is significantly wrinkled, and it is part of the validation dataset.

Quantitative metrics computed on the HIT validation set are presented in the form of hexbin plots comparing the wrinkling factor predicted by the CNNs with the ground truth in Section C.3, and mean-squared errors in Section C.4.

C.1 Sample 1



(**a**) Original

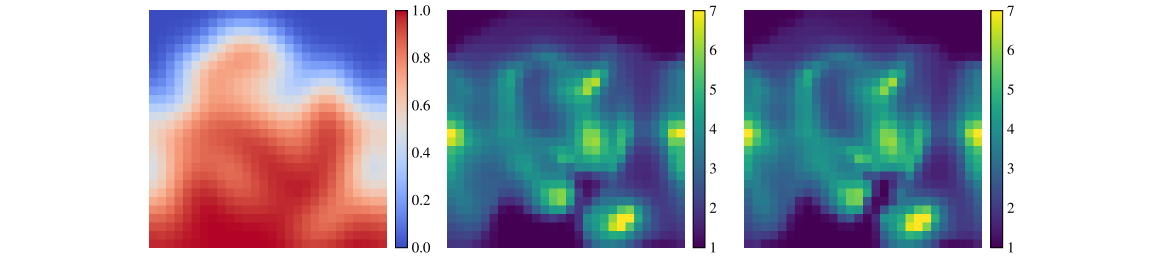
(**b**) LAM

(**c**) VORT

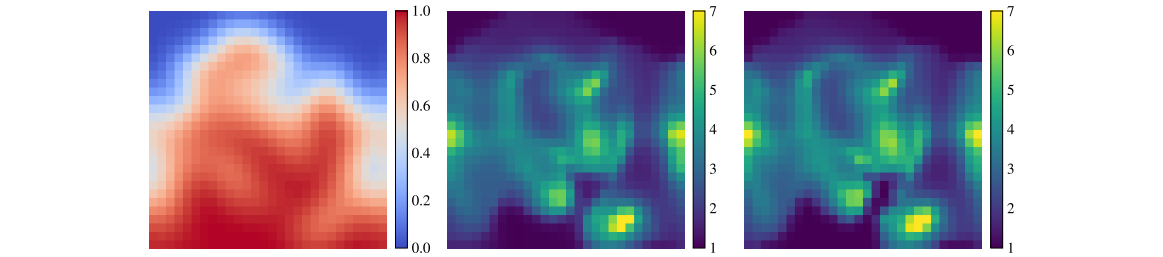
(**d**) TEST-FILT

Figure C.1: Visualizations of CNN inputs, CNN outputs, and ground truth on sample 1: (**a**) Original CNN ($\overline{c}$, Ξ_{CNN}, Ξ_{true}), (**b**) LAM CNN ($\overline{c}$, Ξ_{CNN}, Ξ_{true}), (**c**) VORT CNN ($\overline{c}$, masked ω^+, Ξ_{CNN}, Ξ_{true}), (**d**) TEST-FILT CNN ($\overline{c}$, $\hat{\overline{c}}$, Ξ_{CNN}, Ξ_{true}).

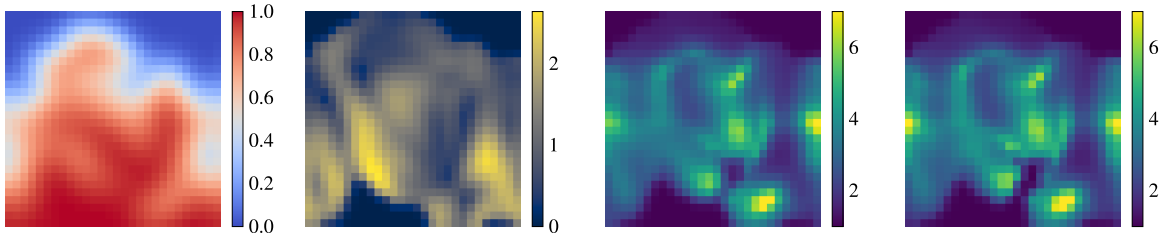
C.2 Sample 2



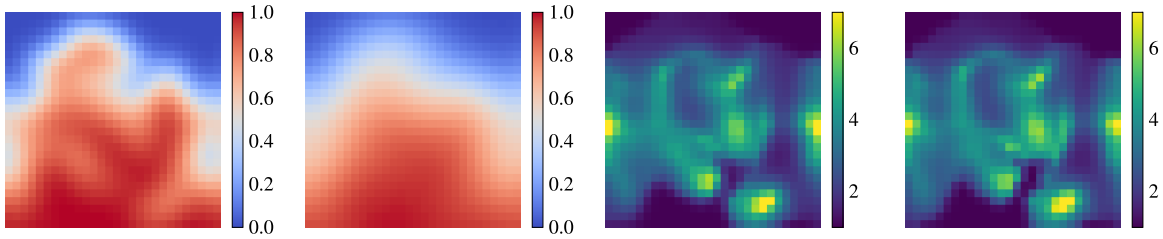
(**a**) Original



(**b**) LAM



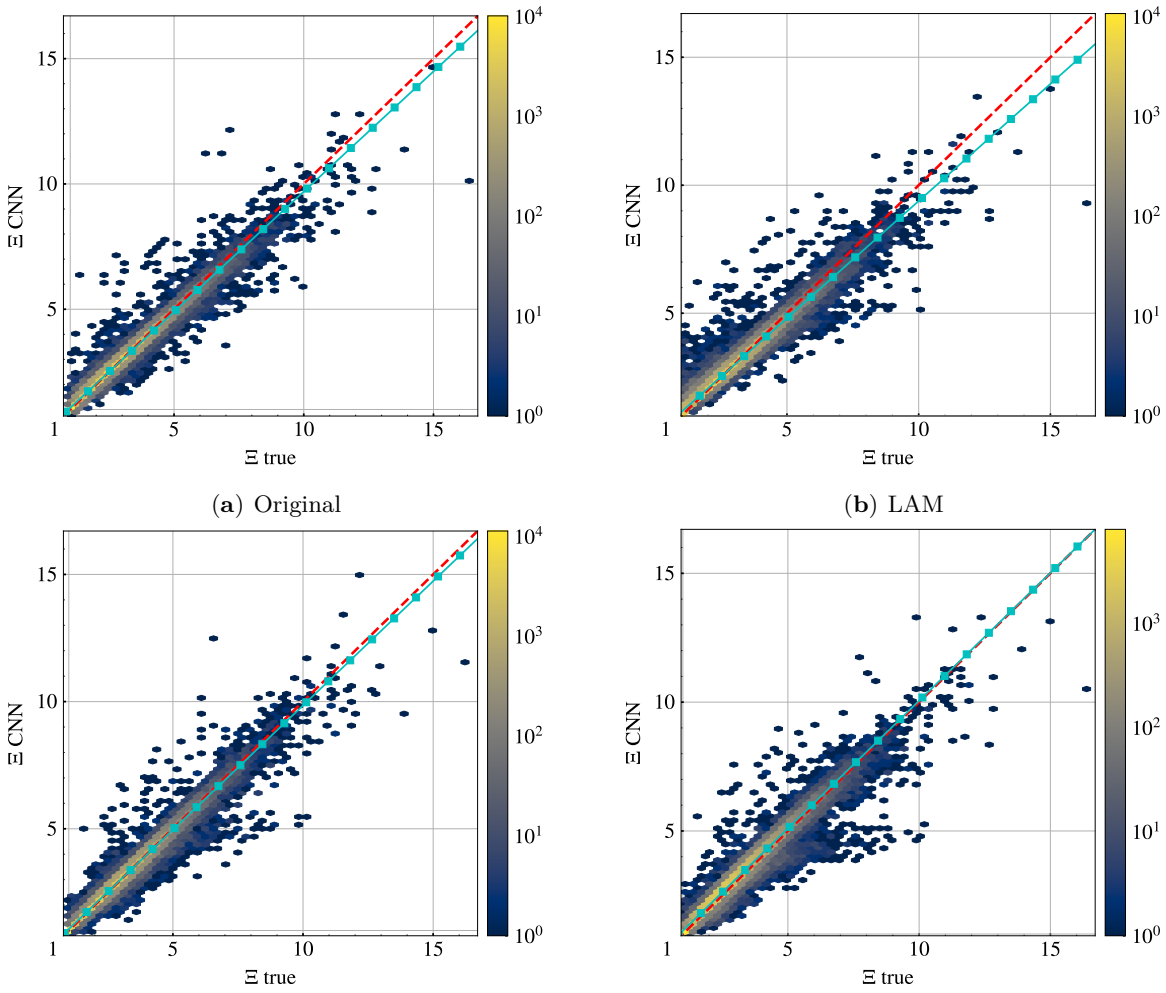
(**c**) VORT



(**d**) TEST-FILT

Figure C.2: Visualizations of CNN inputs, CNN outputs, and ground truth on sample 2: (**a**) Original CNN ($\overline{c}$, Ξ_{CNN}, Ξ_{true}), (**b**) LAM CNN ($\overline{c}$, Ξ_{CNN}, Ξ_{true}), (**c**) VORT CNN ($\overline{c}$, masked ω^+, Ξ_{CNN}, Ξ_{true}), (**d**) TEST-FILT CNN ($\overline{c}$, $\hat{\overline{c}}$, Ξ_{CNN}, Ξ_{true}).

C.3 Hexplots



(**a**) Original

(**b**) LAM

(**c**) VORT

(**d**) TEST-FILT

Figure C.3: Hexbin plots of model predictions against ground truth on the HIT validation set for the CNN models used in the Masri *a posteriori* simulations: (**a**) Original CNN, (**b**) LAM CNN, (**c**) VORT CNN, (**d**) TEST-FILT CNN.

C.4 Mean-squared errors

Original	LAM	VORT	TEST-FILT
0.066	0.085	0.068	0.093

Table C.1: Mean-squared errors on the HIT validation set.

Bibliography

[1] Chemical-kinetic mechanisms for combustion applications. https://web.eng.ucsd.edu/mae/groups/combustion/mechanism.html. Accessed: 2022-05-16. (Cited on page 68.)

[2] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: a system for large-scale machine learning. In *USENIX Symp. Oper. Syst. Des. Implement. (OSDI)*, volume 16, pages 265–283, 2016. (Cited on page 43.)

[3] R. Addanki, P. W. Battaglia, D. Budden, A. Deac, J. Godwin, T. Keck, W. L. S. Li, A. Sanchez-Gonzalez, J. Stott, S. Thakoor, and P. Veličković. Large-scale graph representation learning with very deep GNNs and self-supervision. *arXiv preprint arXiv:2107.09422*, 2021. (Cited on page 154.)

[4] E. Ajuria, A. Alguacil, M. Bauerheim, A. Misdariis, B. Cuenot, and E. Benazera. Towards a hybrid computational strategy based on deep learning for incompressible flows. In *AIAA Aviat. 2020 Forum*, 2020. (Cited on pages 50 and 55.)

[5] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. (Cited on page 4.)

[6] A. Alguacil, M. Bauerheim, M. C. Jacob, and S. Moreau. Predicting the propagation of acoustic waves using deep convolutional neural networks. *J. Sound Vib.*, 512:116285, 2021. (Cited on pages 55 and 56.)

[7] A. Alguacil, W. G. Pinto, M. Bauerheim, M. C. Jacob, and S. Moreau. Effects of boundary conditions in fully convolutional networks for learning spatio-temporal dynamics. In *Jt. Eur. Conf. Mach. Learn. Knowl. Discov. Databases. Appl. Data Sci. Track*, pages 102–117, 2021. (Cited on pages 50 and 154.)

[8] Z. Allen-Zhu, Y. Li, and Z. Song. A convergence theory for deep learning via over-parameterization. In *Int. Conf. Mach. Learn. (ICML)*, 2019. (Cited on page 45.)

[9] J. An, G. He, K. Luo, F. Qin, and B. Liu. Artificial neural network based chemical mechanisms for computationally efficient modeling of hydrogen/carbon monoxide/kerosene combustion. *Int. J. Hydrog. Energy*, 45(53):29594–29605, 2020. (Cited on page 56.)

[10] M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. Learning dexterous in-hand manipulation. *Int. J. Rob. Res.*, 39(1):3–20, 2020. (Cited on page 41.)

[11] A. J. Aspden. A numerical study of diffusive effects in turbulent lean premixed hydrogen flames. In *Proc. Combust. Inst.*, volume 36, pages 1997–2004, 2017. (Cited on page 24.)

[12] A. J. Aspden, M. S. Day, and J. B. Bell. Turbulence-flame interactions in lean premixed hydrogen: Transition to the distributed burning regime. *J. Fluid Mech.*, 680:287–320, 2011. (Cited on page 24.)

[13] A. J. Aspden, M. S. Day, and J. B. Bell. Turbulence-chemistry interaction in lean premixed hydrogen combustion. In *Proc. Combust. Inst.*, volume 35, pages 1321–1329, 2015. (Cited on page 24.)

[14] A. J. Aspden, M. S. Day, and J. B. Bell. Three-dimensional direct numerical simulation of turbulent lean premixed methane combustion with detailed kinetics. *Combust. Flame*, 166:266–283, 2016. (Cited on page 24.)

[15] A. Attili, S. Luca, D. Denker, F. Bisetti, and H. Pitsch. Turbulent flame speed and reaction layer thickening in premixed jet flames at constant Karlovitz and increasing Reynolds numbers. In *Proc. Combust. Inst.*, volume 38, pages 2939–2947, 2021. (Cited on pages 24, 58, 82, 87 and 206.)

[16] A. Attili, N. Sorace, L. Nista, C. Schumann, and A. Karimi. Investigation of the Extrapolation Performance of Machine Learning Models for LES of Turbulent Premixed Combustion. In *Proc. Eur. Combust. Meet. 2021*, 2021. (Cited on page 90.)

[17] C. Bailly and D. Juvé. A stochastic approach to compute subsonic noise using linearized euler's equations. In *5th AIAA/CEAS Aeroacoustics Conf. Exhib.*, pages 496–506, 1999. (Cited on page 118.)

[18] B. Baker, I. Akkaya, P. Zhokhov, J. Huizinga, J. Tang, A. Ecoffet, B. Houghton, R. Sampedro, and J. Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *arXiv preprint arXiv:2206.11795*, 2022. (Cited on page 4.)

[19] G. Balarac, H. Pitsch, and V. Raman. Development of a dynamic model for the subfilter scalar variance using the concept of optimal estimators. *Phys. Fluids*, 20(3):035114, 2008. (Cited on pages 116 and 127.)

[20] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner. Learning data-driven discretizations for partial differential equations. *Proc. Natl. Acad. Sci.*, 116(31):15344–15349, 2019. (Cited on page 55.)

[21] J. Bardina, J. H. Ferziger, and W. C. Reynolds. Improved subgrid-scale models for large-eddy simulation. In *13th Fluid and Plasma Dynamics Conference*, page 1357, 1980. (Cited on page 196.)

[22] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, 2000. (Cited on page 11.)

[23] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, C. Gulcehre, F. Song, A. Ballard, J. Gilmer, G. Dahl, A. Vaswani, K. Allen, C. Nash, V. Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.0, jun 2018. (Cited on pages 48 and 154.)

[24] F. d. A. Belbute-Peres, T. D. Economon, and J. Z. Kolter. Combining Differentiable PDE Solvers and Graph Neural Networks for Fluid Flow Prediction. In *Int. Conf. Mach. Learn. (ICML)*, 2020. (Cited on pages 55 and 154.)

[25] M. Belkin, D. Hsu, S. Ma, and S. Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proc. Natl. Acad. Sci.*, 116(32):15849–15854, 2019. (Cited on page 45.)

[26] P. Bénard, G. Lartigue, V. Moureau, and R. Mercier. Large-eddy simulation of the lean-premixed preccinsta burner with wall heat loss. In *Proc. Combust. Inst.*, volume 37, pages 5233–5243, 2019. (Cited on page 2.)

[27] L. Berger, A. Attili, and H. Pitsch. Synergistic interactions of thermodiffusive instabilities and turbulence in lean hydrogen flames. *Combust. Flame*, 244:112254, 2022. (Cited on pages 3 and 206.)

[28] L. Berger, K. Kleinheinz, A. Attili, and H. Pitsch. Characteristic patterns of thermodiffusively unstable premixed lean hydrogen flames. *Proc. Combust. Inst.*, 37:1879–1886, 2019. (Cited on page 206.)

[29] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. (Cited on pages 40 and 42.)

[30] J. Blasco, N. Fueyo, C. Dopazo, and J. Ballester. Modelling the temporal evolution of a reduced combustion chemical system with an artificial neural network. *Combust. Flame*, 113(1-2):38–52, 1998. (Cited on page 56.)

[31] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. (Cited on page 43.)

[32] M. Bode, M. Gauding, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, and H. Pitsch. Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. In *Proc. Combust. Inst.*, volume 38, pages 2617–2625, 2021. (Cited on pages 56 and 57.)

[33] M. Boger, D. Veynante, H. Boughanem, and A. Trouvé. Direct numerical simulation analysis of flame surface density concept for large eddy simulation of turbulent premixed combustion. In *Symp. Combust.*, volume 27, pages 917–925, 1998. (Cited on pages 28, 29 and 118.)

[34] R. Borghi. On the Structure and Morphology of Turbulent Premixed Flames. In *Recent Adv. Aerosp. Sci.*, pages 117–138. Plenum Press, Boston, MA, 1985. (Cited on page 21.)

[35] K. N. Bray, M. Champion, P. A. Libby, and N. Swaminathan. Finite rate chemistry and presumed PDF models for premixed turbulent combustion. *Combust. Flame*, 146(4):665–673, 2006. (Cited on pages 115 and 133.)

[36] K. N. C. Bray. Studies of the turbulent burning velocity. *Proc. R. Soc. London. Ser. A Math. Phys. Sci.*, 431(1882):315–335, 1990. (Cited on page 115.)

[37] K. N. C. Bray and R. Cant. Some applications of kolmogorov's turbulence research in the field of combustion. *Proc. R. Soc. A: Math. Phys. Eng. Sci.*, 434(1890):217–240, 1991. (Cited on page 24.)

[38] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 33, 2020. (Cited on pages 4 and 43.)

[39] S. L. Brunton, B. R. Noack, and P. Koumoutsakos. Machine Learning for Fluid Mechanics. *Annu. Rev. Fluid Mech.*, 52:477–508, 2020. (Cited on page 54.)

[40] T. Butler and P. O'Rourke. A numerical method for two dimensional unsteady reacting flows. In *Symp. Combust.*, volume 16, pages 1503–1515, jan 1977. (Cited on page 29.)

[41] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 33, 2020. (Cited on pages 41 and 206.)

[42] A. Cellier, C. J. Lapeyre, G. Öztarlik, T. Poinsot, T. Schuller, and L. Selle. Detection of precursors of combustion instability using convolutional recurrent neural networks. *Combust. Flame*, 233:111558, 2021. (Cited on page 3.)

[43] N. Chakraborty and R. S. Cant. Effects of Lewis number on scalar transport in turbulent premixed flames. *Phys. Fluids*, 21(3):1–12, 2009. (Cited on page 118.)

[44] N. Chakraborty and M. Klein. A priori direct numerical simulation assessment of algebraic flame surface density models for turbulent premixed flames in the context of large eddy simulation. *Phys. Fluids*, 20(8):1–15, 2008. (Cited on page 31.)

[45] F. Charlette, C. Meneveau, and D. Veynante. A power-law flame wrinkling model for LES of premixed turbulent combustion Part II: Dynamic formulation. *Combust. Flame*, 131(1-2):181–197, 2002. (Cited on pages 31, 33, 159 and 160.)

[46] F. Charlette, C. Meneveau, and D. Veynante. A power-law wrinkling model for LES of premixed turbulent combustion Part I: non-dynamic formulation and initial tests. *Combust. Flame*, 131(1-2):159–180, 2002. (Cited on pages 31, 32, 33 and 118.)

[47] A. Chatzopoulos and S. Rigopoulos. A chemistry tabulation approach via rate-controlled constrained equilibrium (rcce) and artificial neural networks (anns), with application to turbulent non-premixed ch4/h2/n2 flames. In *Proc. Combust. Inst.*, volume 34, pages 1465–1473, 2013. (Cited on page 56.)

[48] J.-Y. Chen, J. Blasco, N. Fueyo, and C. Dopazo. An economical strategy for storage of chemical kinetics: Fitting in situ adaptive tabulation with artificial neural networks. In *Proc. Combust. Inst.*, volume 28, pages 115–121, 2000. (Cited on page 56.)

[49] M. Chen, A. Beutel, P. Covington, S. Jain, F. Belletti, and E. H. Chi. Top-k off-policy correction for a reinforce recommender system. In *ACM Int. Conf. Web Search Data Min. (WSDM)*, pages 456–464, 2019. (Cited on page 43.)

[50] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. (Cited on page 4.)

[51] M. Chen, Z. Wei, Z. Huang, B. Ding, and Y. Li. Simple and Deep Graph Convolutional Networks. In *Int. Conf. Mach. Learn. (ICML)*, 2020. (Cited on page 154.)

[52] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 30, 2018. (Cited on page 55.)

[53] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Int. Conf. Mach. Learn. (ICML)*, 2020. (Cited on pages 41 and 206.)

[54] X. Chen and K. He. Exploring simple siamese representation learning. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021. (Cited on page 41.)

[55] Z. X. Chen, S. Iavarone, G. Ghiasi, V. Kannan, G. D'Alessio, A. Parente, and N. Swaminathan. Application of machine learning for filtered density function closure in MILD combustion. *Combust. Flame*, 225:160–179, 2021. (Cited on pages 57 and 59.)

[56] L. Cheng, E. A. Illarramendi, G. Bogopolsky, M. Bauerheim, and B. Cuenot. Using neural networks to solve the 2d poisson equation for electric field computation in plasma fluid simulations. *arXiv preprint arXiv:2109.13076*, 2021. (Cited on page 48.)

[57] W. L. Chiang, Y. Li, X. Liu, S. Bengio, S. Si, and C. J. Hsieh. Cluster-GCN: An efficient algorithm for training deep and large graph convolutional networks. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pages 257–266, 2019. (Cited on page 154.)

[58] F. Chollet. *Deep learning with Python*. Manning Publications, 2017. (Cited on page 40.)

[59] S. T. Chong, M. Hassanaly, H. Koo, M. E. Mueller, V. Raman, and K.-P. Geigle. Large eddy simulation of pressure and dilution-jet effects on soot formation in a model aircraft swirl combustor. *Combust. Flame*, 192:452–472, 2018. (Cited on page 2.)

[60] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022. (Cited on page 4.)

[61] F. Christo, A. Masri, E. Nebot, and S. Pope. An integrated pdf/neural network approach for simulating turbulent reacting systems. In *Symp. Combust.*, volume 26, pages 43–48, 1996. (Cited on page 56.)

[62] W. T. Chung, K. S. Jung, J. Chen, and M. Ihme. The bearable lightness of big data: Towards massive public datasets in scientific machine learning. In *AI Sci. Work. (ICML)*, 2022. (Cited on pages 57 and 204.)

[63] W. T. Chung, A. A. Mishra, N. Perakis, and M. Ihme. Data-assisted combustion simulations with dynamic submodel assignment using random forests. *Combust. Flame*, 227:172–185, 2021. (Cited on page 205.)

[64] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3D U-net: Learning dense volumetric segmentation from sparse annotation. In *Int. Conf. Med. image Comput. Comput. Interv.*, pages 424–432, 2016. (Cited on page 49.)

[65] T. S. Cohen, M. Geiger, and M. Weiler. A General Theory of Equivariant CNNs on Homogeneous Spaces. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019. (Cited on page 205.)

[66] T. S. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling. Gauge Equivariant Convolutional Networks and the Icosahedral CNN. In *Int. Conf. Mach. Learn. (ICML)*, 2019. (Cited on page 205.)

[67] O. Colin, F. Ducros, D. Veynante, and T. Poinsot. A thickened flame model for large eddy simulations of turbulent premixed combustion. *Phys. Fluids*, 12(7):1843, 2000. (Cited on pages 29, 31, 159, 160, 196 and 197.)

[68] O. Colin and M. Rudgyard. Development of High-Order Taylor-Galerkin Schemes for LES. *J. Comput. Phys.*, 162(2):338–371, 2000. (Cited on pages 36 and 119.)

[69] A. W. Cook and J. J. Riley. A subgrid model for equilibrium chemistry in turbulent flows. *Phys. Fluids*, 6(8):2868–2870, 1994. (Cited on pages 115 and 116.)

[70] G. Damköhler. Der einfluss der turbulenz auf die flammengeschwindigkeit in gasgemischen. *Z. Elektrochem. angew. phys. Chem.*, 46(11):601–626, 1940. (Cited on page 21.)

[71] O. R. Darbyshire and N. Swaminathan. A presumed joint pdf model for turbulent combustion with varying equivalence ratio. *Combust. Sci. Technol.*, 184(12):2036–2067, 2012. (Cited on page 115.)

[72] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, 2019. (Cited on page 43.)

[73] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020. (Cited on page 4.)

[74] T. Ding, T. Readshaw, S. Rigopoulos, and W. Jones. Machine learning tabulation of thermochemistry in turbulent combustion: An approach based on hybrid flamelet/random data and multiple multilayer perceptrons. *Combust. Flame*, 231:111493, 2021. (Cited on page 56.)

[75] P. Domingo, L. Vervisch, S. Payet, and R. Hauguel. DNS of a premixed turbulent V flame and LES of a ducted flame using a FSD-PDF subgrid scale closure with FPI-tabulated chemistry. *Combust. Flame*, 143(4):566–586, 2005. (Cited on pages 115, 117 and 133.)

[76] P. Domingo, L. Vervisch, and D. Veynante. Large-eddy simulation of a lifted methane jet flame in a vitiated coflow. *Combust. Flame*, 152(3):415–432, 2008. (Cited on pages 117 and 205.)

[77] J. F. Driscoll, J. H. Chen, A. W. Skiba, C. D. Carter, E. R. Hawkes, and H. Wang. Premixed flames subjected to extreme turbulence: Some questions and recent answers. *Prog. Energy Combust. Sci.*, 76:100802, 2020. (Cited on pages 21, 24 and 184.)

[78] F. Duchaine, S. Jauré, D. Poitou, E. Quémerais, G. Staffelbach, T. Morel, and L. Gicquel. Analysis of high performance conjugate heat transfer with the OpenPALM coupler. *Comput. Sci. Discov.*, 8(1):15003, 2015. (Cited on page 143.)

[79] V. Dumoulin, E. Perez, N. Schucher, F. Strub, H. d. Vries, A. Courville, and Y. Bengio. Feature-wise transformations. *Distill*, 2018. https://distill.pub/2018/feature-wise-transformations. (Cited on page 204.)

[80] K. Duraisamy. Perspectives on machine learning-augmented reynolds-averaged and large eddy simulation models of turbulence. *Phys. Rev. Fluids*, 6(5):050504, 2021. (Cited on pages 5 and 54.)

[81] K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence Modeling in the Age of Data. *Annu. Rev. Fluid Mech.*, 51:357–377, 2019. (Cited on page 54.)

[82] T. Echekki and H. Mirgolbabaei. Principal component transport in turbulent combustion: A posteriori analysis. *Combust. and Flame*, 162(5):1919–1933, 2015. (Cited on page 56.)

[83] T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. Su2: An open-source suite for multiphysics simulation and design. *AIAA J.*, 54(3):828–846, 2016. (Cited on page 55.)

[84] H. Eivazi, H. Veisi, M. H. Naderi, and V. Esfahanian. Deep neural networks for nonlinear model order reduction of unsteady flows. *Phys. Fluids*, 32(10):105104, 2020. (Cited on page 54.)

[85] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. Technical report, Université de Montréal, 2009. (Cited on page 49.)

[86] A. Ern and V. Giovangigli. *Multicomponent transport algorithms*, volume 24. Springer Science & Business Media, 1994. (Cited on page 13.)

[87] A. Ern and V. Giovangigli. Thermal diffusion effects in hydrogen-air and methane-air flames. *Combust. Theory Model.*, 2(4):349–372, 1998. (Cited on page 12.)

[88] L. Esclapez, P. C. Ma, E. Mayhew, R. Xu, S. Stouffer, T. Lee, H. Wang, and M. Ihme. Fuel effects on lean blow-out in a realistic gas turbine combustor. *Combust. Flame*, 181:82–99, 2017. (Cited on page 2.)

[89] A. Favre. Turbulence: Space-time statistical properties and behavior in supersonic flows. *Phys. Fluids*, 26(10):2851–2863, 1983. (Cited on page 25.)

[90] H. Federer. Curvature measures. *Trans. Am. Math. Soc.*, 93(3):418–491, 1959. (Cited on page 30.)

[91] W. Fedus, B. Zoph, and N. Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(120):1–39, 2022. (Cited on page 45.)

[92] V. Feldman. Does learning require memorization? a short tale about a long tail. In *Proc. 52nd Annu. ACM SIGACT Symp. Theory Comput. (STOC)*, 2020. (Cited on page 45.)

[93] V. Feldman and C. Zhang. What neural networks memorize and why: Discovering the long tail via influence estimation. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 33, 2020. (Cited on page 45.)

[94] X. Feng, M. Ouyang, X. Liu, L. Lu, Y. Xia, and X. He. Thermal runaway mechanism of lithium ion battery for electric vehicles: A review. *Energy Storage Mater.*, 10:246–267, 2018. (Cited on page 3.)

[95] J. H. Ferziger. Large Eddy Numerical Simulations of Turbulent Flows. *AIAA J.*, 15(9):1261–1267, 1977. (Cited on page 66.)

[96] J. H. Ferziger and T. Echekki. A Simplified Reaction Rate Model and its Application to the Analysis of Premixed Flames. *Combust. Sci. Technol.*, 89(5-6):293–315, 1993. (Cited on page 17.)

[97] F. Fichot, F. Lacas, D. Veynante, and S. Candel. One-Dimensional Propagation of a Premixed Turbulent Flame With a Balance Equation for the Flame Surface Density. *Combust. Sci. Technol.*, 90(1-4):35–60, 1993. (Cited on pages 211 and 214.)

[98] 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. (Cited on pages 29 and 115.)

[99] L. L. Franke, A. K. Chatzopoulos, and S. Rigopoulos. Tabulation of combustion chemistry via artificial neural networks (anns): Methodology and application to les-pdf simulation of sydney flame l. *Combust. Flame*, 185:245–260, 2017. (Cited on page 56.)

[100] J. Frankle and M. Carbin. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *Int. Conf. Learn. Represent. (ICLR)*, 2019. (Cited on page 102.)

[101] H. Frezat, G. Balarac, J. Le Sommer, R. Fablet, and R. Lguensat. Physical invariance in neural networks for subgrid-scale scalar flux modeling. *Phys. Rev. Fluids*, 6(2):24607, 2021. (Cited on page 56.)

[102] K. Fukami, K. Fukagata, and K. Taira. Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.*, 870:106–120, 2019. (Cited on page 54.)

[103] K. Fukami, K. Fukagata, and K. Taira. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *J. Fluid Mech.*, 909:A9, 2021. (Cited on page 54.)

[104] K. Fukami, R. Maulik, N. Ramachandra, K. Fukagata, and K. Taira. Probabilistic neural network-based reduced-order surrogate for fluid flows. In *Work. Mach. Learn. Phys. Sci. (NeurIPS)*, 2020. (Cited on page 54.)

[105] K. Fukami, Y. Nabae, K. Kawai, and K. Fukagata. Synthetic turbulent inflow generator using machine learning. *Phys. Rev. Fluids*, 4(6):064603, 2019. (Cited on page 54.)

[106] C. Fureby. A fractal flame-wrinkling large eddy simulation model for premixed turbulent combustion. In *Proc. Combust. Inst.*, volume 30, pages 593–601, 2005. (Cited on page 31.)

[107] H. Gao, L. Sun, and J.-X. Wang. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Phys. Fluids*, 33(7):073603, 2021. (Cited on page 54.)

[108] E. Garnier, N. Adams, and P. Sagaut. *Large eddy simulation for compressible flows*. Springer Science & Business Media, 2009. (Cited on page 25.)

[109] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *Int. Conf. Learn. Represent. (ICLR)*, 2019. (Cited on page 62.)

[110] N. Geneva and N. Zabaras. Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. *J. Comput. Phys.*, 403:109056, 2020. (Cited on page 54.)

[111] S. Ghosal. An analysis of numerical errors in large-eddy simulations of turbulence. *J. Comput. Phys.*, 125(1):187–206, 1996. (Cited on page 60.)

[112] B. H. Gilding and R. Kersner. *Travelling Waves in Nonlinear Diffusion-Convection Reaction*. Birkhäuser, Basel, 2004. (Cited on page 211.)

[113] V. Giovangigli. Convergent iterative methods for multicomponent diffusion. *IMPACT Comput. Sci. Eng.*, 3(3):244–276, 1991. (Cited on page 13.)

[114] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. *J. Mach. Learn. Res.*, 9:249–256, 2010. (Cited on page 84.)

[115] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016. (Cited on pages 40, 47, 49, 62 and 102.)

[116] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 26, 2014. (Cited on page 41.)

[117] F. C. Gouldin. An application of fractals to modeling premixed turbulent flames. *Combust. Flame*, 68(3):249–266, 1987. (Cited on pages 31 and 32.)

[118] F. C. Gouldin, K. N. C. Bray, and J. Y. Chen. Chemical closure model for fractal flamelets. *Combust. Flame*, 77(3-4):241–259, 1989. (Cited on page 31.)

[119] T. L. Griffiths, N. Chater, C. Kemp, A. Perfors, and J. B. Tenenbaum. Probabilistic models of cognition: exploring representations and inductive biases. *Trends Cogn. Sci.*, 14(8):357–364, 2010. (Cited on page 48.)

[120] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap our own latent: a new approach to self-supervised learning. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 33, 2020. (Cited on pages 41 and 206.)

[121] S. Gu, E. Holly, T. Lillicrap, and S. Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3389–3396, 2017. (Cited on page 41.)

[122] L. Guastoni, A. Güemes, A. Ianiro, S. Discetti, P. Schlatter, H. Azizpour, and R. Vinuesa. Convolutional-network models to predict wall-bounded turbulence from wall quantities. *J. Fluid Mech.*, 928:A27, 2021. (Cited on page 55.)

[123] S. R. Gubba, S. S. Ibrahim, W. Malalasekera, and A. R. Masri. Measurements and LES calculations of turbulent premixed flame propagation past repeated obstacles. *Combust. Flame*, 158(12):2465–2481, 2011. (Cited on pages 163 and 164.)

[124] D. Guest, K. Cranmer, and D. Whiteson. Deep learning and its application to lhc physics. *Annu. Rev. Nucl. Part. Sci.*, 68:161–181, 2018. (Cited on page 5.)

[125] Ö. L. Gülder and G. J. Smallwood. Inner cutoff scale of flame surface wrinkling in turbulent premixed flames. *Combust. Flame*, 103(1-2):107–114, 1995. (Cited on page 32.)

[126] J. Gullbrand and F. K. Chow. Investigation of numerical errors, subfilter-scale models, and subgrid-scale models in turbulent channel flow simulations. In *Proc. Summer Program, Cent. Turbul. Res.*, pages 87–104, 2002. (Cited on page 25.)

[127] X. Guo, W. Li, and F. Iorio. Convolutional Neural Networks for Steady Flow Approximation. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pages 481–490, 2016. (Cited on page 54.)

[128] E. Haghighat, M. Raissi, A. Moure, H. Gomez, and R. Juanes. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput. Methods Appl. Mech. Eng.*, 379:113741, 2021. (Cited on page 55.)

[129] B. Hakberg and A. D. Gosman. Analytical determination of turbulent flame speed from combustion models. In *Symp. Combust.*, volume 20, pages 225–232, 1985. (Cited on page 211.)

[130] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 30, 2017. (Cited on page 154.)

[131] M. Hansinger, Y. Ge, and M. Pfitzner. Deep residual networks for flamelet/progress variable tabulation with application to a piloted flame with inhomogeneous inlet. *Combust. Sci. Technol.*, 194(8):1587–1613, 2022. (Cited on page 56.)

[132] 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. (Cited on page 55.)

[133] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009. (Cited on pages 40, 42, 43 and 81.)

[134] E. R. Hawkes, O. Chatakonda, H. Kolla, A. R. Kerstein, and J. H. Chen. A petascale direct numerical simulation study of the modelling of flame wrinkling for large-eddy simulations in intense turbulence. *Combust. Flame*, 159(8):2690–2703, 2012. (Cited on pages 31 and 32.)

[135] D. C. Haworth. Progress in probability density function methods for turbulent reacting flows. *Prog. Energy Combust. Sci.*, 36(2):168–259, 2010. (Cited on pages 28 and 115.)

[136] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020. (Cited on pages 41 and 206.)

[137] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016. (Cited on pages 4, 50 and 55.)

[138] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021. (Cited on page 62.)

[139] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. (Cited on page 41.)

[140] J. O. Hinze. *Turbulence*. McGraw-Hill, 1975. (Cited on page 18.)

[141] J. Hirschfelder, R. B. Bird, and C. F. Curtiss. *Molecular theory of gases and liquids*. Structure of matter series. Wiley, New York, NY, 1964. (Cited on page 13.)

[142] P. Holl, V. Koltun, K. Um, and N. Thuerey. phiflow: A differentiable pde solving framework for deep learning via physical simulations. In *Work. Differ. Vis. Graph. Phys. Mach. Learn. (NeurIPS)*, 2020. (Cited on page 55.)

[143] L. Hou, Y. Cheng, N. Shazeer, N. Parmar, Y. Li, P. Korfiatis, T. M. Drucker, D. J. Blezek, and X. Song. High Resolution Medical Image Analysis with Spatial Partitioning. In *Med. Imag. Meets NeurIPS Work. (NeurIPS)*, 2019. (Cited on page 145.)

[144] C.-W. Huang, J. H. Lim, and A. C. Courville. A variational perspective on diffusion-based generative models and score matching. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 34, 2021. (Cited on page 55.)

[145] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck. Music transformer: Generating music with long-term structure. In *Int. Conf. Learn. Represent. (ICLR)*, 2019. (Cited on page 4.)

[146] M. Ihme, W. T. Chung, and A. A. Mishra. Combustion machine learning: Principles, progress and prospects. *Prog. Energy Combust. Sci.,*, 91:101010, 2022. (Cited on pages 5 and 56.)

[147] M. Ihme, C. Schmitt, and H. Pitsch. Optimal artificial neural networks and tabulation methods for chemistry representation in les of a bluff-body swirl-stabilized flame. *Proc. Combust. Inst.*, 32:1527–1535, 2009. (Cited on page 56.)

[148] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Int. Conf. Mach. Learn. (ICML)*, 2015. (Cited on pages 48 and 50.)

[149] T. Ishida, P. A. Davidson, and Y. Kaneda. On the decay of isotropic turbulence. *J. Fluid Mech.*, 564:455–475, 2006. (Cited on page 121.)

[150] T. Jaravel. *Prediction of pollutants in gas turbines using Large Eddy Simulation*. PhD thesis, 2016. (Cited on pages 30 and 97.)

[151] Jia Deng, Wei Dong, R. Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 248–255, 2009. (Cited on pages 41, 43, 49 and 83.)

[152] K. Jigjid, Y. Minamoto, N. A. K. Doan, and M. Tanahashi. Sgs reaction rate modelling for mild combustion based on machine-learning combustion mode classification: Development and a priori study. In *Proc. Combust. Inst.*, 2022. (Cited on pages 57, 59 and 204.)

[153] X. Jin, S. Cai, H. Li, and G. E. Karniadakis. Nsfnets (navier-stokes flow nets): Physics-informed neural networks for the incompressible navier-stokes equations. *J. Comput. Phys.*, 426:109951, 2021. (Cited on page 55.)

[154] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. (Cited on pages 5 and 43.)

[155] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila. Alias-Free Generative Adversarial Networks. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 34, 2021. (Cited on pages 4 and 43.)

[156] G. Karypis and V. Kumar. Multilevel k-way Partitioning Scheme for Irregular Graphs. *J. Parallel Distrib. Comput.*, 48(1):96–129, 1998. (Cited on page 143.)

[157] A. Kashefi, D. Rempe, and L. J. Guibas. A point-cloud deep learning framework for prediction of fluid flow fields on irregular geometries. *Phys. Fluids*, 33(2), 2021. (Cited on pages 54 and 205.)

[158] C. Kasten, J. Shin, R. Sandberg, M. Pfitzner, N. Chakraborty, and M. Klein. Modeling subgrid-scale scalar dissipation rate in turbulent premixed flames using gene expression programming and deep artificial neural networks. *Phys. Fluids*, 34:085113, 2022. (Cited on pages 57, 59 and 204.)

[159] F. B. Keil, M. Klein, and N. Chakraborty. Sub-grid Reaction Progress Variable Variance Closure in Turbulent Premixed Flames. *Flow, Turbul. Combust.*, 106(4):1195–1212, 2021. (Cited on pages 116, 117 and 118.)

[160] R. Keppeler, E. Tangermann, U. Allaudin, and M. Pfitzner. LES of low to high turbulent combustion in an elevated pressure environment. *Flow, Turbul. Combust.*, 92(3):767–802, 2014. (Cited on pages 31 and 34.)

[161] S. Kheirkhah and Ö. L. Gülder. A revisit to the validity of flamelet assumptions in turbulent premixed combustion and implications for future research. *Combust. Flame*, 239(239):111635, 2022. (Cited on page 24.)

[162] H. Kim, J. Kim, S. Won, and C. Lee. Unsupervised deep learning for super-resolution reconstruction of turbulence. *J. Fluid Mech.*, 910:A29, 2021. (Cited on pages 54 and 206.)

[163] J. Kim and C. Lee. Deep unsupervised learning of turbulence for inflow generation at various reynolds numbers. *J. Comput. Phys.*, 406:109216, 2020. (Cited on page 54.)

[164] D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. In *Int. Conf. Learn. Represent. (ICLR)*, 2015. (Cited on pages 44, 84 and 126.)

[165] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *Int. Conf. Learn. Represent. (ICLR)*, 2014. (Cited on page 41.)

[166] R. Knikker, D. Veynante, and C. Meneveau. A priori testing of a similarity model for large eddy simulations of turbulent premixed combustion. In *Proc. Combust. Inst.*, volume 29, pages 2105–2111, 2002. (Cited on page 32.)

[167] R. Knikker, D. Veynante, and C. Meneveau. A dynamic flame surface density model for large eddy simulation of turbulent premixed combustion. *Phys. Fluids*, 16(11):91–95, 2004. (Cited on page 31.)

[168] E. Knudsen, S. H. Kim, and H. Pitsch. An analysis of premixed flamelet models for large eddy simulation of turbulent combustion. *Phys. Fluids*, 22(11):115109, 2010. (Cited on pages 115, 116, 117, 127 and 133.)

[169] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, and S. Hoyer. Machine learning – accelerated computational fluid dynamics. *Proc. Natl. Acad. Sci.*, 118(21), 2021. (Cited on page 55.)

[170] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*, 2022. (Cited on page 4.)

[171] H. Kolla, J. W. Rogerson, and N. Swaminathan. Validation of a turbulent flame speed model across combustion regimes. *Combust. Sci. Technol.*, 182(3):284–308, 2010. (Cited on page 211.)

[172] A. Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. In *Dokl. Akad. Nauk SSSR*, volume 30, pages 301–305, 1941. (Cited on page 19.)

[173] A. N. Kolmogorov, I. G. Petrovskii, and N. Piskunov. Investigation of the equation of diffusion combined with increasing of the substance and its application to a biology problem. *Bull. Moscow State Univ. Ser. A: Math. Mech*, 1(6):1–25, 1937. (Cited on page 211.)

[174] R. H. Kraichnan. Diffusion by a random velocity field. *Phys. Fluids*, 13(1):22–31, 1970. (Cited on page 96.)

[175] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, 2009. (Cited on page 41.)

[176] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 25, 2012. (Cited on pages 4, 43 and 48.)

[177] A. Krogh and J. A. Hertz. A simple weight decay can improve generalization. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 4, 1992. (Cited on page 45.)

[178] K. K.-y. Kuo and R. Acharya. *Fundamentals of turbulent and multiphase combustion*. John Wiley & Sons, 2012. (Cited on pages 10, 12 and 14.)

[179] I. E. Lagaris, A. Likas, and D. I. Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Netw.*, 9(5):987–1000, 1998. (Cited on pages 54 and 56.)

[180] N. Lamarque. *Schémas numériques et conditions limites pour la simulation aux grandes échelles de la combustion diphasique dans les foyers d'hélicoptère*. PhD thesis, 2007. (Cited on page 36.)

[181] I. Langella, N. A. Doan, N. Swaminathan, and S. B. Pope. Study of subgrid-scale velocity models for reacting and nonreacting flows. *Phys. Rev. Fluids*, 3(5):1–24, 2018. (Cited on page 196.)

[182] I. Langella, N. Swaminathan, F. A. Williams, and J. Furukawa. Large-Eddy Simulation of Premixed Combustion in the Corrugated-Flamelet Regime. *Combust. Sci. Technol.*, 188(9):1565–1591, 2016. (Cited on page 115.)

[183] 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. (Cited on pages 5, 50, 57, 58, 59, 75, 76, 111 and 125.)

[184] S. Lapointe and G. Blanquart. A priori filtered chemical source term modeling for LES of high Karlovitz number premixed flames. *Combust. Flame*, 176:500–510, 2017. (Cited on page 115.)

[185] C. K. Law. *Combustion physics*. Cambridge University Press, 2010. (Cited on pages 10, 12 and 14.)

[186] Y. LeCun, S. Chopra, R. Hadsell, M. A. Ranzato, and F. J. Huang. A Tutorial on Energy-Based Learning. In *Predict. Struct. Data*. MIT Press, 2006. (Cited on page 40.)

[187] J. P. Legier, T. Poinsot, and D. Veynante. Dynamically thickened flame LES model for premixed and non-premixed turbulent combustion. In *Proc. Summer Program, Cent. Turbul. Res.*, pages 157–168, 2000. (Cited on page 37.)

[188] A. Leonard. Energy cascade in large-eddy simulations of turbulent fluid flows. *Adv. Geophys.*, 18(PA):237–248, 1975. (Cited on pages 34 and 66.)

[189] A. Lewkowycz, A. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022. (Cited on page 4.)

[190] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018. (Cited on page 100.)

[191] Q. Li, Z. Han, and X.-M. Wu. Deeper insights into Graph Convolutional Networks for Semi-Supervised Learning. In *Thirty-Second AAAAI Conf. Artif. Intell.*, 2018. (Cited on page 154.)

[192] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Multipole Graph Neural Operator for Parametric Partial Differential Equations. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 33, 2020. (Cited on page 154.)

[193] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. In *Int. Conf. Learn. Represent. (ICLR)*, 2021. (Cited on pages 55 and 205.)

[194] D. K. Lilly. The representation of small-scale turbulence in numerical simulation experiments. In *IBM Scientific Computing Symposium on Environmental Sciences*, pages 195–210. IBM DP Division, Whitey Plains, N. Y., 1967. (Cited on pages 36 and 196.)

[195] D. K. Lilly. A proposed modification of the Germano subgrid-scale closure method. *Phys. Fluids A Fluid Dyn.*, 4(3):633–635, 1992. (Cited on page 117.)

[196] J. Ling, R. Jones, and J. Templeton. Machine learning strategies for systems with invariance properties. *J. Comput. Phys.*, 318:22–35, 2016. (Cited on page 205.)

[197] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.*, 807:155–166, 2016. (Cited on page 56.)

[198] A. N. Lipatnikov and J. Chomiak. Turbulent flame speed and thickness: Phenomenology, evaluation, and application in multi-dimensional simulations. *Prog. Energy Combust. Sci.*, 28(1):1–74, 2002. (Cited on page 211.)

[199] B. Liu, J. Tang, H. Huang, and X. Y. Lu. Deep learning methods for super-resolution reconstruction of turbulent flows. *Phys. Fluids*, 32(2):025105, 2020. (Cited on page 54.)

[200] M. Liu, H. Gao, and S. Ji. Towards Deeper Graph Neural Networks. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pages 338–348, 2020. (Cited on page 154.)

[201] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised Image-to-Image Translation Networks. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 30, 2017. (Cited on page 206.)

[202] M.-Y. Liu, X. Huang, A. Mallya, T. Karras, T. Aila, J. Lehtinen, and J. Kautz. Few-shot unsupervised image-to-image translation. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019. (Cited on page 206.)

[203] S. Liu, H. Wang, J. Ren, K. Luo, and J. Fan. High-resolution reconstruction and a-priori modeling of turbulent flames in the context of large eddy simulation using the convolutional neural network. In *Proc. Combust. Inst.*, 2022. (Cited on pages 56, 57 and 59.)

[204] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 3431–3440, 2015. (Cited on page 41.)

[205] I. Loschchilov and F. Hutter. SGDR : Stochastic Gradient Descent with Warm Restarts. In *Int. Conf. Learn. Represent. (ICLR)*, 2017. (Cited on page 84.)

[206] Y. Lu, A. Zhong, Q. Li, and B. Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *Int. Conf. Mach. Learn. (ICML)*, 2018. (Cited on page 55.)

[207] S. Luca, A. N. Al-Khateeb, A. Attili, and F. Bisetti. Comprehensive Validation of Skeletal Mechanism for Turbulent Premixed Methane–Air Flame Simulations. *J. Propuls. Power*, 34(1):153–160, 2017. (Cited on pages 78 and 124.)

[208] S. Luca, A. Attili, and F. Bisetti. Direct Numerical Simulation of Turbulent Lean Methane-Air Bunsen Flames with Mixture Inhomogeneities. In *54th AIAA Aerosp. Sci. Meet.*, page 0189, 2016. (Cited on pages 18 and 125.)

[209] 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. In *Proc. Combust. Inst.*, volume 37, pages 2451–2459, 2019. (Cited on pages 3, 58, 78, 79 and 125.)

[210] J. L. Lumley. *Stochastic tools in turbulence*. Academic Press, 1970. (Cited on page 54.)

[211] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 30, 2017. (Cited on page 57.)

[212] P. Luo, X. Wang, W. Shao, and Z. Peng. Towards understanding regularization in batch normalization. In *Int. Conf. Learn. Represent. (ICLR)*, 2019. (Cited on page 48.)

[213] W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the Effective Receptive Field in Deep Convolutional Neural Networks. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 29, 2016. (Cited on pages 100 and 101.)

[214] T. Ma, O. T. Stein, N. Chakraborty, and A. M. Kempf. A posteriori testing of algebraic flame surface density models for LES. *Combust. Theory Model.*, 17(3):431–482, 2013. (Cited on page 31.)

[215] J. F. Macart, T. Grenga, and M. E. Mueller. Evolution of flame-conditioned velocity statistics in turbulent premixed jet flames at low and high Karlovitz numbers. In *Proc. Combust. Inst.*, volume 37, pages 2503–2510, 2019. (Cited on page 197.)

[216] A. R. Masri, A. Alharbi, S. Meares, and S. S. Ibrahim. A comparative study of turbulent premixed flames propagating past repeated obstacles. *Ind. Eng. Chem. Res.*, 51(22):7690–7703, 2012. (Cited on pages 157 and 158.)

[217] A. R. Masri, S. S. Ibrahim, N. Nehzat, and A. R. Green. Experimental study of premixed flame propagation over various solid obstructions. *Exp. Therm. Fluid Sci.*, 21(1-3):109–116, 2000. (Cited on page 157.)

[218] R. Maulik, K. Fukami, N. Ramachandra, K. Fukagata, and K. Taira. Probabilistic neural networks for fluid flow surrogate modeling and data recovery. *Phys. Rev. Fluids*, 5(10):104401, 2020. (Cited on page 54.)

[219] R. Maulik, B. Lusch, and P. Balaprakash. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Phys. Fluids*, 33(3):037106, 2021. (Cited on page 54.)

[220] R. Maulik, O. San, J. D. Jacob, and C. Crick. Sub-grid scale model classification and blending through deep learning. *J. Fluid Mech.*, 870:784–812, 2019. (Cited on page 55.)

[221] R. Maulik, O. San, A. Rasheed, and P. Vedula. Data-driven deconvolution for large eddy simulations of Kraichnan turbulence. *Phys. Fluids*, 30(12), 2018. (Cited on page 55.)

[222] R. Maulik, O. San, A. Rasheed, and P. Vedula. Subgrid modelling for two-dimensional turbulence using neural networks. *J. Fluid Mech.*, 858:122–144, 2019. (Cited on page 55.)

[223] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.*, 5(4):115–133, 1943. (Cited on page 43.)

[224] R. Mercier, P. Auzillon, V. Moureau, N. Darabiha, O. Gicquel, D. Veynante, and B. Fiorina. Les modeling of the impact of heat losses and differential diffusion on turbulent stratified flame propagation: Application to the tu darmstadt stratified flame. *Flow, Turb. Combust.*, 93(2):349–381, 2014. (Cited on page 2.)

[225] R. Mercier, C. Mehl, B. Fiorina, and V. Moureau. Filtered Wrinkled Flamelets model for Large-Eddy Simulation of turbulent premixed combustion. *Combust. Flame*, 205:93–108, 2019. (Cited on page 29.)

[226] R. Mercier, T. Schmitt, D. Veynante, and B. Fiorina. The influence of combustion SGS submodels on the resolved flame propagation. Application to the LES of the Cambridge stratified flames. In *Proc. Combust. Inst.*, volume 35, pages 1259–1267, 2015. (Cited on page 36.)

[227] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, and A. Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020. (Cited on page 4.)

[228] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Comm. ACM*, 65(1):99–106, 2021. (Cited on page 4.)

[229] H. Mirgolbabaei and T. Echekki. A novel principal component analysis-based acceleration scheme for les–odt: An a priori study. *Combust. Flame*, 160(5):898–908, 2013. (Cited on page 56.)

[230] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. (Cited on page 41.)

[231] S. Mohammadnejad, Q. An, P. Vena, S. Yun, and S. Kheirkhah. Thick reaction zones in non-flamelet turbulent premixed combustion. *Combust. Flame*, 222:285–304, 2020. (Cited on page 24.)

[232] A. Mohan, D. Daniel, M. Chertkov, and D. Livescu. Compressed Convolutional LSTM: An Efficient Deep Learning framework to Model High Fidelity 3D Turbulence. *arXiv preprint arXiv:1903.00033*, 2019. (Cited on page 54.)

[233] A. T. Mohan, N. Lubbers, D. Livescu, and M. Chertkov. Embedding hard physical constraints in convolutional neural networks for 3d turbulence. In *Work. Integr. Deep. Neural Model. Differ. Equ. (ICLR)*, 2020. (Cited on page 56.)

[234] M. Morimoto, K. Fukami, K. Zhang, A. G. Nair, and K. Fukagata. Convolutional neural networks for fluid flow analysis: toward effective metamodeling and low-dimensionalization. *Theor. Comput. Fluid Dyn.*, 35(5):1–18, 2021. (Cited on page 50.)

[235] V. Moureau. *Simulation aux grandes échelles de l'aérodynamique interne des moteurs à piston*. PhD thesis, 2004. (Cited on page 120.)

[236] V. Moureau, P. Domingo, and L. Vervisch. From Large-Eddy Simulation to Direct Numerical Simulation of a lean premixed swirl flame: Filtered laminar flame-PDF modeling. *Combust. Flame*, 158(7):1340–1357, 2011. (Cited on page 38.)

[237] V. Moureau, B. Fiorina, and H. Pitsch. A level set formulation for premixed combustion LES considering the turbulent flame structure. *Combust. Flame*, 156(4):801–812, 2009. (Cited on page 28.)

[238] S. Mouriaux. *Large eddy simulation of the turbulent spark ignition and flame propagation in spark ignition engines*. PhD thesis, Université Paris-Saclay, 2016. (Cited on pages 35, 159 and 175.)

[239] S. Mouriaux, O. Colin, and D. Veynante. Adaptation of a dynamic wrinkling model to an engine configuration. In *Proc. Combust. Inst.*, volume 36, pages 3415–3422, 2017. (Cited on pages 30, 35, 36, 68, 77, 101, 166 and 167.)

[240] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 7074–7082, 2017. (Cited on page 41.)

[241] S. Mukhopadhyay, J. A. Van Oijen, and L. P. De Goey. A comparative study of presumed PDFs for premixed turbulent combustion modeling based on progress variable and its variance. *Fuel*, 159:728–740, 2015. (Cited on pages 116 and 117.)

[242] P. Mullen, A. McKenzie, Y. Tong, and M. Desbrun. A variational approach to eulerian geometry processing. *ACM Trans. Graph.*, 26(3):66–es, 2007. (Cited on page 30.)

[243] T. Murata, K. Fukami, and K. Fukagata. Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. *J. Fluid Mech.*, 882:A13, 2020. (Cited on page 54.)

[244] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Int. Conf. Mach. Learn. (ICML)*, 2010. (Cited on page 44.)

[245] T. Nakamura, K. Fukami, K. Hasegawa, Y. Nabae, and K. Fukagata. Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Phys. Fluids*, 33(2):025116, 2021. (Cited on page 54.)

[246] R. Nakazawa, Y. Minamoto, N. Inoue, and M. Tanahashi. Species reaction rate modelling based on physics-guided machine learning. *Combust. Flame*, 235:111696, 2022. (Cited on pages 57 and 59.)

[247] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever. Deep double descent: Where bigger models and more data hurt. *J. Stat. Mech.: Theory Exp.*, 2021(12):124003, 2021. (Cited on page 45.)

[248] F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity. *Flow, Turbul. Combust.*, 62:183–200, 1999. (Cited on page 36.)

[249] F. Nicoud, H. B. Toda, O. Cabrit, S. Bose, and J. Lee. Using singular values to build a subgrid-scale model for large eddy simulations. *Phys. Fluids*, 23(8), 2011. (Cited on page 37.)

[250] 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. (Cited on pages 56, 57, 101 and 125.)

[251] T. Nilsson, I. Langella, N. A. K. Doan, N. Swaminathan, R. Yu, and X. S. Bai. A priori analysis of sub-grid variance of a reactive scalar using DNS data of high Ka flames. *Combust. Theory Model.*, 23(5):885–906, 2019. (Cited on page 117.)

[252] G. Nivarti and S. Cant. Direct numerical simulation of the bending effect in turbulent premixed flames. In *Proc. Combust. Inst.*, volume 36, pages 1903–1910, 2017. (Cited on page 24.)

[253] G. Novati, H. L. de Laroussilhe, and P. Koumoutsakos. Automating turbulence modelling by multi-agent reinforcement learning. *Nat. Mach. Intell.*, 3(1):87–96, 2021. (Cited on page 55.)

[254] N. A. Okong'o and J. Bellan. Consistent large-eddy simulation of a temporal mixing layer laden with evaporating drops. Part 1. Direct numerical simulation, formulation and a priori analysis. *J. Fluid Mech.*, 499:1–47, 2004. (Cited on page 26.)

[255] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization. (Cited on pages 49, 100 and 102.)

[256] I. P. on Climate Change (IPCC). Summary for Policymakers. In *Clim. Chang. 2021: Phys. Sci. Basis. Contrib. Work. Group I Sixth Assess. Rep. Intergov. Panel Clim. Chang.*, pages 3–32. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2021. (Cited on pages 1 and 2.)

[257] O. Owoyele, P. Kundu, M. M. Ameen, T. Echekki, and S. Som. Application of deep artificial neural networks to multi-dimensional flamelet libraries and spray flames. *Int. J. Engine Res.*, 21(1):151–168, 2020. (Cited on page 56.)

[258] L. Paris. Comportement des structures soumises à une explosion. *Tech. Ing.*, 2010. (Cited on page 157.)

[259] J. Park and H. Choi. Toward neural-network-based large eddy simulation: Application to turbulent channel flow. *J. Fluid Mech.*, 914:A16, 2021. (Cited on page 55.)

[260] T. Passot and A. Pouquet. Numerical Simulation of Compressible Homogeneous Flows in the Turbulent Regime. *J. Fluid Mech.*, 181:441–466, 1987. (Cited on page 97.)

[261] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 32, 2019. (Cited on page 43.)

[262] C. Pérez Arroyo, J. Dombard, F. Duchaine, L. Gicquel, B. Martin, N. Odier, and G. Staffelbach. Towards the Large-Eddy Simulation of a full engine: Integration of a 360 azimuthal degrees fan, compressor and combustion chamber. Part I: Methodology and initialisation. *J. Glob. Power Propuls. Soc.*, (Special issue):1–16, 2021. (Cited on page 24.)

[263] C. Pérez Arroyo, J. Dombard, F. Duchaine, L. Gicquel, B. Martin, N. Odier, and G. Staffelbach. Towards the Large-Eddy Simulation of a full engine: Integration of a 360 azimuthal degrees fan, compressor and combustion chamber. Part II: Comparison against stand-alone simulations. *J. Glob. Power Propuls. Soc.*, (Special issue):1–16, 2021. (Cited on page 24.)

[264] B. A. Perry, M. T. H. de Frahan, and S. Yellapantula. Co-optimized machine-learned manifold models for large eddy simulation of turbulent combustion. *Combust. Flame*, 244:112286, 2022. (Cited on page 56.)

[265] N. Peters. Laminar flamelet concepts in turbulent combustion. In *Symp. Combust.*, volume 21, pages 1231–1250, 1988. (Cited on pages 21, 23 and 125.)

[266] N. Peters. The turbulent burning velocity for large-scale and small-scale turbulence. *J. Fluid Mech.*, 384:107–132, 1999. (Cited on page 125.)

[267] N. Peters. *Turbulent Combustion*. Cambridge University Press, 2000. (Cited on pages 21, 22 and 32.)

[268] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia. Learning Mesh-Based Simulation with Graph Networks. In *Int. Conf. Learn. Represent. (ICLR)*, 2021. (Cited on pages 154 and 205.)

[269] M. Pfitzner. A New Analytic pdf for Simulations of Premixed Turbulent Combustion. *Flow, Turbul. Combust.*, 106(4):1213–1239, 2021. (Cited on pages 17, 115 and 138.)

[270] 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. (Cited on pages 17, 68 and 138.)

[271] C. D. Pierce and P. Moin. A dynamic model for subgrid-scale variance and dissipation rate of a conserved scalar. *Phys. Fluids*, 10(12):3041–3044, 1998. (Cited on page 116.)

[272] U. Piomelli. Large-eddy simulation: achievements and challenges. *Prog. Aerosp. Sci.*, 35(4):335–362, 1999. (Cited on page 26.)

[273] H. Pitsch. A consistent level set formulation for large-eddy simulation of premixed turbulent combustion. *Combust. Flame*, 143(4):587–598, 2005. (Cited on page 28.)

[274] H. Pitsch. Large eddy simulation of turbulent combustion. *Annu. Rev. Fluid Mech.*, 38:453–482, 2006. (Cited on page 115.)

[275] T. Poinsot and D. Veynante. *Theoretical and Numerical Combustion*. R.T. Edwards, 2005. (Cited on pages 11, 12, 13, 14, 15, 16, 17, 125 and 212.)

[276] T. Poinsot, D. Veynante, and S. Candel. Diagrams of premixed turbulent combustion based on direct simulation. In *Symp. Combust.*, volume 23, pages 613–619, 1991. (Cited on page 21.)

[277] J. Poinsot, Thierry and S. K. Lele. Boundary conditions for direct simulations of compressible viscous flows. *J. Comput. Phys.*, 101(1):104–129, 1992. (Cited on pages 36, 97 and 118.)

[278] S. Pope. The evolution of surfaces in turbulence. *Int. J. Eng. Sci.*, 26(5):445–469, 1988. (Cited on page 29.)

[279] S. Pope. Computations of turbulent combustion: Progress and challenges. In *Symp. Combust.*, volume 23, pages 591–612, 1991. (Cited on pages 28 and 115.)

[280] S. B. Pope. PDF methods for turbulent reactive flows. *Prog. Energy Combust. Sci.*, 11(2):119–192, 1985. (Cited on page 28.)

[281] S. B. Pope. Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation. *Combust. Theory Model.*, 1(1):41–63, 1997. (Cited on page 115.)

[282] S. B. Pope. *Turbulent flows*. Cambridge University Press, 2000. (Cited on pages 18, 19, 20, 24, 34 and 196.)

[283] G. D. Portwood, B. T. Nadiga, J. A. Saenz, and D. Livescu. Interpreting neural network models of residual scalar flux. *J. Fluid Mech.*, 907:A23, 2021. (Cited on page 55.)

[284] A. Prakash, K. E. Jansen, and J. A. Evans. Invariant data-driven subgrid stress modeling in the strain-rate eigenframe for large eddy simulation. *Comput. Methods Appl. Mech. Eng.*, 399:115457, 2022. (Cited on pages 56 and 205.)

[285] F. Proch, P. Domingo, L. Vervisch, and A. M. Kempf. Flame resolved simulation of a turbulent premixed bluff-body burner experiment. Part I: Analysis of the reaction zone dynamics with tabulated chemistry. *Combust. Flame*, 180:321–339, 2017. (Cited on page 36.)

[286] F. Proch and A. M. Kempf. Numerical analysis of the Cambridge stratified flame series using artificial thickened flame LES with tabulated premixed flame chemistry. *Combust. Flame*, 161(10):2627–2646, 2014. (Cited on page 115.)

[287] S. Puggelli, D. Veynante, and R. Vicquelin. Impact of dynamic modelling of the flame subgrid scale wrinkling in large-Eddy simulation of light-round in an annular combustor. *Combust. Flame*, 230:111416, 2021. (Cited on pages 2, 30, 36 and 68.)

[288] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017. (Cited on page 205.)

[289] C. Qin, K. D. Dvijotham, B. O'Donoghue, R. Bunel, R. Stanforth, S. Gowal, J. Uesato, G. Swirszcz, and P. Kohli. Verification of non-linear specifications for neural networks. In *Int. Conf. Learn. Represent. (ICLR)*, 2019. (Cited on page 56.)

[290] P. Quillatre. *Simulation Aux Grandes Echelles d'Explosions en Domaine Semi-Confiné*. PhD thesis, INP Toulouse, 2014. (Cited on pages 18, 35, 157, 163, 164 and 170.)

[291] P. Quillatre, O. Vermorel, T. Poinsot, and P. Ricoux. Large eddy simulation of vented deflagration. *Ind. Eng. Chem. Res.*, 52(33):11414–11423, 2013. (Cited on pages 118, 138, 159, 171 and 184.)

[292] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *Int. Conf. Mach. Learn. (ICML)*, 2021. (Cited on page 4.)

[293] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. (Cited on page 4.)

[294] M. Raissi, H. Babaee, and P. Givi. Deep learning of turbulent scalar mixing. *Phys. Rev. Fluids*, 4(12):124501, 2019. (Cited on page 55.)

[295] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019. (Cited on pages 54 and 56.)

[296] M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis. Deep learning of vortex-induced vibrations. *J. Fluid Mech.*, 861:119–137, 2019. (Cited on page 55.)

[297] P. Ralon, M. Taylor, A. Ilas, H. Diaz-Bone, and K.-P. Kairies. Electricity storage and renewables: Costs and markets to 2030. Technical report, International Renewable Energy Agency (IRENA), 2017. (Cited on page 2.)

[298] V. Raman and M. Hassanaly. Emerging trends in numerical simulations of combustion systems. In *Proc. Combust. Inst.*, volume 37, pages 2073–2089, 2019. (Cited on page 3.)

[299] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. (Cited on pages 4 and 43.)

[300] R. Ranjan, B. Muralidharan, Y. Nagaoka, and S. Menon. Subgrid-Scale Modeling of Reaction-Diffusion and Scalar Transport in Turbulent Premixed Flames. *Combust. Sci. Technol.*, 188(9):1496–1537, 2016. (Cited on pages 116 and 127.)

[301] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, S. Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021. (Cited on page 5.)

[302] S. P. Reddy Muppala, N. K. Aluri, F. Dinkelacker, and A. Leipertz. Development of an algebraic reaction rate closure for the numerical calculation of turbulent premixed methane, ethylene, and propane/air flames for pressures up to 1.0 MPa. *Combust. Flame*, 140(4):257–266, 2005. (Cited on page 31.)

[303] A. Refloch, B. Courbet, A. Murrone, P. Villedieu, C. Laurent, P. Gilbank, J. Troyes, L. Tessé, G. Chaineray, J. Dargaud, et al. Cedre software. *Aerospace Lab*, 2, 2011. (Cited on page 142.)

[304] J. Ren, H. Wang, G. Chen, K. Luo, and J. Fan. Predictive models for flame evolution using machine learning: A priori assessment in turbulent flames without and with mean shear. *Phys. Fluids*, 33(5), 2021. (Cited on page 54.)

[305] 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. (Cited on pages 57, 59, 101 and 125.)

[306] Y. Ren, C. Hu, X. Tan, T. Qin, S. Zhao, Z. Zhao, and T.-Y. Liu. Fastspeech 2: Fast and high-quality end-to-end text to speech. In *Int. Conf. Learn. Represent. (ICLR)*, 2021. (Cited on page 4.)

[307] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti. Detecting novel associations in large data sets. *Science*, 334:1518–1524, 2011. (Cited on page 57.)

[308] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Int. Conf. Mach. Learn. (ICML)*, pages 1530–1538, 2015. (Cited on page 41.)

[309] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Int. Conf. Mach. Learn. (ICML)*, 2014. (Cited on page 41.)

[310] M. Rieth, A. Gruber, F. A. Williams, and J. H. Chen. Enhanced burning rates in hydrogen-enriched turbulent premixed flames by diffusion of molecular and atomic hydrogen. *Combust. Flame*, 239:111740, 2021. (Cited on page 24.)

[311] H. Ritchie, M. Roser, and P. Rosado. Co2 and greenhouse gas emissions. *Our World in Data*, 2020. https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions. (Cited on page 1.)

[312] H. Ritchie, M. Roser, and P. Rosado. Energy. *Our World in Data*, 2020. https://ourworldindata.org/energy. (Cited on page 3.)

[313] R. S. Rogallo and P. Moin. Numerical simulation of turbulent flow. *Annu. Rev. Fluid Mech.*, 16(1):99–137, 1984. (Cited on pages 34, 66 and 126.)

[314] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022. (Cited on page 4.)

[315] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Med. Image Comput. Comput. Interv. – MICCAI 2015*, volume 9351, pages 234–241, 2015. (Cited on page 49.)

[316] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. (Cited on page 43.)

[317] V. A. Sabelnikov, A. N. Lipatnikov, N. Nikitin, S. Nishiki, and T. Hasegawa. Application of Helmholtz-Hodge decomposition and conditioned structure functions to exploring influence of premixed combustion on turbulence upstream of the flame. In *Proc. Combust. Inst.*, volume 38, pages 3077–3085, 2021. (Cited on page 197.)

[318] V. A. Sabelnikov, A. N. Lipatnikov, S. Nishiki, H. L. Dave, F. E. Hernández Pérez, W. Song, and H. G. Im. Dissipation and dilatation rates in premixed turbulent flames. *Phys. Fluids*, 33(3), 2021. (Cited on page 122.)

[319] P. Sagaut. *Large eddy simulation for incompressible flows: an introduction*. Springer Science & Business Media, 2006. (Cited on pages 25 and 60.)

[320] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. (Cited on page 4.)

[321] A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, and P. Battaglia. Hamiltonian Graph Networks with ODE Integrators. In *Work. Mach. Learn. Phys. Sci. (NeurIPS)*, 2019. (Cited on page 154.)

[322] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry. How does batch normalization help optimization? In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 31, 2018. (Cited on page 48.)

[323] D. H. Sattinger. On the stability of waves of nonlinear parabolic systems. *Adv. Math.*, 22(3):312–355, 1976. (Cited on pages 211 and 212.)

[324] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.*, 656:5–28, 2010. (Cited on page 54.)

[325] R. M. Schmidt, F. Schneider, and P. Hennig. Descending through a Crowded Valley – Benchmarking Deep Learning Optimizers. In *Int. Conf. Mach. Learn. (ICML)*, 2021. (Cited on page 44.)

[326] T. Schmitt, M. Boileau, and D. Veynante. Flame wrinkling factor dynamic modeling for large eddy simulations of turbulent premixed combustion. *Flow, Turbul. Combust.*, 94(1):199–217, 2015. (Cited on page 36.)

[327] T. Schönfeld and M. Rudgyard. Steady and unsteady flow simulations using the hybrid flow solver AVBP. *AIAA J.*, 37(11):1378–1385, 1999. (Cited on pages 36 and 119.)

[328] L. Selle, G. Lartigue, T. Poinsot, R. Koch, K. U. Schildmacher, W. Krebs, B. Prade, P. Kaufmann, and D. Veynante. Compressible large eddy simulation of turbulent combustion in complex geometry on unstructured meshes. *Combust. Flame*, 137(4):489–505, 2004. (Cited on pages 36, 119 and 205.)

[329] 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. (Cited on pages 57, 59 and 101.)

[330] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In

Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2017. (Cited on page 100.)

[331] B. A. Sen and S. Menon. Linear eddy mixing based tabulation and artificial neural networks for large eddy simulations of turbulent flames. *Combust. Flame*, 157(1):62–74, 2010. (Cited on page 56.)

[332] S. Seo, C. Meng, and Y. Liu. Physics-aware Difference Graph Networks for Sparsely-Observed Dynamics. In *Int. Conf. Learn. Represent. (ICLR)*, 2020. (Cited on page 154.)

[333] C. J. Shallue, J. Lee, J. Antognini, J. Sohl-Dickstein, R. Frostig, and G. E. Dahl. Measuring the effects of data parallelism on neural network training. *J. Mach. Learn. Res.*, 20(112):1–49, 2019. (Cited on page 83.)

[334] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, pages 4779–4783, 2018. (Cited on page 4.)

[335] J. Shin, Y. Ge, A. Lampmann, and M. Pfitzner. A Data-Driven Subgrid Scale Model in Large Eddy Simulation of Turbulent Premixed Combustion. (Cited on page 204.)

[336] 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. (Cited on pages 57, 59 and 101.)

[337] J. Shin, M. Hansinger, M. Pfitzner, and M. Klein. A priori analysis on deep learning of filtered reaction rate. *Flow, Turbul. Combust.*, 109:383–409, 2022. (Cited on pages 57, 59 and 204.)

[338] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–489, 2016. (Cited on page 41.)

[339] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362:1140–1144, 2018. (Cited on page 41.)

[340] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Int. Conf. Learn. Represent. (ICLR) - Work. Track Proc.*, 2014. (Cited on page 100.)

[341] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Int. Conf. Learn. Represent. (ICLR)*, 2015. (Cited on page 50.)

[342] J. Sirignano and K. Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *J. Comput. Phys.*, 375:1339–1364, 2018. (Cited on page 54.)

[343] A. W. Skiba, C. D. Carter, S. D. Hammack, and J. F. Driscoll. Experimental assessment of the progress variable space structure of premixed flames subjected to extreme turbulence. In *Proc. Combust. Inst.*, volume 38, pages 2893–2900, 2021. (Cited on pages 32 and 115.)

[344] A. W. Skiba, C. D. Carter, S. D. Hammack, and J. F. Driscoll. High-fidelity flame-front wrinkling measurements derived from fractal analysis of turbulent premixed flames with large Reynolds numbers. In *Proc. Combust. Inst.*, volume 38, pages 2809–2816, 2021. (Cited on pages 32, 33 and 34.)

[345] A. W. Skiba, C. D. Carter, S. D. Hammack, and J. F. Driscoll. Premixed flames subjected to extreme levels of turbulence part II: Surface characteristics and scalar dissipation rates. *Combust. Flame*, 239(239):111703, 2022. (Cited on page 115.)

[346] A. W. Skiba, T. M. Wabel, C. D. Carter, S. D. Hammack, J. E. Temme, and J. F. Driscoll. Premixed flames subjected to extreme levels of turbulence part I: Flame structure and a new measured regime diagram. *Combust. Flame*, 189:407–432, 2018. (Cited on pages 23 and 24.)

[347] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. In *Int. Conf. Learn. Represent. (ICLR)*, 2021. (Cited on page 41.)

[348] K. Sreenivasan and C. Meneveau. The fractal facets of turbulence. *J. Fluid Mech.*, 173:357–386, 1986. (Cited on page 199.)

[349] P. A. Srinivasan, L. Guastoni, H. Azizpour, P. Schlatter, and R. Vinuesa. Predictions of turbulent shear flows using deep neural networks. *Phys. Rev. Fluids*, 4(5):054603, 2019. (Cited on page 54.)

[350] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014. (Cited on page 45.)

[351] K. Stachenfeld, D. B. Fielding, D. Kochkov, M. Cranmer, T. Pfaff, J. Godwin, C. Cui, S. Ho, P. Battaglia, and A. Sanchez-Gonzalez. Learned coarse models for efficient turbulence simulation. In *Int. Conf. Learn. Represent. (ICLR)*, 2022. (Cited on page 54.)

[352] I. Staffell, D. Scamman, A. V. Abad, P. Balcombe, P. E. Dodds, P. Ekins, N. Shah, and K. R. Ward. The role of hydrogen and fuel cells in the global energy system. *Energy Environ. Sci*, 12(2):463–491, 2019. (Cited on page 3.)

[353] A. N. Stokes. Nonlinear diffusion waveshapes generated by possibly finite initial disturbances. *J. Math. Anal. Appl.*, 61(2):370–381, 1977. (Cited on pages 211 and 212.)

[354] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Int. Conf. Mach. Learn. (ICML)*, 2017. (Cited on page 57.)

[355] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018. (Cited on page 41.)

[356] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015. (Cited on pages 4, 49 and 50.)

[357] 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. (Cited on pages 57 and 59.)

[358] J. Tarbouriech. *Goal-Oriented Exploration for Reinforcement Learning*. PhD thesis, 2022. (Cited on page 41.)

[359] N. Thomas, T. Smidt, S. Kearnes, L. Yang, L. Li, K. Kohlhoff, and P. Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018. (Cited on page 205.)

[360] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin. Accelerating eulerian fluid simulation with convolutional networks. In *Int. Conf. Mach. Learn. (ICML)*, 2017. (Cited on page 55.)

[361] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In *Int. Conf. Mach. Learn. (ICML)*, 2021. (Cited on page 41.)

[362] H. Touvron, A. Vedaldi, M. Douze, and H. Jégou. Fixing the train-test resolution discrepancy. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 32, 2019. (Cited on page 69.)

[363] A. Trouvé and T. Poinsot. The evolution equation for the flame surface density in turbulent premixed combustion. *J. Fluid Mech.*, 278:1–31, 1994. (Cited on page 29.)

[364] J. W. Tukey. *Exploratory data analysis*. Addison-Wesley, 1977. (Cited on page 150.)

[365] A. van den Oord, Y. Li, and O. Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv preprint arXiv:1807.03748*, 2018. (Cited on page 41.)

[366] J. A. van Oijen, A. Donini, R. J. Bastiaans, J. H. ten Thije Boonkkamp, and L. P. de Goey. State-of-the-art in premixed combustion modeling using flamelet generated manifolds. *Prog. Energy Combust. Sci.*, 57:30–74, 2016. (Cited on page 115.)

[367] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. In *Int. Conf. Mach. Learn. (ICML)*, 2016. (Cited on page 41.)

[368] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 30, 2017. (Cited on page 48.)

[369] O. Vermorel, P. Quillatre, and T. Poinsot. LES of explosions in venting chamber: A test case for premixed turbulent combustion models. *Combust. Flame*, 183:207–223, 2017. (Cited on pages 24, 30, 138, 158, 159, 160, 161, 170, 171 and 184.)

[370] D. Veynante and R. Knikker. Comparison between LES results and experimental data in reacting flows. *J. Turbul.*, 7:N35, 2006. (Cited on pages 116 and 127.)

[371] D. Veynante and V. Moureau. Analysis of dynamic models for large eddy simulations of turbulent premixed combustion. *Combust. Flame*, 162(12):4622–4642, 2015. (Cited on pages 34, 36, 101 and 197.)

[372] D. Veynante, A. Trouvé, K. Bray, and T. Mantel. Gradient and counter-gradient scalar transport in turbulent premixed flames. *J. Fluid Mech.*, 332:263–293, 1997. (Cited on page 27.)

[373] R. Vinuesa and S. L. Brunton. The potential of machine learning to enhance computational fluid dynamics. *arXiv preprint arXiv:2110.02085*, 2021. (Cited on page 54.)

[374] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575:350–354, 2019. (Cited on page 41.)

[375] A. Vollant, G. Balarac, and C. Corre. Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures. *J. Turbul.*, 18(9):854–878, 2017. (Cited on page 56.)

[376] P. S. Volpiani, T. Schmitt, O. Vermorel, P. Quillatre, and D. Veynante. Large eddy simulation of explosion deflagrating flames using a dynamic wrinkling formulation. *Combust. Flame*, 186:17–31, 2017. (Cited on pages 36, 158, 159 and 161.)

[377] P. S. Volpiani, T. Schmitt, and D. Veynante. A posteriori tests of a dynamic thickened flame model for large eddy simulations of turbulent premixed combustion. *Combust. Flame*, 174:166–178, 2016. (Cited on pages 36, 38, 68, 101 and 149.)

[378] P. S. Volpiani, T. Schmitt, and D. Veynante. Large eddy simulation of a turbulent swirling premixed flame coupling the TFLES model with a dynamic wrinkling formulation. *Combust. Flame*, 180:124–135, 2017. (Cited on pages 30 and 36.)

[379] A. W. Vreman, R. J. Bastiaans, and B. J. Geurts. A similarity subgrid model for premixed turbulent combustion. *Flow, Turbul. Combust.*, 82(2):233–248, 2009. (Cited on page 116.)

[380] B. Vreman, B. Geurts, and H. Kuerten. A priori tests of large eddy simulation of the compressible plane mixing layer. *J. Eng. Math.*, 29(4):299–327, 1995. (Cited on page 26.)

[381] C. Wall, B. J. Boersma, and P. Moin. An evaluation of the assumed beta probability density function subgrid-scale model for large eddy simulation of nonpremixed, turbulent combustion with heat release. *Phys. Fluids*, 12(10):2522–2529, 2000. (Cited on page 116.)

[382] 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. (Cited on pages 56, 101 and 125.)

[383] K. Wan, S. Hartl, L. Vervisch, P. Domingo, R. S. Barlow, and C. Hasse. Combustion regime identification from machine learning trained by Raman/Rayleigh line measurements. *Combust. Flame*, 219:268–274, 2020. (Cited on page 205.)

[384] N. Wandel, M. Weinmann, and R. Klein. Learning Incompressible Fluid Dynamics from Scratch - Towards Fast, Differentiable Fluid Models that Generalize. In *Int. Conf. Learn. Represent. (ICLR)*, 2021. (Cited on page 54.)

[385] G. Wang, M. Boileau, and D. Veynante. Implementation of a dynamic thickened flame model for large eddy simulations of turbulent premixed combustion. *Combust. Flame*, 158(11):2199–2213, 2011. (Cited on pages 31, 33, 36 and 77.)

[386] G. Wang, M. Boileau, D. Veynante, and K. Truffin. Large eddy simulation of a growing turbulent premixed flame kernel using a dynamic flame surface density model. *Combust. Flame*, 159(8):2742–2754, 2012. (Cited on page 36.)

[387] R. Wang, K. Kashinath, M. Mustafa, A. Albert, and R. Yu. Towards Physics-informed Deep Learning for Turbulent Flow Prediction. In *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pages 1457–1466, 2020. (Cited on pages 54 and 56.)

[388] R. Wang, R. Walters, and R. Yu. Incorporating Symmetry into Deep Dynamics Models for Improved Generalization. In *Int. Conf. Learn. Represent. (ICLR)*, 2021. (Cited on page 205.)

[389] S. Wang, X. Yu, and P. Perdikaris. When and why pinns fail to train: A neural tangent kernel perspective. *J. Comput. Phys.*, 449:110768, 2022. (Cited on page 55.)

[390] Z. Wang, K. Luo, D. Li, J. Tan, and J. Fan. Investigations of data-driven closure for subgrid-scale stress in large-eddy simulation. *Phys. Fluids*, 30(12):125101, 2018. (Cited on page 55.)

[391] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. Cohen. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 31, 2018. (Cited on page 205.)

[392] H. G. Weller, G. Tabor, A. D. Gosman, and C. Fureby. Application of a flame-wrinkling les combustion model to a turbulent mixing layer. In *Symp. Combust.*, volume 27, pages 899–907, 1998. (Cited on page 31.)

[393] D. C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, Inc, third edition, 2006. (Cited on page 18.)

[394] G. Wilson and D. J. Cook. A Survey of Unsupervised Deep Domain Adaptation. *ACM Trans. Intell. Syst. Technol.*, 11(5), 2020. (Cited on page 206.)

[395] S. Wiseman, M. Rieth, A. Gruber, J. R. Dawson, and J. H. Chen. A comparison of the blow-out behavior of turbulent premixed ammonia/hydrogen/nitrogen-air and methane–air flames. In *Proc. Combust. Inst.*, volume 38, pages 2869–2876, 2021. (Cited on page 3.)

[396] D. E. Worrall and M. Welling. Deep scale-spaces: Equivariance over scale. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 32, 2019. (Cited on page 205.)

[397] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Networks Learn. Syst.*, 32(1):4–24, 2021. (Cited on page 154.)

[398] Z. Xiao, K. Kreis, and A. Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. In *Int. Conf. Learn. Represent. (ICLR)*, 2022. (Cited on page 41.)

[399] C. Xie, J. Wang, and E. Weinan. Modeling subgrid-scale forces by spatial artificial neural networks in large eddy simulation of turbulence. *Phys. Rev. Fluids*, 5(5):54606, 2020. (Cited on page 55.)

[400] 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. (Cited on page 114.)

[401] H. Yamashita, M. Shimada, and T. Takeno. A numerical study on flame stability at the transition point of jet diffusion flames. In *Symp. Combust.*, volume 26, pages 27–34, 1996. (Cited on page 205.)

[402] 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. (Cited on pages 57 and 59.)

[403] S. Yellapantula, B. A. Perry, and R. W. Grout. Deep learning-based model for progress variable dissipation rate in turbulent premixed flames. In *Proc. Combust. Inst.*, volume 38, pages 2929–2938, 2021. (Cited on pages 57, 59 and 125.)

[404] H. You, C. Li, P. Xu, Y. Fu, Y. Wang, X. Chen, R. G. Baraniuk, Z. Wang, and Y. Lin. Drawing early-bird tickets: Towards more efficient training of deep networks. In *Int. Conf. Learn. Represent. (ICLR)*, 2020. (Cited on page 102.)

[405] Y. You, Z. Zhang, C.-J. Hsieh, J. Demmel, and K. Keutzer. Imagenet training in minutes. In *Proc. Int. Conf. Parallel Process. (ICPP)*, pages 1–10, 2018. (Cited on page 83.)

[406] Z. Yuan, C. Xie, and J. Wang. Deconvolutional artificial neural network models for large eddy simulation of turbulence. *Phys. Fluids*, 32(11):115106, 2020. (Cited on page 55.)

[407] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *Int. Conf. Mach. Learn. (ICML)*, 2021. (Cited on pages 41 and 206.)

[408] M. D. Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. In *Eur. Conf. Comput. Vis.*, pages 818–833, 2014. (Cited on page 100.)

[409] Y. Zeldovich. Flame propagation in a substance reacting at initial temperature. *Combust. Flame*, 39(3):219–224, 1980. (Cited on page 211.)

[410] C. Zhang, S. Bengio, M. Hardt, M. C. Mozer, and Y. Singer. Identity Crisis: Memorization and Generalization Under Extreme Overparameterization. In *Int. Conf. Learn. Represent. (ICLR)*, 2020. (Cited on page 48.)

[411] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115, 2021. (Cited on page 45.)

[412] T. Zhang, Y. Yi, Y. Xu, Z. X. Chen, Y. Zhang, E. Weinan, and Z.-Q. J. Xu. A multi-scale sampling method for accurate and robust deep neural network to predict combustion chemical kinetics. *Combust. Flame*, 245:112319, 2022. (Cited on page 56.)

[413] L. Zhao and L. Akoglu. PairNorm: Tackling Oversmoothing in GNNs. In *Int. Conf. Learn. Represent. (ICLR)*, 2020. (Cited on page 154.)

[414] H. Zhou, J. Lan, R. Liu, and J. Yosinski. Deconstructing Lottery Tickets: Zeros, Signs, and the Supermask. In *Adv. Neural Inf. Process. Syst. (NeurIPS)*, volume 32, 2019. (Cited on page 102.)

[415] Z. Zhou, G. He, S. Wang, and G. Jin. Subgrid-scale model for large-eddy simulation of isotropic turbulent flows using an artificial neural network. *Comput. Fluids*, 195:104319, 2019. (Cited on page 55.)

[416] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support*, pages 3–11, 2018. (Cited on pages 49 and 50.)

[417] Y. Zhu, N. Zabaras, P. S. Koutsourelakis, and P. Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *J. Comput. Phys.*, 394:56–81, 2019. (Cited on page 54.)