Data-Centric Engineering (2025), 6: e2
doi:10.1017/dce.2024.56

RESEARCH ARTICLE

Data-driven multifidelity surrogate models for rocket engines injector design

Jose Felix Zapata Usandivaras[1], Michael Bauerheim[2], Bénédicte Cuenot[2] and Annafederica Urbano[1]

[1]Fédération ENAC ISAE-SUPAERO ONERA, Université de Toulouse, Toulouse, France
[2]Centre Europén de Recherche et de Formation Avancé en Calcul Scientifique (CERFACS), Toulouse, France
Corresponding author: Jose Felix Zapata Usandivaras; Email: jose.zapata-usandivaras@isae-supaero.fr

Received: 07 May 2024; **Revised:** 05 November 2024; **Accepted:** 06 November 2024

Keywords: Rocket engine; shear-coaxial injector; surrogate modeling; transfer learning

Abstract

Surrogate models of turbulent diffusive flames could play a strategic role in the design of liquid rocket engine combustion chambers. The present article introduces a method to obtain data-driven surrogate models for coaxial injectors, by leveraging an inductive transfer learning strategy over a U-Net with available multifidelity Large Eddy Simulations (LES) data. The resulting models preserve reasonable accuracy while reducing the offline computational cost of data-generation. First, a database of about 100 low-fidelity LES simulations of shear-coaxial injectors, operating with gaseous oxygen and gaseous methane as propellants, has been created. The design of experiments explores three variables: the chamber radius, the recess-length of the oxidizer post, and the mixture ratio. Subsequently, U-Nets were trained upon this dataset to provide reasonable approximations of the temporal-averaged two-dimensional flow field. Despite the fact that neural networks are efficient non-linear data emulators, in purely data-driven approaches their quality is directly impacted by the precision of the data they are trained upon. Thus, a high-fidelity (HF) dataset has been created, made of about 10 simulations, to a much greater cost per sample. The amalgamation of low and HF data during the the transfer-learning process enables the improvement of the surrogate model's fidelity without excessive additional cost.

Impact statement

The present study showcases the novel use of inductive transfer learning for the generation of multifidelity surrogate models (MFSMs) of rocket engine coaxial injectors, combining large eddy simulations of different fidelities, while resulting in a net reduction of the offline cost of data generation. The MFSMs obtained are able to recover qualitatively the correct topology of the injectors' diffusion flame while preserving acceptable prediction accuracies over the dedicated test datasets.

1. Introduction

The space-sector has seen a wave of renewed interest primarily driven by private endeavors under the promise of revenue. However, a significant logistical challenge exists, mainly driven by the excessive cost-of-access to space, that is, launchers. The adequate design-optimization of these is one of the multiple avenues proposed to render space flight more affordable. Launch-vehicles can be interpreted as complex systems, composed of a multitude of subsystems, which combined yield the preliminary-design of these multidisciplinary problems. Note that in many cases, the links among the different subjects obey a strong non-linear coupling. In this context, the accuracy of the models of the subsystems becomes relevant, as well as their restitution times. In the case of the chemical rocket engine, the multiscale unsteady nature of combustion and turbulence at high strain rates makes this subsystem particularly complex to model with satisfactory accuracy.

The accuracy–cost balance is known to industrial as well as research actors. This has motivated the development of a community of applied science whose aim is to develop methodologies to obtain so-called *surrogate models* or *metamodels.* Such are elaborated on the basis of data coming from simulations or experiments and aspire to provide relatively fast approximations of target functions and constraints at a new design-point (Bouhlel et al., 2019; Chen et al., 2021; Hwang and Martins, 2018). The cost is typically orders of magnitude below the cost of a single high-fidelity (HF) numerical model survey.

In this study, the meta-objective has been to develop suitable data-driven surrogate models for a rocket engine shear-coaxial injector from data sourced from LES simulations. These models are meant to provide approximations of temporal-averaged two-dimensional (2D)-flow fields, namely: the temperature, oxygen mass-fraction, mixture-fraction, axial velocity, and axial velocity root mean squared value (RMS) fields. We concentrate on the analysis of a multifidelity model (MFM) strategy, achieved by inductive transfer learning, to reduce the *offline* data generation cost. In essence, this is achieved by staging the learning of the system under scrutiny over two datasets of different "quality," and consequently, cost. A brief introduction of the use of Deep Learning (DL) for surrogate modeling is given in Section 1.1. In what follows, the multifidelity strategy adopted is explained and an introduction to transfer learning is provided in Section 1.2.

1.1. DL for surrogate modeling

Historically, the multiple ways to build surrogate models have been split into three major groups (Eldred and Dunlavy, 2012; Hwang and Martins, 2018; Yu and Hesthaven, 2019): MFMs, reduced order models (ROMs) and data-fit models[1]. More recently, deep neural networks (DNNs) have gained popularity as a modeling technology applicable across all of these three categories because of their intrinsic ability to extract high-level representations from data. Moreover, it is for these reasons they have caught the attention of the fluid mechanics community for their potential in regression tasks in the face of limited data (Brunton et al., 2020; Mendez et al., 2023).

In the realm of DL, surrogate models may be present as physics-informed, purely data-driven, or a combination of both. The first and third categories involve those models in which inductive biases have been put in place to channel the learning within a constrained manifold, which typically alleviates data requirements (Vinuesa and Brunton, 2021). However, these models are not addressed in the reminder of this work, although the reader may find a comprehensive review of these methods in (Cai et al., 2021; Karniadakis et al., 2021). The second category of models is the reference avenue taken in the present work because of its inherent non-intrusiveness and the recent developments within the fluid mechanics field. Examples in the literature of recent years of DL-based surrogate models are abundant and constantly evolving at a fast pace. Some examples are to be found in: (Bermejo-Barbanoj et al., 2024; Farimani et al., 2017; Guo et al., 2016; Hasegawa et al., 2020; Thuerey et al., 2020; White et al., 2019; Zhang et al., 2018). Moreover, recent innovations in the realm of DL have seen successful applications in the surrogate modeling of fluid flows. Such is the case of neural operators such as Fourier Neural Operators (FNOs) (Z. Li et al., 2021), DeepONet (Goswami et al., 2022; Goswami et al., 2023; Lu et al., 2021), Latent Dynamic Networks (LDNets) (Regazzoni et al., 2024), Transformers (Hemmasian and Barati Farimani, 2023; Solera-Rico et al., 2024; Y. Wang et al., 2024; Zhao et al., 2024), and Mamba (Hu et al., 2024) (although only developed very recently for modeling of dynamical systems).

The field of turbulent combustion, for industrial applications has not been exempt from the advent of ML techniques (Le Clainche et al., 2023; Parente, 2024; Swaminathan and Parente, 2023). In this context,

[1] The exhaustive introduction of this taxonomy remains out of the scope of this text. However, shall the reader be interested, a more complete introduction is presented by Zapata Usandivaras et al [Zdybal et al., 2022]

ML methods have been used for a palette of problems (Parente, 2024). For instance, data analysis and feature extraction (Malik et al., 2021; Zdybał et al., 2022), where DL-based ROM methods are used primarily to reconstruct and explore a low-dimensional combustion manifold; adaptive simulation frameworks and subgrid models, where ML methods are used to resolve adequately the turbulent combustion model (Chung et al., 2021), replace combustion-tabulated approaches (Bhalla et al., 2019), or even model subgrid turbulent–chemistry interaction (Lapeyre et al., 2019; Xing et al., 2021). Furthermore, the direct use of ROMs of complex combustion systems has been leveraged for prediction for *digital twins* applications (Aversano et al., 2021; Aversano et al., 2021).

More concretely in the context of rocket engine coaxial injectors, Krügener et al. (Krügener et al., 2022) used a combination of Fully Connected Neural Networks (FCNN) and a variation of the aforementioned U-Net from Thuerey et al. (Thuerey et al., 2020) as surrogate models to predict sets of key-performance-indicators (KPIs), wall heat-flux and the temperature 2D-field on a single-element shear-coaxial injector rocket combustor. The models were trained on RANS data generated offline. Zapata Usandivaras et al. (Zdybal et al., 2022) followed through by developing similar data-driven models from an LES dataset of shear-coaxial injectors. Bear in mind that in the absence of any physics-informed inductive biases, the neural networks are bound to learn the structural relations of the data they are trained upon. Consequently, the surrogate models obtained from low-fidelity (LF) data will reproduce the inherent errors of low-resolution samples.

In this work, we intend to address this conundrum by adopting a *multifidelity* (MF) strategy, realized by adopting an inductive transfer learning technique. The idea of multifidelity surrogate models (MFSMs) is not new and abundant examples of these methods are found in the literature. The core concept behind these models is to leverage the data from LF samples, of easier access, and a limited pool of HF samples to estimate the HF function/task. Fernández-Godino et al. (Fernandez-Godino, 2023) conducted a comprehensive review on the state-of-the-art of MFMs. Some relevant works involving artificial neural networks (ANNs) include that of Liu and Wang (Liu and Wang, 2019), who proposed the use of a physics-constrained neural network (PCNN) trained on LF data, and later corrected by a second neural network. This framework was demonstrated in a 2D heat transfer and phase transition problem. Minisci and Vasile (Minisci and Vasile, 2013) made use of ANNs to correct aerodynamic forces obtained from an LFM using HF model CFD data. Their model is used within the framework of an optimization problem for the design of a reentry space vehicle.

To materialize the MFSMs in this work, two additional LES datasets are created. Contrary to the LF-LES mentioned in the previous paragraphs, these simulations constitute HF-LES of shear-coaxial injectors, operating on a gaseous oxygen (GOx) and gaseous methane (GCH_4) propellant couple. The number of HF samples is limited because of their high computational cost. The design space corresponds to the same one specified for the under-resolved LES (LF) dataset from Zapata Usandivaras et al (Zdybal et al., 2022). Finally, the application of transfer learning techniques, while making use of both the high and LF datasets, enables the elaboration of MF data-driven emulators with satisfactory performance on the HF test dataset.

1.2. Transfer learning

The notion of transfer learning spawned from the natural ability of humans to learn a new, unknown task leaning on the basis of knowledge acquired from learning a similar or related task (Ribani and Marengoni, 2019). The idea has been around since the 1990s, although it gained significant momentum in the 2010s with the appearance of publicly available datasets of labeled images such as ImageNet (Deng et al., 2009). Even pre-trained CNNs such as ResNet (He et al., 2016) are public and widely used because of their feature extraction capabilities, which makes them useful in image recognition and classification tasks. These have found multiple uses in the diagnosis of medical/biological images (Cheng, 2017; X. Li et al., 2017).

The transfer learning definition presented in (Pan and Yang, 2010) introduces two main concepts: domain and task. On the one hand, a domain $\mathcal{D}$ is related to the distribution of the training dataset. This training data represented by $\mathcal{D}$ resides in a *feature space* $\mathcal{X}$; however, it does so with a marginal probability distribution $P(X)$, where $X = [x_1, \ldots, x_n]$ is a particular learning sample expressed in features terms, such that x_i is the value of the ith feature. To consider two domains, $\mathcal{D}_A$ and $\mathcal{D}_B$, as different ($\mathcal{D}_A \neq \mathcal{D}_B$), either they both do not belong to the same feature space $\mathcal{X}$, or they have distinct probability distributions $P(X|\mathcal{D}_A) \neq P(X|\mathcal{D}_B)$.

On the other hand, a task $\mathcal{T} = \{\mathcal{D}, f(\cdot)\}$ is defined by a set of all possible labels $\mathcal{Y}$ and a predictor $f(\cdot)$ which estimates the label over a given domain $\mathcal{D}$. Note that a task can be learned, within a supervised learning case, from a collection of $\{x, y\}$ of pairs regarded as training data. Now, considering a source domain and task $\mathcal{D}_S$, $\mathcal{T}_S$ and a target domain and task $\mathcal{D}_T$, $\mathcal{T}_T$, *transfer learning* is defined as a process that will help improve $f_T(\cdot)$, given the knowledge obtained from $\mathcal{D}_S$ and $\mathcal{T}_S$, with $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$. An example of *transfer-learning* would be to improve the diagnosis of medical images ($\mathcal{T}_T$), where there are limited samples, by reusing features learned by a network pre-trained for another image classification task ($\mathcal{T}_S$). Note that even though the features are reused, the respective domains of the tasks are different ($\mathcal{D}_S = \mathcal{D}_T$), that is, $P(X|\mathcal{D}_S) = P(X|\mathcal{D}_T)$.

The approaches for transfer learning on DL have been identified by Tan et al. (Tan et al., 2018) as four: instance-based transfer, mapping-based transfer learning, network-based transfer (a.k.a. feature representation transfer), and finally adversarial based transfer. In this study, we have decided to follow an *inductive transfer learning* strategy implemented via a network-based transfer. The idea is to reuse a pre-trained U-Net variation (Krügener et al., 2022; Zdybal et al., 2022) on LF-LES data ($\mathcal{D}_S, \mathcal{T}_S$) to obtain a data-driven emulator fine-tuned on the HF-LES samples ($\mathcal{D}_T, \mathcal{T}_T$). The main hypothesis is, that the relationships between inputs and outputs in the HF and LF datasets do not differ significantly, thus, enabling the reuse of layers of the pretrained U-Net. In particular, the reused layers are those in which progressive feature maps are extracted from the inputs (the encoder block of the U-Net) and will be frozen. Finally, in those layers where the feature maps are used to construct the output (decoding block), such as an average temperature field ($\overline{T}(\boldsymbol{x})$), layers are fine-tuned, to embed the HF additional information linked to these features.

In the following, Sections 2.2 through 2.4 detail the LES datasets of shear-coaxial injectors at hand, and Section 2.5 explains in detail the methodology used for implementing the transfer learning. Finally, results are presented in Section 3 followed by concluding remarks in Section 5.

2. Methodology

The elaboration of the data-driven surrogate models detailed in this work begins with the data. Two sets of samples, one composed of 92 under-resolved large eddy simulations (LES), and the second of 19 highly refined LES simulations, were obtained via the *AVBP* solver (Schonfeld and Rudgyard, 1999). In both, the simulated case corresponds to a single-element shear-coaxial injector, burning, gaseous oxygen, GOx, and gaseous methane GCH_4, in a non-premixed configuration. The simulated design-points, each related to a different injector configuration, form a discrete set of samples from a three-dimensional (3D) design-space. The three design parameters selected for the design of experiments (DOE) are: the recess length of the oxidizer post (l_r), the chamber radius (d_c), which represent geometrical quantities of the injector; and the mixture-ratio (O/F), which is linked to its combustion-regime.

2.1. Shear-coaxial Injector LES setup

The injector used as template for the configuration of the different design-points was inspired on an experimentally tested single-element combustion chamber from the Technische Universität Munchen (TUM). The reader may find more details about this injector in (Celano et al., 2014; Chemnitz et al., 2018; Maestro et al., 2019; Perakis et al., 2017). The combustion-chamber geometry was modified to a circular cross-section chamber, to simplify the modeling. Only a third of the chamber is modeled because of computational constraints, yielding a chamber length of $l_c = 96.67$ mm. Considering the domain is axisymmetric along the injectors' axis, only a 30° slice is simulated. The resulting patches are conditioned

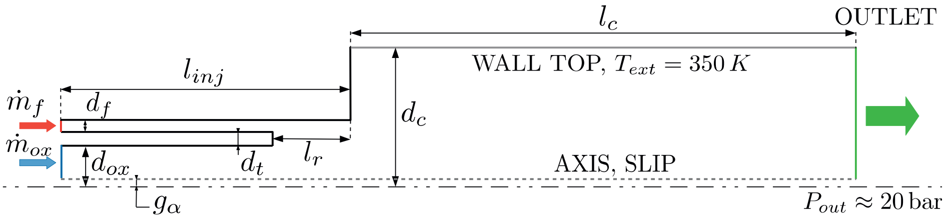


Figure 1. *Drawing of the geometry of the shear-coaxial injector used, with key dimensions identified. The fuel and oxidizer inlet channels and the outlet are signalized too.*

to a periodic boundary condition, while the axis is cut at a gap-height of $g_\alpha = 0.25$ mm to prevent a numerical singularity at this point. A slip surface is imposed over the resulting boundary. The remaining geometrical dimensions are based on the TUM injector: lip thickness $d_t = 0.5$ mm, oxidizer post radius $d_{ox} = 2.0$ mm, and methane annulus thickness $d_f = 0.5$ mm. Figure 1 depicts a drawing of a longitudinal cut of the adopted geometry.

Homogeneous mass flows of gaseous oxygen ($\dot{m}_{ox}$) and methane ($\dot{m}_f$) are imposed at the inlets. The injection temperatures are 278 and 269 K, respectively. The total mass flow $\dot{m}_{tot} = \dot{m}_{ox} + \dot{m}_f$ is kept constant to a value of 0.062 kg s^{-1} following the reference TUM experiment detailed in (Maestro et al., 2019). The outlet pressure is set to a constant magnitude of $P_{out} = 20$ bar. All inlets and outlets have Navier–Stokes characteristic boundary conditions (NSCBC) (Poinsot and Lelef, 1992) to handle the exiting acoustic waves from the modeled domain. In this first approach, no turbulence injection was considered.

The walls of the domain are modeled differently depending on the dataset. With regard to the under-resolved, *LF*-LES dataset, all walls are considered slip adiabatic except the chamber wall, denoted as "WALL TOP" in Figure 1. As for the *HF*-LES dataset, a no-slip wall law boundary condition was applied. Further details on these datasets' particularities are given in Sections 2.2 and 2.4. The "WALL TOP" is modeled as a conducting 1-mm thick copper surface, with conductivity $\kappa_{\mathrm{Cu}} = 401.0\frac{\mathrm{W}}{\mathrm{m}^2\mathrm{K}}$. A homogeneous and constant external temperature $T_s = 350$ K was applied on the dry-surface side. A thermally coupled wall-law, as implemented in *AVBP*, was chosen as the wall model. In this model velocity and temperature are coupled, which enables it to take into account significant density and temperature variations, as well as molecular Prandtl number effects. For the velocity profile, it is based on the Van Driest transformation (Van Driest, 1951), of the form:

$$u_{VD}^{+} = \frac{1}{\kappa}\ln(y^{+}) + 5.5 \tag{1}$$

where the u_{VD}^{+} corresponds to the Van Driest scaled velocity. Note that $u_{VD}^{+} \neq u^{+}$, as long as density variations occur inside the turbulent boundary layer. Meanwhile, $\kappa = 0.41$. The temperature profile in this "coupled" model, is thus recovered from the expression,

$$T^{+} = (\mathrm{Pr}y^{+})\exp(\Gamma) + (\mathrm{Pr}_t u^{+} + K)\exp\left(\frac{1}{\Gamma}\right) \tag{2}$$

where K is a constant that depends on the molecular Prandtl number (Pr) and Γ is a function that smooths the T^{+} profile between the laminar and the turbulent part, namely

$$\Gamma = -0.01\frac{(\mathrm{Pr}y^{+})^4}{1 + 5\mathrm{Pr}^3 y^{+}}. \tag{3}$$

The coupling between velocity and temperature is hence explicit from Eqn. (3). With respect to thermodynamic properties, these are taken from the node on the wall, at temperature T_{wall}. Finally, the wall heat flux $\dot{Q}_{wall}$ is expressed as,

$$\dot{Q}_{wall} = \frac{T_{ext} - T_{wall}}{R_{wall}} \tag{4}$$

where T_{ext} is the dry-surface temperature and R_{wall} the corresponding thermal resistance from the 1 mm copper wall.

The oxy-methane combusting mixture is modeled as a perfect gas with a global chemistry scheme (GLOMEC), developed by Blanchard and Strauss (Blanchard, 2021). The scheme is composed of six species: O_2, CH_4, CO, H_2O, CO_2 and H_2; and four chemical reactions and has been developed to target an operating pressure of ~ 20bar and similar equivalence ratios as those explored in our database. Note that no subgrid-scale TCI model has been used in this work to compensate for unresolved wrinkling and numerical diffusion. The known stiff chemistry linked to oxy-methane combustion was dealt with a similar exponential chemistry integration to that of Blanchard et al. (Blanchard et al., 2022). In all cases, a fully tetrahedral mesh was used, albeit of different characteristic sizes depending on the dataset. As for the convective terms, a Lax–Wendroff numerical scheme (Lax and Wendroff, 1960) was used in all cases. The σ model, developed by Nicoud et al. (Nicoud et al., 2011) is used to model the subgrid-scale stresses. More information is given concerning the datasets in Sections 2.2 and 2.4.

2.2. *Coarse LES database*

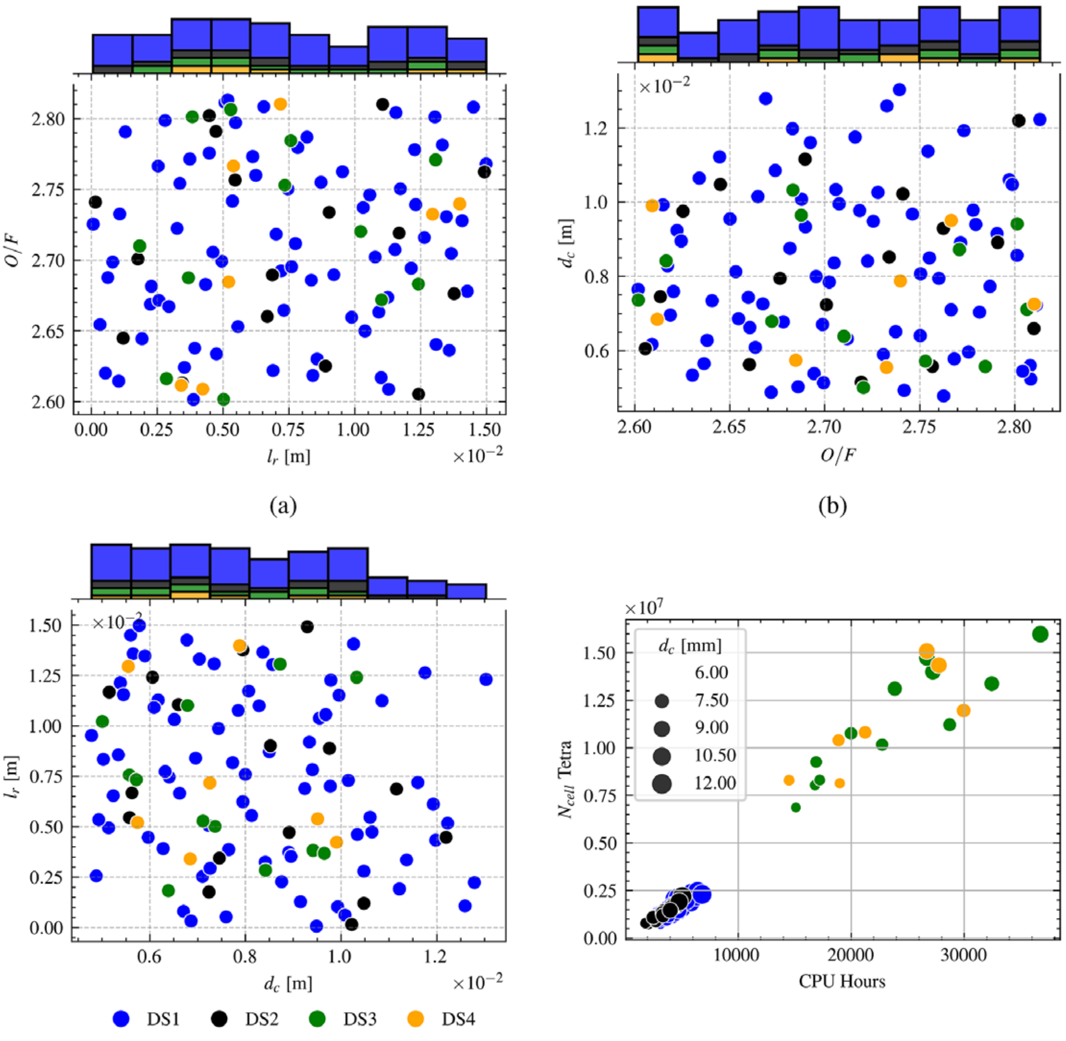
The first LES database consists of 92 LES simulations of shear-coaxial injectors. The mesh resolution is evaluated by the element thickness at the injector lip. On this under-resolved mesh, the reference length is $\Delta_0 \sim 100\mu$m at the injector lip. This yields five tetrahedral cells at the flame-anchoring position, where the highest mesh-resolution is required. For reference, simulations of a similar experimental test-case carried out by Maestro et al. (Maestro et al., 2019) used a value of $\Delta_0 = 25\mu$m. Chung et al. (Chung et al., 2021) reported 30μm close to the walls, while Blanchard et al. (Blanchard et al., 2022) reported $\Delta_0 = 40\mu$m, albeit for a different rocket-engine experimental combustor. The mesh under-resolution was intentional because of the limited computational resources available. To economize further in computational costs, slip adiabatic walls were adopted, except "WALL TOP" where a no-slip condition with a *coupled*[2] wall law was applied (see Section 2.1). Each LES sample is propagated for $\Delta t = 5$ms of time, including the ignition transient. This results on an average cost per sample of ~ 5000 CPU hours.

The numerical simulations which make up the *LF* database are labeled as points within a design-space $\mathcal{S} \in \mathbb{R}^3$. The dimensions of $\mathcal{S}$ are given by: the recess length (l_r), the mixture-ratio (O/F), and the chamber radius (d_c). The limits of $\mathcal{S}$ are detailed in Table 1. A detailed explanation of how these are derived is provided in (Zdybal et al., 2022). Finally, two different Latin Hypercube Samplings (LHS) (McKay et al., 1979) using the enhanced stochastic evolutionary algorithm (ESE) (Jin et al., 2005) were conducted. The first one (DS1), was designed to contain 100 samples, and the second (DS2) to contain 20 samples. However, not all LES conducted over the sampled points reach the 5 ms milestone. This is because of crashes of diverse nature which are beyond the interest of this work. The locations of the successful points $\mathcal{S}$, for example, those which reached the $\Delta t = 5$ms milestone, are highlighted in Figure 2a–2c. The convective time for these successful simulations is estimated via the formula,

Table 1. *Parameter range and reference values for the three parameters of the design of experiments*

Parameter	Symbol	Range	Reference
Recess-length	l_r	0–15 mm	0 mm
Chamber radius	d_c	6.7–13.41 mm	6.77 mm
Mixture-fraction	O/F	2.6–2.94	2.62

[2] In fact, the wall is modeled as a thin conductive wall with a constant temperature profile applied on the external surface.



(a)

(b)

(c)

(d)

Figure 2. *Joint plots of the points from the DOEs projected in 2D subspaces and consumption statistics graph. Colored histograms of the abscissa are displayed on top.* ***a)*** *l_r versus O/F,* ***b)*** *O/F versus d_c,* ***c)*** *d_c versus l_r,* ***d)*** *CPU hours consumed versus the number of tetrahedral cells in meshes used.*

$$\tau_{conv} = \frac{l_c}{\overline{u}_c}, \overline{u}_c = \frac{1}{\Omega_c}\int_{\Omega_c} u d\Omega \tag{5}$$

where u corresponds to the axial velocity component and Ω_c the chamber volume, for example, for all x beyond the injection plane, while l_c constitutes the chamber length. From the LES time-averages of DS1 and DS2, an average convective time across the coarse datasets ($\overline{\tau}_{conv}^{\mathrm{DS1}\cup\mathrm{DS2}}$) is computed, obtaining $\overline{\tau}_{conv}^{\mathrm{DS1}\cup\mathrm{DS2}} = 0.8 \pm 0.4\,\mathrm{ms}$. We hence consider the bulk of the LES converged for DS1 and DS2.

The top axis of each of the plots in Figures 2a–2c shows the histogram of the sampled points over the abscissa. Because of the uneven distribution of the aforementioned crashes, the LHS property of the sets is not verified in their final distribution. A sharp decrease in the density of samples is observed in Figure 2c for $d_c \geq 11\,\mathrm{mm}$, which can impact the performance of the surrogates overall in this region. After filtering

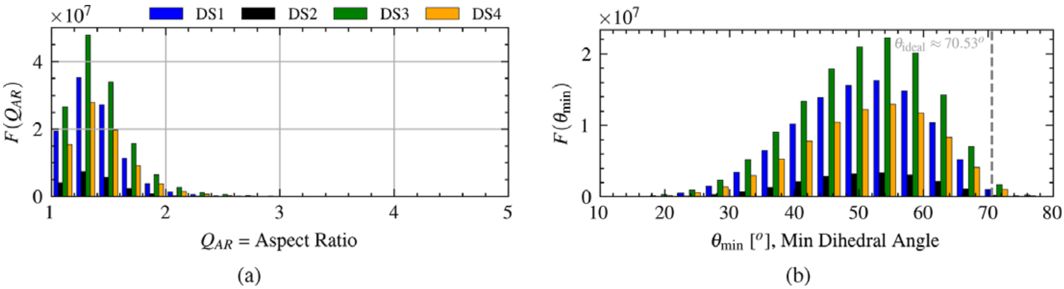


Figure 3. *Mesh quality measures statistics: a) Ensemble of datasets tetrahedral cells aspect ratio histograms, and b) tetrahedra minimum dihedral angle.*

the crashed simulations from DS1 and DS2, their respective run counts are 76 and 16, leading to a total of $|DS1| + |DS2| = 92$ coarse LES samples. Finally, the cost of each sample from DS1 and DS2 is portrayed in Figure 2d. Note that at similar grid resolutions, the cost scales almost linearly with the number of cells, and thus, d_c.

Two statistics on mesh quality have been estimated for the datasets involved. These are the tetrahedra cell aspect ratio (Q_{AR}) and the minimum dihedral angle of the cell (θ_{min}). The reader may found more information on how these are computed in (Stimpson et al., 2007). We proceed to compute these for every concerned grid via the *pyvista* mesh quality tool[3] In Figures 3a and 3a, we show the distribution of Q_{AR} and θ_{min}, respectively, for the all the datasets concerned, that is, $(\text{DS1}, \ldots, \text{DS4})$. As it can be seen in Figure 3a, a large proportion of the cells fall in the $Q_{AR} \in [1.0, 3.0]$ interval, which is considered to be the acceptable range for tetrahedral meshes. Meanwhile, in Figure 3b, the θ_{min} quantity histogram is presented. Note that the ideal dihedral angle for equilateral tetrahedra cells ($\theta_{\text{ideal}} \approx 70.53^o$), is superimposed to aid in the interpretation of the histogram. As we can see, the spread is quite large, with θ_{min} reaching as little as 20^o for some cells. However, the large majority of the tetrahedra cells in the meshes concerned are contained within the acceptable range interval (Stimpson et al., 2007) of $\theta_{\text{min}} \in [40^o, \theta_{\text{ideal}}]$, from which we conclude that the grid quality used in the elaboration of our datasets is acceptable.

From these datasets DS1 and DS2, a collection of U-Nets were trained as emulators of the longitudinal cut of mean quantities: the temperature field ($\overline{T}(\vec{x}), \vec{x} \in \mathbb{R}^2$), velocity[4] u ($\overline{u}(\vec{x})$), mixture-fraction ($\overline{Z}(\vec{x})$), oxygen mass fraction ($\overline{Y}_{O_2}(\vec{x})$) and velocity-u RMS Value ($u_{RMS}(\vec{x})$). The definition of Bilger(Bilger, 1989) is used for the mixture fraction, which for a O_2/CH_4 mixture yields a stoichiometric value of $Z_{st} = 0.2$. More details on the derivation of these models are given in Section 2.6.

2.3. LF-LES data analysis

The fidelity of supervised learning regression models developed on numerically sourced data and in the absence of physics-informed inductive biases, is directly related to the fidelity of the data they are based on. For such reason, it is of paramount importance to audit the LES samples, or at least the associated data-pipeline, as spurious structures may be present in the data, which are later distilled into the sought surrogates.

In the database introduced in Section 2.2, several modeling simplifications were conducted to reduce the computational cost per sample, notably: the mesh-resolution, the chemical-scheme, the geometry[5], the numerical schemes, and the wall boundary condition, among others. With such simplifications, it is thus

[3] The *pyvista* tool is a *Python* API of the well-known vTK toolkit [Schroeder et al., 2006], which resorts to the *Verdict* library [Stimpson et al., 2007] for computing diverse mesh quality parameters.

[4] Velocity-u corresponds to the axial-component of the velocity vector field in an Eulerian description of the flow-field.

[5] A 30° sector was defined as geometry instead of a complete revolution.

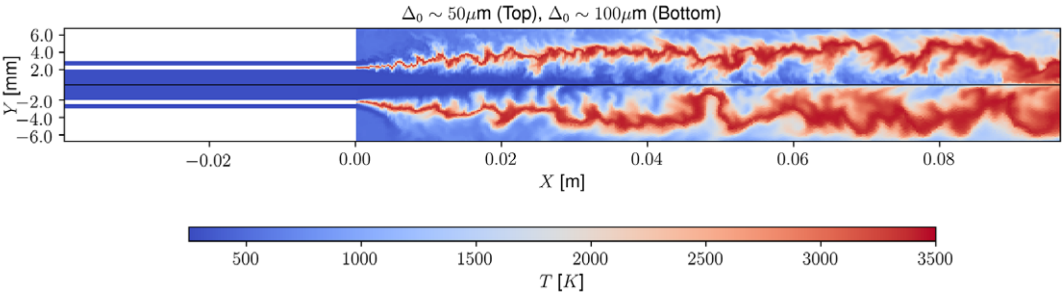


Figure 4. *The instantaneous longitudinal cut of the temperature field at different levels of mesh resolution. On top, a fine mesh is utilized ($\Delta_0 \sim 50\mu m$), while a coarse one ($\Delta_0 \sim 100\mu m$), is used for the bottom image.*

expected to obtain an LES of reduced quality. For instance, in Figure 4 instantaneous snapshots of the temperature field, at two different refinements, are presented for the TUM reference case. The differences between the fine mesh flame (Top), and the coarse one (Bottom), are evident. The coarse flame is much more diffused and expands faster. In the following, we determine the relative impact of the mesh resolution for the TUM reference case (Chemnitz et al., 2018).

To begin with, the wall heat flux was evaluated on different refinement levels for the baseline configuration, that is, that of the aforementioned TUM experimental rocket combustor. The azimuth-averaged wall heat-flux ($\dot{Q}_{wall}$) for different meshes is presented in Figure 5a, where N_{cells} denotes the number of tetrahedral cells in the mesh. The negative values of $\dot{Q}_{wall}$ indicate heat is exiting the control-volume. In addition, experimental measurements reported by Chemnitz et al. (Chemnitz et al., 2018) have been superimposed in Figure 5a. We can clearly observe a large overestimation of $\dot{Q}_{wall}$, by almost 400%, in the coarser mesh with respect to experimental values. This mesh resolution is the one utilized in the datasets DS1 and DS2 introduced in Section 2.2. However, increasing the mesh-resolution significantly reduces $\dot{Q}_{wall}$ along the entire chamber. Moreover, it also modifies the shape of the valley located at

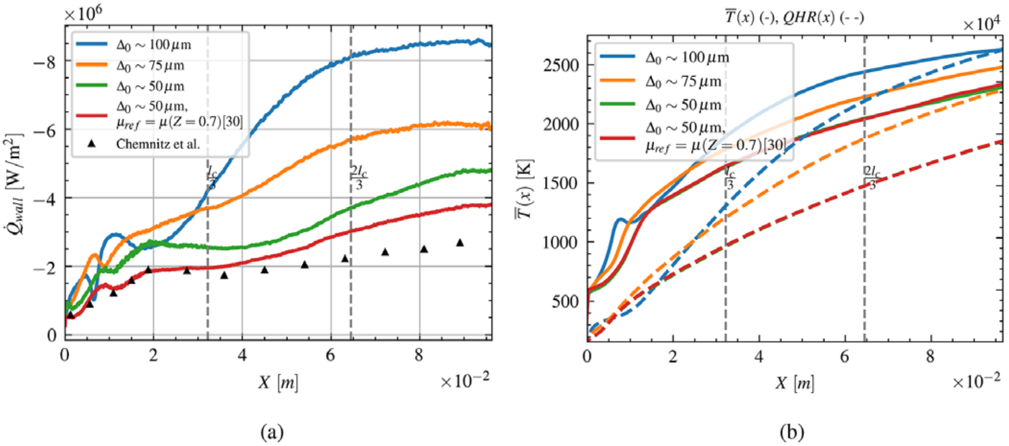


Figure 5. *Relevant quantities of interest (QoIs) evolution along the injector axial dimension in a 30° sector with parameters corresponding to the TUM reference case (Chemnitz et al. (Chemnitz et al., 2018). For reference $l_r = 0$ mm, $O/F = 2.62$ and $d_c = 6.77$ mm, while $l_c = 96.67$ mm designates the chambers' length.* ***a)*** *Azimuthally and time-averaged wall-heat flux $\dot{Q}_{wall}$ for three different mesh sizes, a fine case with an adjusted viscosity (μ) power-law and experimental measurements from Chemnitz et al.* ***b)*** *Cross-section and time-averaged temperature solution ($\overline{T}(x)$, full line) and cumulative heat release ($QHR(x)$, dashed lines) for three different mesh size.*

~ 10 mm of the origin (injection plane). This valley is normally linked to the location of the stagnation point following the recirculation zone of the injection.

The red curve in Figure 5a corresponds to the same mesh resolution of the green curve ($\Delta_0 \sim 50\mu$m), although the numerical simulation was carried out with a correction in the molecular viscosity power law used by Maestro et al (Maestro et al., 2019) on the LES of the same experiment. They argue that the better compromise for wall heat-flux evaluations is to fit the molecular viscosity, mass, and thermal diffusion terms to that of a mixture of mixture-fraction $Z = 0.7$. This comes from the fact that the mixture close to the walls is predominantly fuel-rich, given that the methane is injected through the annulus. Note that the perfect gas implementation of *AVBP* used in this work is not equipped with a calculator for laminar coefficients over mixture composition. Instead, a homogeneous laminar reference viscosity is considered in combination of a power-law to account for temperature dependency. The net impact of the correction, seen on the red line, is a further reduction in the wall heat-flux, with values matching those from the experiment for $x \leq l_c/3$. For $x > l_c/3$, predicted $\dot{Q}_{wall}(x)$ increases at a greater speed than that of experimental values. This heat flux rise is potentially caused by the misrepresentation of turbulent mixing in a thin axisymmetric volume, as previously reported in a similar configuration by Chung et al. (Chung et al., 2021) and other axisymmetric studies (Lapenna et al., 2018; Zips et al., 2017). Note that also a numerically thickened flame is expected because of the enhanced thermal diffusion, resulting in a thicker time-averaged temperature field.

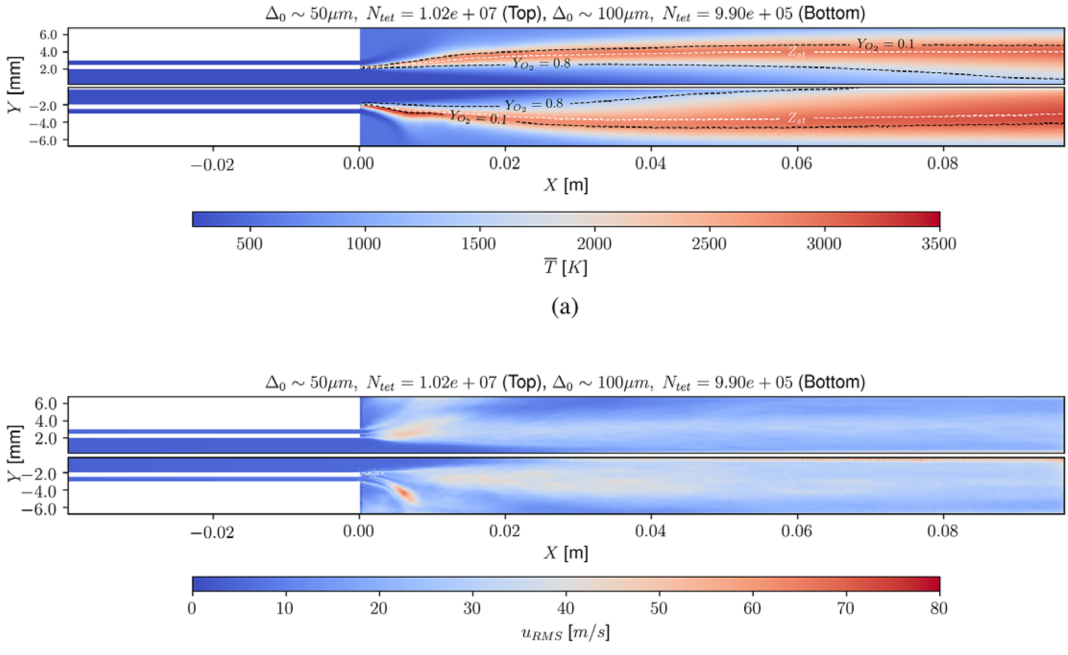
Figure 5b emphasizes the effects of the mesh resolution. In Figure 5b, the cross-section average temperature evolution along the axis $\overline{T}(x)$ and the cumulative heat-release $QHR(x) = \int_{-\infty}^{x}\int_S (x)HR(\eta,y,z)dSd\eta$, are shown. We can see that the $\overline{T}(x)$ difference grows along the axis of the chamber between the coarse mesh and the finer meshes. This trend is similarly followed by $QHR(x)$, indicating a much larger reaction rate in the coarser domain. The larger reaction rate is motivated by the numerical diffusion upstream, leading to greater dilatation and enhanced mixing downstream in the chamber. Moreover, we can observe that the effect of modifying the molecular transport properties as already mentioned by Maestro et al. (Maestro et al., 2019) (red line), has negligible effects in both the $\overline{T}(x)$ and $QHR(x)$. The green ($\Delta_0 \sim 50\mu$m) and red ($\Delta_0 \sim 50\mu\text{m}, \mu_{ref} = \mu(Z = 0.7)$) curves are superimposed. This is because of the fact that, contrary to near-wall transport phenomena, the bulk of the flow is predominantly driven by the turbulent mixing and less affected by the laminar diffusion processes.

The effects of the mesh resolution over the time-averaged fields are better seen in Figures 6a through 6b. In these, the fine mesh solutions ($\Delta_0 \sim 50\mu$m) are displayed on top, and the coarse ones ($\Delta_0 \sim 100\mu$m) below. Also in Figure 6a time-averaged oxygen mass fraction field isolines $\overline{Y}_{O_2} = 0.1, 0.8$ and the stoichiometric line denoted by Z_{st} have been superimposed. From the former, we can clearly observe the longer penetration of the oxygen jet for the fine mesh as the $\overline{Y}_{O_2} = 0.8$ isoline reaches further downstream. Furthermore, the difference in u_{RMS} appears predominantly at the near-field, with a *spurious* highly fluctuating structure developing at ~ 7.5 mm for the coarse mesh and overall greater intensity for $X \leq l_c/3$. Note that the location of this spurious structure coincides with that of the valley of $\dot{Q}_{wall}$ and the crest in $\overline{T}(x)$ in Figures 5a and 5b and disappears when refining.

The results hereby shown motivated the creation of two more datasets of HF samples, DS3 and DS4. To serve for fine-tuning during the transfer-learning task, from models trained and tested on coarse LES simulations. The details of this additional dataset are given in Section 2.4.

2.4. HF-LES dataset

The shortcomings evidenced on the coarse LES simulations scrutiny motivated the creation of an HF dataset with greater mesh resolution. A uniform scaling factor of 0.5 is applied on both surface and volume cells, which leads to $\Delta_0 \sim 50\mu$m, albeit to a much greater computational cost. As we do not focus on wall transport phenomena in this work, we have not modified the molecular diffusion coefficients, nor the laminar reference viscosity of the mixture with respect to the coarse base configuration. Moreover, this would require knowing a priori the composition close to the walls throughout the database, which is not attainable. Two LHS of 20 samples each are conducted to generate DS3 and DS4 over the same design-space described in Table 1. As detailed in Section 2.7, the DS3 samples are meant for training and



(a)

(b)

Figure 6. *Side-by-side comparison of two different levels of mesh-resolution for* ***a)*** *averaged temperature field solution longitudinal cut with oxygen mass fraction isolines $\overline{Y}_{O_2} = 0.1,\ 0.8$ and stoichiometric line, Z_{st} superimposed.* ***c)*** *Velocity-u root mean squared (RMS) field longitudinal cut. The fine mesh solution, $\Delta_0 \sim 50\mu m$ is shown on top ($N_{tet} = 1.02 \times 10^7$), and below are displayed the solutions for the coarse case, $\Delta_0 \sim 100\mu m$ ($N_{tet} = 9.9 \times 10^5$) which corresponds to the resolution adopted in datasets DS1 and DS2. The scale of the figures has been modified to simplify its visualization.*

validation, meanwhile, the DS4 ones are reserved as a test dataset. In a similar fashion to DS1 and DS2, the sampled points have been included in Figure 2a through 2c, as well as their computational cost included in Figure 2d.

To save computing resources, the injectors are ignited for 2 ms of simulated time in a coarse mesh and later interpolated to the finer one and continued until 6 ms of total simulated time is achieved. Note that, in spite of this measure, the cost per sample may well reach 40,000 CPUhs, compared with <10,000 in DS1 and DS2. Of the original samples, only 12 reached the $\Delta t = 6$ms milestone for DS3, whereas 7 for DS4. Following the convective time introduced in Eqn. (5), and average convective time for the HF-LES datasets, DS3 and DS4 is obtained, yielding $\overline{\tau}_{conv}^{\mathrm{DS3} \cup \mathrm{DS4}} = 0.95 \pm 0.4$ms. Thus, we regard the LES from DS3 and DS4 as converged in terms of their time-averaged fields. In Table 2, a synthesis of the dataset count

Table 2. *Summary of LES datasets of shear coaxial injectors simulations. The sampled design-space in all cases is detailed in Table 1*

Dataset	Count	Resolution -Δ_0	Avg. CPU hours
DS1	76	Low $\sim 100\mu m$	3699.47
DS2	16	Low $\sim 100\mu m$	3290.66
DS3	12	High $\sim 50\mu m$	23703.0
DS4	7	High $\sim 50\mu m$	22575.8

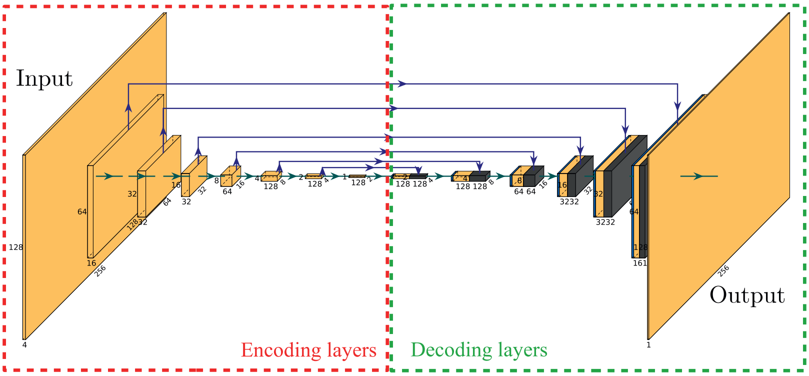


Figure 7. *U-Net architecture is being used. The input layer expects a tensor of four channels, each with of* 128 × 256*. These channels correspond to the three normalized parameters O/F,* l_r *and* d_c*, and a one-hot encoding tensor, the mask, which delimits the fluid region. The output layer returns a single channel corresponding to the normalized predicted quantity with the same resolution. The arrows connecting the encoding layers to the decoding ones are skip-connections.*

and basic metrics is provided. The design-space from which the points have been sampled is detailed in Table 1.

2.5. Transfer learning methodology

The transfer-learning strategy adopted in this work is that of *inductive transfer learning* with a *network-based transfer* approach, also known as feature representation transfer. As described in Section 1.2, this approach reuses part of a neural-network $f_S(\cdot)$ trained on a source domain $\mathcal{D}_S$ to learn a new target task $\mathcal{T} = \mathcal{D}_T, f_T(\cdot)$.

The problem at hand thus can be framed as, learning a data-driven model ($f_T(\cdot)$) from HF-LES samples ($\mathcal{T}, y_T$) by reusing the pre-trained model ($f_S(\cdot)$) on *coarse* LF-LES samples ($\mathcal{D}_S$). The practical aspects of such approach are better explained when visualizing the U-Net architecture. Figure 7 displays a schema of the U-Net architecture being used. The input layer expects a tensor of four (4) channels, three are flat-tensors embedding the normalized input quantities O/F, l_r and d_c, while the fourth is a one-hot encoding tensor, or *mask*, delimiting the fluid region. The output layer returns a single channel with the normalized predicted field, over a 128 × 256 grid resolution.

The U-Net may be partitioned into two major zones: an encoding zone and a decoding one. The former is composed of successive blocks of 2D convolutional, batch-normalization, and dropout[6] layers. The latter is composed of blocks of transposed 2D convolutional, upsampling, batch normalization, and dropout layers. Leaky ReLU activation functions are used all across the different layers. Skip-connections between encoding and decoding blocks are signaled by arrows in Figure 7.

As with any CNN, the encoding blocks decompose the input channels into *feature-maps* which are higher-level representations of the information contained in the input. Through the skip-connections, these are forwarded to the decoding blocks, which progressively recover the desired output. For the transfer-learning task at hand, we can exploit the fact that both the low and HF samples have been obtained from the same design-space. Hence, it is reasonable to think that the feature-maps recovered from the LF-LES model ($f_S(\cdot)$) encoding layers are shared by a potential HF model ($f_T(\cdot)$). The principal difference relies on how decoding layers process the feature-maps to recover the intended field.

[6] The dropout layers are only required during training as a counter-overfitting measure [Chollet, 2017. They do not apply during inference. In case dropout value is set to 0 then these are not considered during training.

With this in mind, we split the collection of the U-Nets 486.337 trainable parameters in two sets: θ_F and θ_{ft}, such that the training over the HF model $f_T(\cdot)$ is defined by:

$$f_T(\cdot) = p_{\theta_F \cup \theta^*_{ft}}(\cdot) \,|\, \theta^*_{ft} = \text{argmin}_{\theta_{ft}} \mathcal{L}\left(p_{\theta_F \cup \theta_{ft}}(\cdot), X_T, Y_T\right) \tag{6}$$

where $p_\theta(\cdot)$ is the U-Net, θ_F represent the reused *source* network parameters, θ_{ft} the parameters to fine-tune the HF dataset $[X_T, Y_T]$ via the loss function $\mathcal{L}(\cdot)$. θ^*_{ft} identifies the already fine-tuned parameters. A priori, during the transfer-learning, the θ_F parameters correspond to those of the encoding blocks, except the last encoding convolutional layer. The tunable parameters θ_{ft} in the first approach are a collection of weights and biases from the last encoding convolutional layer and decoding transposed convolutional layers.

2.6. Source models training

The LFMs, defined as sources of the transfer-learning process, are trained following the procedure detailed in (Zdybal et al., 2022). The training and validation samples are drawn from LF samples belonging to DS1, meanwhile, DS2 samples are reserved as test datasets. Input and outputs are normalized following,

$$z_i = \frac{x_i^k - \mu^i_{x,S}}{\sigma^i_{x,S}}, \quad \left[\boldsymbol{q}_j^k\right]_{pq} = \frac{\left(\left[\boldsymbol{y}_j^k\right]_{pq} - \mu^k_{y,S}\right)}{\sigma^k_{y,S}} \tag{7}$$

where z_i is the normalized i-component of the input space with axes O/F, l_r and d_c. $\mu^i_{x,S}$ and $\sigma^i_{x,S}$ are, respectively, the mean and standard deviation of the i-component calculated over the parameters of the DS1 samples. Similarly, the outputs fields j-th sample $\boldsymbol{y}_j^k \in \mathbb{R}^{128 \times 256}$ are linearly scaled by their respective standard deviations and means $\sigma^k_{y,S}$, $\mu^k_{y,S}$. The superscript k is introduced to denote the k-th field-quantity. The subscript S denotes it belongs to the *source* dataset DS1. The field quantities modeled are the time-averaged temperature $\overline{T}$, oxygen-mass fraction field $\overline{Y}_{O_2}$, mixture-fraction $\overline{Z}$, velocity-u component $\overline{u}$, and the velocity-u RMS value u_{RMS}.

A linear combination of the standard mean-squared error loss (MSE) and gradient difference loss (GDL) is defined as a loss function,

$$\mathcal{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \alpha_{MSE}\text{MSE}(\boldsymbol{y}, \hat{\boldsymbol{y}}) + \alpha_{GDL}\text{GDL}(\boldsymbol{y}, \hat{\boldsymbol{y}}) \tag{8}$$

It is worth noticing that the loss is only considered on the fluid zone, which is signaled by passing the one-hot encoding tensor, or *mask*, to compute $\mathcal{L}(\cdot)$ over the region of interest. The training is conducted with the Adam optimization algorithm (Kingma and Ba, 2014) with a batch size of 16 over 500 epochs in all cases involved, except for u_{RMS}, where a batch size of 32 was used. An exponential learning rate decay (LR Decay) scheduler is also used to control the decrease of learning rate with the epoch number. To find the best hyperparameters combination for performance over the validation set, a Bayesian search is conducted via the *wandb* (Biewald, 2020) framework with a maximum number of ~ 250 different hyperparameters configurations. Because of the limited number of samples, the performance metric is estimated over a 10-fold cross-validation. The performance metric varies on the field quantity predicted. The best performing models are introduced in Table 3, alongside their average performances over the 10-fold validation and test datasets. The performance metric for $\overline{T}$ is the 10-fold validation average relative error $\overline{E}_{r,val}$, whereas for the remaining quantities, it is the 10-fold validation average normalized error $\overline{E}_{norm,val}$, defined by:

$$\begin{aligned}
\overline{E}^k_{rel,DS} &= \frac{1}{N_S}\sum_j^{N_S}\frac{1}{S_j}\sum_{p=0}^{N_y}\sum_{q=0}^{N_x}\left[\boldsymbol{M}_j\right]_{pq}\frac{\left|\left[\boldsymbol{y}_j^k\right]_{pq} - \left[\hat{\boldsymbol{y}}_j^k\right]_{pq}\right|}{\left|\left[\boldsymbol{y}_j^k\right]_{pq}\right|} \\
\overline{E}^k_{norm,DS} &= \frac{1}{N_S}\sum_j^{N_S}\frac{1}{S_j}\sum_{p=0}^{N_y}\sum_{q=0}^{N_x}\left[\boldsymbol{M}_j\right]_{pq}\frac{\left|\left[\boldsymbol{y}_j^k\right]_{pq} - \left[\hat{\boldsymbol{y}}_j^k\right]_{pq}\right|}{\left|\mu^k_{y,S}\right|}
\end{aligned} \tag{9}$$

Table 3. *Best-performing networks resulting from the Bayesian hyperparameters optimization. The parameter values and the network performances over the validation and test dataset (DS2) are given*

Parameter	$\overline{T}$	$\overline{Y}_{O_2}$	$\overline{Z}$	$\overline{u}$	u_{RMS}
Batch size	16	16	16	16	32
Initial LR	0.003388	0.004124	0.001861	0.003411	0.00562
LR Decay	0.98	0.98	0.98	0.98	0.98
Weight Decay	0.0	0.0	0.02	0.02	0.02
α_{GDL}	70.0	40.0	40.0	100.0	55.0
Performances					
$\overline{E}_{r,val}$	2.89%	–	–	–	–
$\overline{E}_{norm,val}$	2.108%	2.955%	2.562%	3.292%	6.403%
$\overline{E}_{r,test}$	2.847%	–	–	–	–
$\overline{E}_{norm,test}$	2.032%	2.793%	2.226%	3.006%	6.203%

where $N_x = 256$, $N_y = 128$. $\left[\boldsymbol{y}_j^k\right]_{pq}$ and $\left[\hat{\boldsymbol{y}}_j^k\right]_{pq}$ are the p, q elements of the j-sample and predicted tensor, respectively, on the dataset DS of N_s samples, over the k-th field. The quantity $\left[\boldsymbol{M}_j\right]_{pq}$ represents the one-hot encoding tensor, or mask defining the fluid region where $S_j = \sum_{p=0}^{N_y}\sum_{q=0}^{N_x}\left[\boldsymbol{M}_j\right]_{pq}$ is a mere count of the number of fluid cells contained in it. The quantity $\mu_{y,S}^k$, defined above, is introduced in $\overline{E}_{norm}$ to prevent zero division errors in those fields which can be zero-valued, such as $\overline{Y}_{O_2}$, $\overline{\Xi}$, $\overline{u}$ and u_{RMS}.

2.7. Target models training

The training of the HF models begins by initializing the networks, with the architecture depicted in Figure 7 with the weights and biases of the best-performing networks from the LF-LES (*source*) task, indicated in Table 3. The layers of the U-Net to fine-tune during the HF-LES (*target*) task are defined as an additional discrete hyperparameter for the Bayesian hyperparameters' optimization of the network. The layers involved span from the last encoding layer to the output layer, passing through all the decoding layers. The samples from DS3 are used for training and validation. In the meantime, those from DS4 are guarded as test datasets, with the exclusive purpose of providing an unbiased performance estimation. Because of the small size of DS3, it was chosen to proceed to calculate the performance metrics, either $\overline{E}_{rel}$ or $\overline{E}_{norm}$, via a six-fold cross-validation scheme. Furthermore, the full training set is fixed as batch size.

Before training, both inputs and outputs in the training dataset are scaled following Eqn. (7) while preserving $\mu_{x,S}^i$, $\mu_{y,S}^k$, $\sigma_{x,S}^i$ and $\sigma_{y,S}^k$ from the *source* dataset DS1. The hyperparameter optimization is carried out with the corresponding performance metric of the field, and a maximum of 200 hyperparameter combinations are tested per quantity. Each configuration is trained with a 1000 maximum epochs limit. Similarly to the source model hyperparameters optimization, the performance metric to minimize is the six-fold average relative error over the validation samples $\overline{E}_{rel,val}$, for the $\overline{T}$ field. Simultaneously, the six-fold average normalized error over the validation set $\overline{E}_{norm,val}$ is chosen for the remaining fields to prevent zero division. These performance metrics are summarized in Table 4. The loss function corresponds to that defined in Eqn. (8), identical to that of the *source* task. The remaining hyperparameters range for this task are presented in Table 4. Bear in mind that during the hyperparameters' optimization, these are uniformly sampled from a discrete set a priori defined by its range and quantization, except for the initial learning rate (Init LR in Table 4) which is sampled uniformly from a continuous real interval.

As shown in Table 4, the parameter β_1 involved in the first momentum term update, that is, the running average of gradients, of the Adam algorithm, has been added. In addition, dropout is considered, albeit over two values, as a counter-overfitting measure in view of the limited number of samples available for

Table 4. *The list of hyperparameters varied for the HF model learning, their preset intervals and quantization for each of the field-quantities models provided*

Description		$\overline{T}$	$\overline{Y}_{O_2}$	$\overline{Z}$	$\overline{u}$	u_{RMS}
Initial LR	Range	0.001–0.01	0.001–0.01	0.001–0.01	0.001–0.01	0.001–0.01
LR Decay	Range	0.96–0.99	0.96–0.99	0.96–0.99	0.96–0.99	0.96–0.99
	Quantization	0.01	0.01	0.01	0.01	0.01
β_1	Range	0.5–0.9	0.5–0.9	0.5–0.9	0.5–0.9	0.5–0.9
	Quantization	0.1	0.1	0.1	0.1	0.1
Dropout	Values	[0, 0.01]	[0, 0.01]	[0, 0.01]	[0, 0.01]	[0, 0.01]
Weight Decay	Range	0.1–0.3	0.1–0.3	0.1–0.3	0.1–0.3	0.1–0.3
	Quantization	0.05	0.05	0.05	0.05	0.05
α_{GDL}	Range	50–100	50–100	50–100	50–100	50–100
	Quantization	10	10	10	10	10
Performance Metric		$\overline{E}_{rel,val}$	$\overline{E}_{norm,val}$	$\overline{E}_{norm,val}$	$\overline{E}_{norm,val}$	$\overline{E}_{norm,val}$

training. However, it is expected that the dropout layers might hinder convergence because of the randomness added during the training process, in particular when the batch-size is small. Weight decay is deemed necessary to prevent overfitting in all cases.

3. Results and analysis

The best-performing network configurations obtained from the Bayesian optimization of the hyperparameters are introduced in Table 5. The configuration selected minimizes the performance metric detailed in Table 4, within the set of probed configurations. Their average relative and normalized errors over the training, validation, and test samples, where suitable, are given. Note that for these emulators, training, and validation datasets are made of HF-LES samples drawn from the DS3 dataset. In the following of this

Table 5. *Best-performing networks resulting from the Bayesian hyperparameters' optimization for the MFMs*

Parameter	$\overline{T}$	$\overline{Y}_{O_2}$	$\overline{Z}$	$\overline{u}$	u_{RMS}
Trainable Parameters	91,921	91,921	91,921	91,921	91,921
Initial LR	0.00878587	0.00541211	0.00902745	0.00164691	0.00285097
β_0	0.7	0.7	0.6	0.7	0.5
LR Decay	0.98	0.97	0.99	0.99	0.98
Weight Decay	0.1	0.1	0.1	0.15	0.1
Dropout	0	0.01	0	0.01	0.01
α_{GDL}	80	80	60	50	80
Final Epochs[a]	1000	1000	1000	1000	1000
Performances					
$\overline{E}_{rel,train}$	5.11%	–	–	–	–
$\overline{E}_{norm,train}$	3.23%	4.65%	4.17%	5.22%	6.68%
$\overline{E}_{rel,val}$	6.99%	–	–	–	–
$\overline{E}_{norm,val}$	4.42%	5.98%	5.80%	6.87%	8.87%
$\overline{E}_{rel,test}$	5.98%	–	–	–	–
$\overline{E}_{norm,test}$	3.86%	5.12%	4.87%	6.06%	7.81%

[a] Final Epochs is the number of epochs over which the training has been conducted until the convergence criteria for loss value ($< 10^{-5}$) is fulfilled.

Table 6. *LF and MFMs performances comparison*

	$\overline{T}$	$\overline{Y}_{O_2}$	$\overline{Z}$	$\overline{u}$	u_{RMS}
Task	$\overline{E}_{rel,test}$	$\overline{E}_{norm,test}$	$\overline{E}_{norm,test}$	$\overline{E}_{norm,test}$	$\overline{E}_{norm,test}$
Low-fidelity[a]	2.85%	2.79%	2.23%	3.01%	6.20%
Multifidelity	5.98%	5.12%	4.87%	6.06%	7.81%

[a]LFMs test dataset corresponds to DS2. The $\overline{E}_{rel,test}$ and $\overline{E}_{norm,test}$ have been extracted from Table 3. Meanwhile, for the MFMs the test dataset is DS4 and the values have been drawn from Table 5.

text, this set of data-driven emulators will be referred to as MFMs, to differentiate them from those trained on LF-LES datasets detailed in Table 3.

All the best-performing networks show a total number of trainable parameters (θ_{ft}) of 91,921. Note that the total number of parameters of the network is 486,337 ($\theta_F \cup \theta_{ft}$). Of all the possibilities of layers combinations explored for this fine-tuning task, $|\theta_{ft}| = 91{,}921$ is the largest one in terms of trainable parameters. This configuration involves the last encoding convolutional-layer and the ensemble of transposed convolutional layers of the decoding side of the U-Net (see Figure 7). In fact, it was seen that choosing a smaller set of layers had a detrimental effect on the performance of the MFM.

The number of samples for training available being small, weight decay was enforced to prevent overfitting while dropout was kept optional. In Table 5 we see that for $\overline{T}$ and $\overline{Z}$ models use predominantly weight decay whereas $\overline{u}$, $\overline{Y}_{O_2}$, and u_{RMS} use a combination of both dropout and weight-decay. In all cases, the weight-decay value is close to the lower end of the enabled range, as shown in Table 4. This may indicate the chosen range for this hyperparameter is inadequate. Nonetheless, no significant evidence of overfitting was observed in any of the models involved.

It is worth comparing the performances of the LF and MFMs, over their respective test datasets and tasks, to verify if they attain similar scores. On the one hand, the LFMs task consists of predicting the LF-LES solutions[7]: $\overline{T}$, $\overline{Y}_{O_2}$, $\overline{u}$, $\overline{Z}$, and u_{RMS} from the normalized design-space coordinates z_i. These models have been trained consequently on LF-LES data, DS1, and evaluated in a separate test-dataset of the same characteristics, DS2. On the other hand, the MF model tasks aim to predict the HF-LES solutions from the normalized design-space coordinates, while being fine-tuned with HF-LES data, DS3, and consequently evaluated over samples of the same characteristics, DS4. The scores of both MFMs and LFMs over their respective test-datasets are summarized in Table 6. The LFMs show a smaller error for their respective tasks compared with the multifidelity ones, although they both remain low and in the same order of magnitude. A potential explanation for the larger errors seen in the multifidelity task is the limited number of HF samples available for training which may result in poor coverage of the design-space. Nevertheless, the core advantage of the transfer learning strategy lies in the economy of CPU hours when creating the datasets. Conducting the training and testing purely on sets of HF samples with the same count as DS1 and DS2, would have implied, approximately, an additional ~ 1.4 million CPUhs.

It is worth noting that during the training of the U-Net, a linear combination of a mean squared error (MSE), and gradient difference loss (GDL), as indicated in Eqn. 8. The approach hereby taken is therefore purely data-driven, as no physics-based term has been added as a constraint to guide the training. While it is reported in the literature that the inclusion of physics-informed inductive biases enhances the performance of machine learning models in both interpolation and extrapolation in the low data regime (Bermejo-Barbanoj et al., 2024; Choi, 2023; Cicirello, 2024; Cueto and Chinesta, 2023; Hernandez et al., 2021; Jeon et al., 2024; Kim et al., 2022; Z. Wang et al., 2023), in this work we have not implemented any physics-based constraint to guide the learning. The primary reason for this relies on practical grounds: for

[7] In this context, we refer to solution as specific scalar-fields longitudinal cuts of the simulated domain.

the present isolated average field prediction workflow, we have not found a suitable learning bias. Moreover, many physics-based terms found in the literature are evaluated over the instantaneous state values, as is the case of Thermodynamics Informed Neural Networks (Cueto and Chinesta, 2023). In future works, we intend to adopt this avenue, however by shifting the focus from time-averaged predictions to modeling the problem as a structured dynamical system.

3.1. MFMs benchmarking

The subsequent figures present a side-by-side comparison of the predictions of the MFMs and LFMs for the different field-quantities involved. These are evaluated on an arbitrary sample of the HF test dataset (DS4) with parameters: $O/F = 2.81$, $l_r = 7.2\text{mm}$ and $d_c = 7.3\text{mm}$. The MF models are those detailed in Table 5 whereas the LF have been used as *source* networks on the transfer-learning procedure and explained in Section 2.6. Figure 8a displays on top the MF $\overline{T}$ prediction, with the LF one below. The predicted $\overline{Y}_{O_2} = [0.1, 0.8]$ isolines have been superimposed as well as the stoichiometric line Z_{st}, located at $\overline{Z} = 0.2$ for the $O_2 - CH_4$ propellant couple.

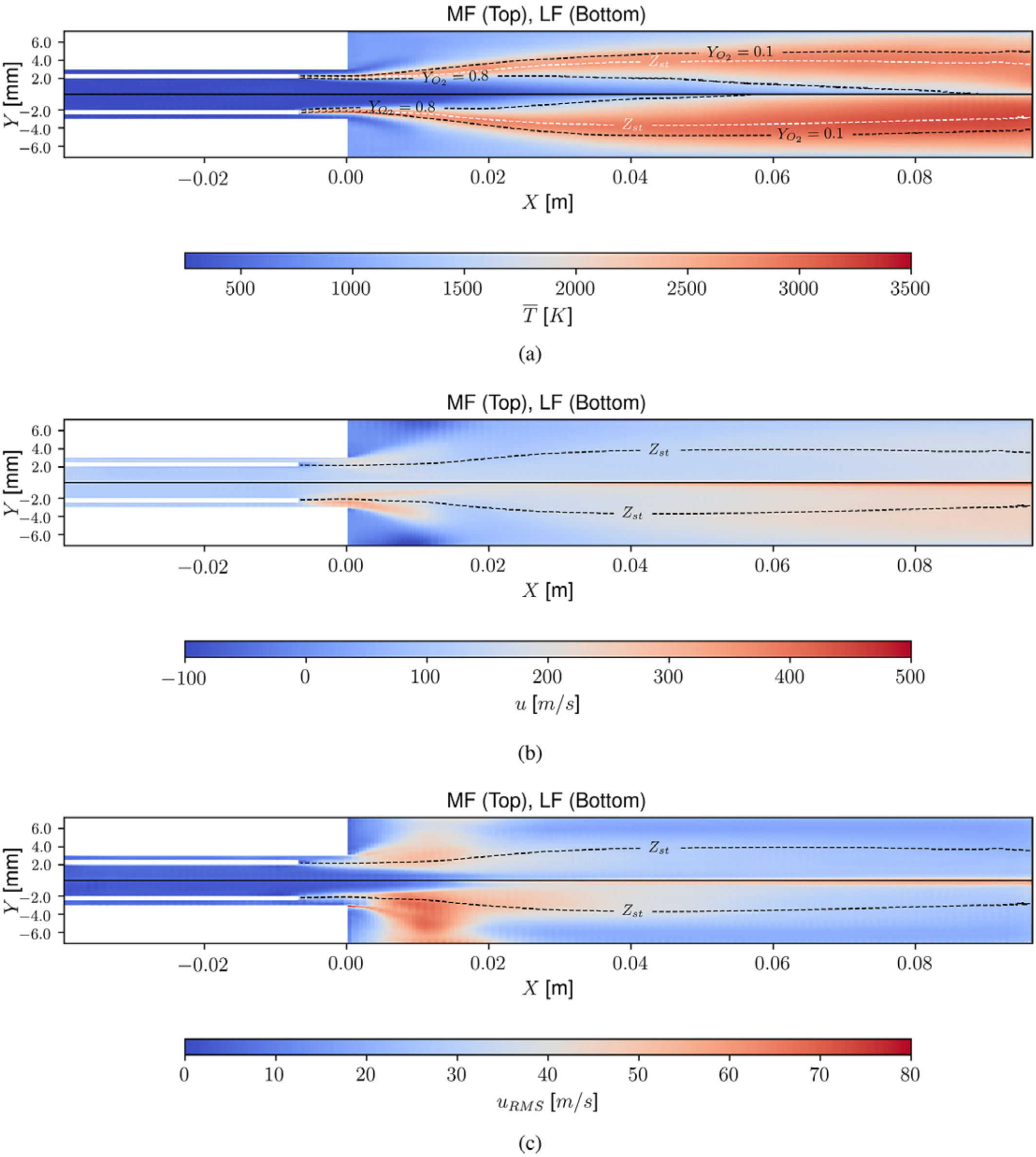
It is evident that the MF flame shape is in line with what is seen in Figure 6a. The hot gases zone is comparatively thinner, with an oxygen-jet penetrating further into the chamber, a sign of a smaller reaction rate. Also, the Z_{st} line is slightly less concave, indicating a longer flame overall. The change between MF and LF is better seen in Figures 10a and 10b, where the difference between the estimated, $\overline{T}_{pred}, \overline{Y}_{O_2,pred}$, and the ground-truth fields $\overline{T}_{gt}, \overline{Y}_{O_2,gt}$ of the test sample are displayed. A strong red area, below the Z_{st} line, is shown in Figure 10a for LF. Similarly, the same area is marked in blue in Figure 10b. Comparatively, the MF model error figures do not display this region, indicating that the transfer-learning provides the sought correction.

Figures 9a and 9b show the y-averaged values of the MF, LF, and ground-truth (HF-LES*) for $\overline{T}$ and $\overline{Y}_{O_2}$, respectively. The averages were calculated over the first dimension of the 128×256 grids. In these, the effect of the correction is evident, as we see the trend from the HF-LES* is recovered, whereas the LF prediction largely overestimates it. The $\overline{Y}_{O_2}(x)$ plot shows also the HF-LES* values are reconstructed correctly by MF, whereas LF largely underpredicts the oxygen content by the end of the chamber.

The $\overline{u}$ and u_{RMS} models estimation on the selected sample are plot in Figures 8b and 8c, respectively. Their errors with respect to ground-truth are shown in Figures 10c and 10d. The predicted stoichiometric line (Z_{st}) has been added for guidance. Similarly to $\overline{T}$ and $\overline{Y}_{O_2}$, the MF model shows positive results, with an overall smaller velocity-u downstream in the chamber, consistent with the smaller temperature. The velocity-u is also corrected appropriately at the recessed channel and in the near-field of the injection plane ($x \sim 0$).

As for the u_{RMS}, the MF indeed predicts less intense fluctuations at the near-field[8] and in the first half of the chamber. The error plot shown in Figure 10d accentuates this change. Furthermore, the near-axis turbulence, observable as a high u_{RMS} line in the proximity of the axis of the LF prediction in Figure 8c, attenuates significantly for the MF prediction. These spurious fluctuations have been associated with the boundary condition at the axis. Nonetheless, the stronger expansion in the chamber when utilizing coarse resolutions, leads to a local acceleration of the flow and greater velocity gradients, increasing the turbulent activity near the axis. The $u_{RMS,pred} - u_{RMS,gt}$ field in Figure 10d shows also the MF model struggles with the topology of the u_{RMS} field in the near-field. In spite of providing an overall more accurate estimate than LF, it still shows a complex error map in the near-field. A similar issue is seen in the $\overline{T}$ field MF predictions. The near-field in shear-coaxial injectors is an intricate zone where multiple fluid structures of different scales interact, thus yielding a complex local topology, characterized by high gradients. Therefore, in future works, it could be beneficial to prioritize learning in this area through a locally oriented loss function.

[8] As depicted later in Figure 12a, near field is referred to as the region of the chamber, in the vicinity of the injection plane, such that $x \in (0, 10d_{ox}]$. x and d_{ox} represent the axial coordinate and the oxidizer.



(a)

(b)

(c)

Figure 8. *Predictions of the MFM, shown on top and LFM below for the test dataset sample with parameters: $O/F = 2.8104$, $l_r = 7.2$mm and $d_c = 7.3$mm.* ***a)*** *The time-averaged temperature field $\overline{T}$ with $\overline{Y}_{O_2} = 0.1, 0.8$ oxygen mass fraction isolines as well as the stoichiometric line Z_{st} in white.* ***b)*** *The time-averaged velocity-u (axial) component, $\overline{u}$ with the predicted stoichiometric line superimposed in dashed-black lines.* ***c)*** *Velocity-u RMS field (u_{RMS}) with the predicted stoichiometric line superimposed in dashed-black lines. The figures' aspect-ratio has been adjusted to ease visualization.*

Figures 11 and 12b are meant to show insight into the error distribution, both in the design-space but also along the geometry of the injector. In Figure 11, the corresponding MFMs and LFMs, error metrics for each sample of the MF test-dataset, DS4, are given as bar plots. The design-space coordinates, in terms of $(O/F, l_r, d_c)$ are provided for each of the samples to help identify patterns in the error distribution. The MF model error metrics per sample are averaged over the six folds over which training was conducted.

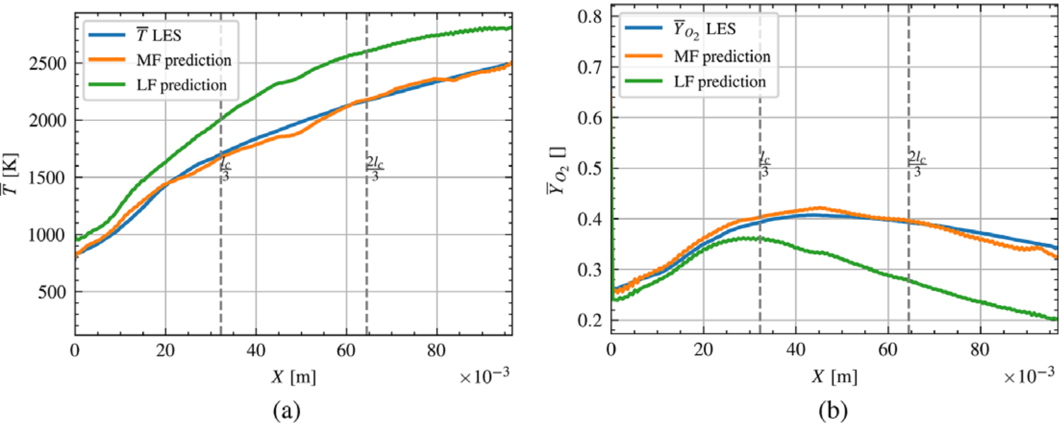


Figure 9. *Cross-section averaged of ground-truth and predictions of the:* ***a)*** *time-averaged temperature field predictions and ground-truth and* ***b)*** *time-averaged oxygen mass-fraction predictions and ground-truth. Dashed gray lines 0 indicating the position of third ($l_c/3$) and two-thirds ($l_c/3$) of the chamber-length have been added for reference. The origin $x = 0$ corresponds to the location of the injection plane.*

The dataset averages are overlaid on each bar as dark circles. The error-bars of these dots show the dispersion of the associated MF model error metric over the dataset DS4. These are meant to serve as a reference for comparison.

When focusing the attention on the $\overline{E}_{rel,test}(\overline{T})$, $\overline{E}_{norm,test}(\overline{Y}_{O_2})$, and $\overline{E}_{norm,test}(\overline{Z})$, in many of the samples displayed their respective errors are well in line with the dataset averages, or even below, with an exception to the samples 2 ($O/F: 2.61, l_r: 4.23\,\text{mm}, d_c: 9.90\,\text{mm}$) and 6 ($O/F: 2.77, l_r: 5.39\,\text{mm}$, $d_c: 9.51\,\text{mm}$), counting from the left. For sample 2, the models display a higher error than the dataset average for the $\overline{E}_{rel,test}(\overline{T})$, $\overline{E}_{norm,test}(\overline{Y}_{O_2})$, and $\overline{E}_{norm,test}(\overline{Z})$ quantities, whereas for 6 it is only observed for $\overline{E}_{rel,test}(\overline{T})$. Note that these two samples are located at the higher end of the d_c axis, where only a few of HF samples are available for training (DS3 dataset in Figure 2c). Meanwhile, samples 1 ($O/F: 2.73, l_r: 12.95\,\text{mm}, d_c: 5.55\,\text{mm}$), 2, and 7 ($O/F: 2.74, l_r: 13.98\,\text{mm}, d_c: 7.88\,\text{mm}$) display relative large values for $\overline{E}_{norm,test}(\overline{u})$ and $\overline{E}_{norm,test}(u_{RMS})$. In particular, samples 1 and 7 have the largest values of l_r in DS4, close to the design-space edges, while sample 2 has large values of d_c. These regions of the design-space are characterized of having a low density of HF samples. Thus, we can indicate the models performance decreases when traversing the limits of the design-space. Finally, in Figure 11 the LFMs error metrics, calculated over the DS4 test dataset, have been added to show the reader their comparatively poor performance. In all the metrics and samples involved the LFMs largely surpass the MF models error metrics averages (black dots), as expected.

To study the distribution of the error across the different injector regions, four distinct regions, common to the shear-coaxial injectors configurations present in the DS4 dataset, have been identified. These are depicted in Figure 12a. The “Injector” corresponds to the area between the inlets and the oxidizer post lip; the “Recess” denotes the area between the post lip and the injection plane; the “Near Field,” the area between the injection plane and $10d_{ox}$; and the “Chamber,” the remaining fluid area. For each of these regions, the average *local* error metrics, across the test dataset DS4 have been calculated. These metrics are displayed as bars in Figure 12b. To simplify the comparison, the dataset averages of these metrics, are overlaid on each of the bars as dark dashed lines. Except $\overline{E}_{norm,test}(\overline{T})$, all metrics show below-average errors at the “Injector” region. The large relative error of $\overline{T}$ at the injection is possibly because of the fact that the lowest temperatures are seen in this area. Moreover, the “Chamber” zone is also characterized by lower error metrics. This is expected as the field topology in the chamber area is rather uniform, even for the u_{RMS} field, as we previously showed in Figures 8a through 8c. However, the “Recess” and “Near Field” areas show clearly larger than average metrics. This is in line to what was

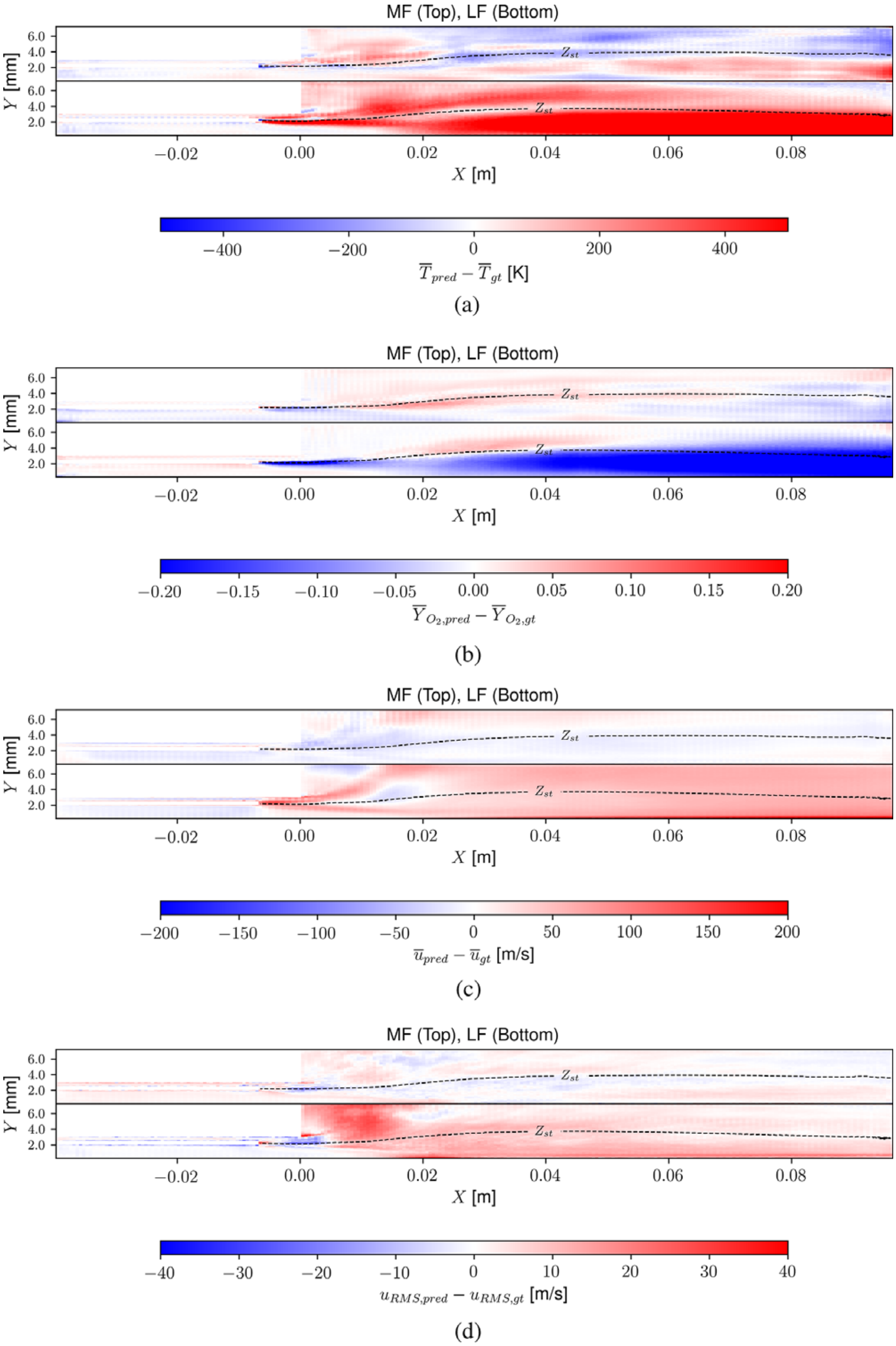


Figure 10. *Prediction errors between predictions (pred), and ground truth (gt), over the selected test sample, for MF (top) and LF (bottom) on the: time-averaged temperature ($\overline{T}$, **a**), oxygen mass fraction ($\overline{Y}_{O_2}$, **b**), and velocity-u ($\overline{u}$, **c**), as well as the velocity-u RMS field (u_{RMS}, **d**). The figures' aspect-ratio has been adjusted to ease visualization.*

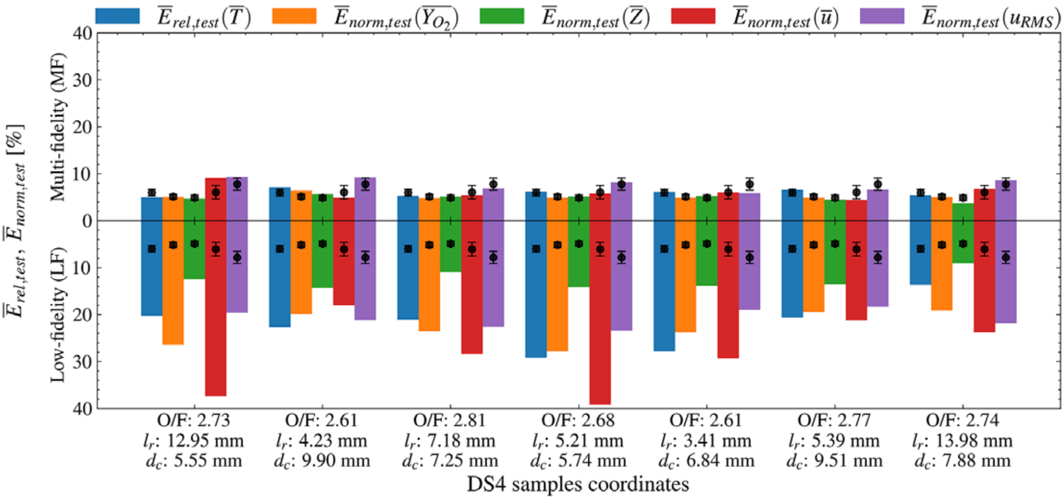


Figure 11. *MF models error metrics per field across the samples of the test-dataset DS4. The error displayed per sample is estimated by averaging the errors from all the models obtained from the six folds conducted during training. The dataset averages of the errors are indicated as black dots superimposed over each bar. Also, the bars for each dot indicate the dispersion of the associated metric over DS4. For reference, the error metrics calculated for the predictions of the LFMs over DS4 have been added.*

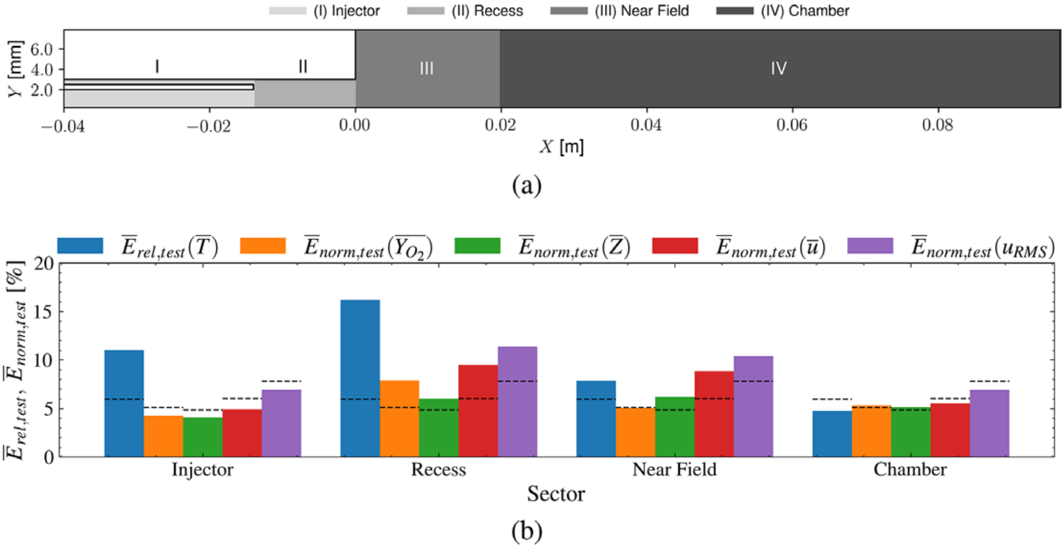


Figure 12. *Localized MF models error metrics: a) Injector representative schematic highlighting the locations of the regions identified. b) Bar plot showing the corresponding error metric, either $\overline{E}_{rel,test}$ or $\overline{E}_{norm,test}$, per field for the different sectors identified.*

detailed in Figures 10a–10d. The more complex local topology in these areas leads to higher local error. Note that, many relevant structures linked to the mixing in shear-coaxial injectors develop in the “Near Field” and “Recess.” Hence, this showcases the potential of directing the learning process through a localized loss function, which ponders the information from these regions. Additionally, a *diffusion flame* based loss, using the mixture fraction field as sensor could be used as to better constrained the loss in the area where the flame is located.

4. Analysis of the features preservation hypothesis

In Section 1.2 of the manuscript, it was stated that the main hypothesis of this work is, that the relationships between inputs and outputs in the HF and LF datasets do not differ significantly, thus, enabling the reuse, in particular, of the feature maps. In rigor, this hypothesis indicates that the feature maps extracted in the encoding layers of the U-Net during the training for the LF task[9], *remain relevant* for the HF task[10]. This is briefly justified by the fact that both networks operate over the same input-space, that is the design space composed of the normalized O/F, l_r, and d_c quantities. Shall both tasks not be largely dissimilar, it is intuitive to think that the features learned in the *source* task, will remain relevant for the *target* task. Thus, we set now to analyze this last statement.

We proceed by defining the functions $\bar{y}^S(\boldsymbol{z},\boldsymbol{h})$ and $\bar{y}^T(\boldsymbol{z},\boldsymbol{h})$,

$$\begin{aligned} \bar{y}^S(\boldsymbol{z},\boldsymbol{h}) &= \frac{1}{S(\boldsymbol{z})}\sum_{p=0}^{N_y}\sum_{q=0}^{N_x}[\boldsymbol{M}(\boldsymbol{z})]_{pq}\left[\boldsymbol{q}^{S,\overline{\mathrm{T}}}(\boldsymbol{z},\boldsymbol{h})\right]_{pq} \\ \bar{y}^T(\boldsymbol{z},\boldsymbol{h}) &= \frac{1}{S(\boldsymbol{z})}\sum_{p=0}^{N_y}\sum_{q=0}^{N_x}[\boldsymbol{M}(\boldsymbol{z})]_{pq}\left[\boldsymbol{q}^{T,\overline{\mathrm{T}}}(\boldsymbol{z},\boldsymbol{h})\right]_{pq} \end{aligned} \tag{10}$$

where $S(\boldsymbol{z}) = \sum_{p=0}^{N_y}\sum_{q=0}^{N_x}[\boldsymbol{M}(\boldsymbol{z})]_{pq}$ is total sum of the mask, defining the fluid region, associated to the design point identified by (normalized) design-space coordinates $\boldsymbol{z} \in \mathcal{D} \in \mathbb{R}^3$. The quantities $[\boldsymbol{q}^{S,\mathrm{T}}(\boldsymbol{z},\boldsymbol{h})]_{pq}$, and $[\boldsymbol{q}^{T,\mathrm{T}}(\boldsymbol{z},\boldsymbol{h})]_{pq}$ correspond to the pq-element of the normalized predicted tensor of the *source*, and *target* U-Nets, respectively. Note that in this exercise we have focused only on the U-Nets which predict the normalized temperature field, hence the *overline*T superscript. Essentially, $\bar{y}^S(\boldsymbol{z},\boldsymbol{h})$, and $\bar{y}^T(\boldsymbol{z},\boldsymbol{h})$ represent the masked average normalized prediction of the average temperature field U-Nets, for the normalized input $\boldsymbol{z}$.

The quantity $\boldsymbol{h} \in \mathbb{H}$ represents the ensemble of encoded "hidden states" of interest, that is the features (outputs) from the encoded layers that remained frozen during the *transfer learning* exercise. These are comprised of the first 6 encoder convolutional layers, as can be in the diagram of Figure 7. For notation convenience, we have stacked all the features (h_j), regardless of their layer of origin, as a single vector such that: $\boldsymbol{h} \in \mathbb{H} \subset \mathbb{R}^{226304}$.

The dimension of the hidden-states space of interest (226304) is considerably large, which hinders a direct study on the relevance of each hidden state, separately. We propose to study the relative behavior of the $\bar{y}^S(\boldsymbol{z},\boldsymbol{h})$ and $\bar{y}^T(\boldsymbol{z},\boldsymbol{h})$ functions w.r.t. to $\boldsymbol{h}$. For such, we sample $\frac{\partial \bar{y}^S}{\partial h_i}\Big|_{\mathbf{z}_j}$ and $\frac{\partial \bar{y}^T}{\partial h_i}\Big|_{\mathbf{z}_j}$, for all $h_i = [\boldsymbol{h}]_i$ and $\boldsymbol{z}_j \in \mathrm{DS1} \cup \mathrm{DS3}$. In this way we can compute statistics of the gradients of both functions across the normalized input (or design)-space, which both U-Nets share. A total of $\sim 21.5 \times 10^6$ different values of $\frac{\partial \bar{y}^S}{\partial h_i}\Big|_{\mathbf{z}_j}$ and $\frac{\partial \bar{y}^T}{\partial h_i}\Big|_{\mathbf{z}_j}$ are computed, for each network, via automatic differentiation.

In Figure 13a we show the histograms of $\frac{\partial \bar{y}^S}{\partial h_i}\Big|_{\mathbf{z}_j}$, on the top, and $\frac{\partial \bar{y}^T}{\partial h_i}\Big|_{\mathbf{z}_j}$, on the bottom. The histograms are normalized so that the area covered equals to 1. The histograms are generated over 5000 bins. A priori, both $\nabla_{\boldsymbol{h}\in\mathbb{H}}\bar{y}^S = \frac{\partial \bar{y}^S}{\partial h_i}\Big|_{\mathbf{z}_j}$ and $\nabla_{\boldsymbol{h}\in\mathbb{H}}\bar{y}^T = \frac{\partial \bar{y}^T}{\partial h_i}\Big|_{\mathbf{z}_j}$ show very similar distributions. Both U-Nets show a peak around ~ 0, which indicates that in most cases the influence of a given feature h_i (or hidden-state coordinate) is small. This hints to the fact that only a small set of the network contributes when generating a prediction, which is potential evidence of parsimony. Moreover, in both cases, the gradients values decay rapidly. This is potentially because of the fact that L2 regularization (weight decay) was used to limit gradient explosion. The most relevant difference between both histograms is thus the larger

[9] *Source* task, following the transfer learning vocabulary.

[10] *Target* task.

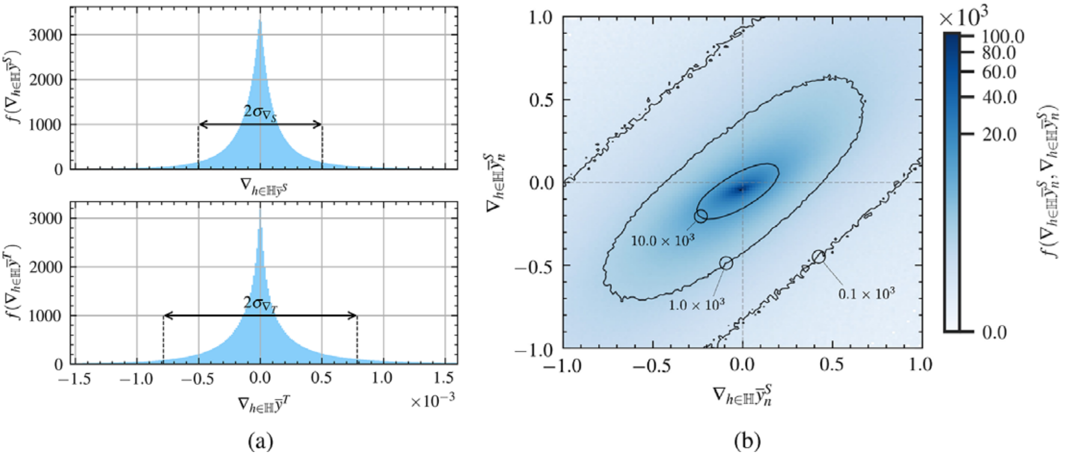


Figure 13. *Mean valued U-Net outputs ($\overline{T}$-models), feature gradients statistics.* ***a)*** *Histogram of gradients on source network, expressed as a probability density, and* ***b)*** *joint histogram for gradients of source network (x-axis) and target network (y-axis). In the latter, both variables have been normalized by their respective statistical average and standard deviations.*

dispersion of values of the gradients on the *target* network, seen by the larger standard deviation of $\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}^T$ (σ_{∇_T}) compared with that of $\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}^S$ (σ_{∇_S}), both drawn in Figure 13a.

Meanwhile, Figure 13b shows a joint-histogram for both quantities: $\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^S$ and $\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^T$. The former is shown on the x-axis, while the latter on the y-axis. The subscript n indicates that both variables shown have been normalized by their respective mean and standard deviations. Note that both distributions in Figure 13a had shown already a mean value of approximately 0. As in Figure 13a, the bi-variate histogram is normalized to have a total "volume" of 1, therefore approximating the probability density function $f\left(\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^S, \nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^T\right)$. Contour values of $f(.,.)\in[100,1{,}000,10{,}000]$ have been overlaid to ease the interpretation of the figure.

Figure 13b shows an interesting phenomenon. The distribution of the gradients calculated on both U-Nets, w.r.t. the reused features, over the shared design-space, appear to be correlated. Furthermore, the oval-shape of the bi-variate distribution, alongside with its alignment on the first quadrant diagonal allow to conclude about the relative behavior of $\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^S$ and $\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^S$. For instance, shall we desire to calculate the probability of a given feature to be relevant for the *source* network prediction, but irrelevant to the *target* network one, we should estimate $P\left(|\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^S|>\Gamma\cap|\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^T|<\epsilon\right)$. Where $\Gamma>>0$, while $\varepsilon\to0$. This equates to integrate $f(,)$ over the vicinity of the $\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^T=0$ line, but for regions with $\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^S>\Gamma$ and $\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^S<-\Gamma$. What we can observe from Figure 13b, is that the probability density rapidly decreases along the $\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^T=0$ direction. Thus, if we took $\Gamma=0.5\sigma_{\nabla_S}$ (the relevant feature on the source network), we can see that the probability density function has approximately decreased by 2 orders of magnitude. The same behavior is attained when we want to assess the probability of a feature being relevant on the *target* network, but irrelevant on the *source* one, that is, $P\left(|\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^S|<\epsilon\cap|\nabla_{\boldsymbol{h}\in\mathbb{H}}\overline{y}_n^T|>\Gamma\right)$.

However, following this probability density map, it appears to be more likely that features that are relevant on the *source* network, to remain relevant on the *target* one. This behavior observed in the bi-variate distribution allows us to partially conclude that the feature set reused, from one network to the other, remains relevant when conducting a prediction for both tasks. Note however, that this is not equal to state that the latent-space (or feature-space) that would be obtained by fine-tuning the encoder layers on the target task, would be the same. We emphasize that the core hypothesis of our work is that the features learned in the source task, remain relevant for the target task, which is what we have attempted to explain, statistically, in Figure 13b.

5. Conclusions & future work

A feature-transfer, network-based transfer learning approach has been put in place to develop data-driven MFSMs for a shear-coaxial rocket engine injector operating on a GOx/Methane mixture. The ultimate goal is to reduce the *offline* data generation cost for rendering suitable predictions. The models aim to predict axial cuts of time-average LES fields, for example, the temperature, oxygen mass-fraction, mixture-fraction, axial velocity component, and axial velocity RMS value. The analysis of mesh-convergence on the reference injector configuration evidences the modeling errors on the under-resolved LES. The coarse mesh solution is overly dissipative, with largely overpredicted heat-release rates, resulting in greater downstream mixing and over-estimated reaction rates. Also, the penetration of the oxygen-jet is under-estimated as a consequence of greater consumption rates.

Two coarse LES datasets, totaling 92 different shear-coaxial injector simulations, are readily available. The Design of Experiments is performed over a 3D design-space comprising the: mixture-ratio (O/F), recess-length of the oxidizer post (l_r), and the chamber radius (d_c). Two additional LES datasets of shear-coaxial injectors, comprised of 12 and 7 samples, are created. The LES simulations are performed over a much finer mesh, albeit at a larger cost per sample.

First, the coarse LES datasets are utilized to train and validate the *source* models of the time-averaged field quantities: temperature, oxygen mass-fraction, mixture fraction, velocity-u component, and velocity-u RMS. Second, the best-performing *source* models are retrained with the refined LES dataset, while freezing the encoding convolutional layers. The resulting MFMs, even though trained on a reduced set of high-resolution LES samples, show qualitative evidence of correcting the *source* models' shortcomings. In effect, when evaluated over a test dataset, the multifidelity emulators predict a longer-flame, thinner hot-gases area, and slower flame expansion. The right penetration of the oxygen jet appears to be recovered, at least for the audited validation sample.

The analysis of the error distribution over the test samples indicates that the MFMs struggle close to the edges of the design-space, where a lower density of HF samples is available for training. Moreover, a study of the local error metrics highlighted that errors larger than average are attained on the recess and near-field areas of the shear-coaxial injectors. This is caused by the relatively more complex topology of the fields in these regions. Future works will address this issue by implementing a local loss term that accents the contribution of these areas, thus locally reinforcing the learning.

Acknowledgments. The authors gratefully acknowledge Région d'Occitanie for co-founding the Ph.D. of J. F. Zapata Usandivaras under the project MODAR. This work was supported by the Chair for Advanced Space Concepts Laboratory (SaCLab2) resulting from the partnership between Airbus Defense and Space, Ariane Group, and ISAE-SUPAERO. This work was granted access to the GENCI HPC resources under the allocation A0152B12992. Finally, financial support was partly provided by the French "Programme d'Investissements d'avenir" ANR-17-EURE0005 conducted by the ANR through the TSAE scholarship.

Author contribution. AU: Conceptualization, funding acquisition, methodology, project administration, resources, supervision, writing (review and editing). MB: Conceptualization, methodology, supervision, writing (review and editing). BC: Conceptualization, methodology, supervision, writing (review and editing). JFZU: Conceptualization, data curation, formal analysis, investigation, methodology, validation, visualization, writing (original draft). All authors approved the final submitted draft.

Competing interest. There are no competing interests from the authors in this publication.

Data availability statement. Data, code, and other materials can be made accessible upon request to the corresponding author.

Funding statement. This research was hosted within the framework of project MODAR (under allocation 20,007,323/ALDOCT-001030); GENIAL (project number 5700007526); and ADMIRE chair (project ID 4000139505/22/NL/MG).

Ethical standard. The research meets all ethical guidelines, including adherence to the legal requirements of the study country.

References

Aversano G, **D'Alessio G**, **Coussement A**, **Contino F and Parente A** (2021) Combination of polynomial chaos and Kriging for reduced-order model of reacting flow applications. *Results in Engineering 10*, 100223. https://doi.org/10.1016/j.rineng.2021.100223.

Aversano G, **Ferrarotti M and Parente A** (2021) Digital twin of a combustion furnace operating in flameless conditions: Reduced-order model development from CFD simulations. *Proceedings of the Combustion Institute 38*(4), 5373–5381. https://doi.org/10.1016/j.proci.2020.06.045.

Bermejo-Barbanoj C, **Moya B**, **Badías A**, **Chinesta F and Cueto E** (2024) Thermodynamics-informed super-resolution of scarce temporal dynamics data. *Computer Methods in Applied Mechanics and Engineering 430*, 117210. https://doi.org/10.1016/j.cma.2024.117210.

Bhalla S, **Yao M**, **Hickey J-P and Crowley M** (2019) Compact representation of a multi-dimensional combustion manifold using deep neural networks, 17.

Biewald L (2020) Experiment tracking with weights and biases. https://www.wandb.com/.

Bilger RW (1989) The structure of turbulent nonpremixed flames. *Symposium (International) on Combustion 22*(1), 475–488. https://doi.org/10.1016/S0082-0784(89)80054-2.

Blanchard S (2021) *Multi-Physics Large-Eddy Simulation of Methane Oxy-Combustion In Liquid Rocket Engines* (Doctoral dissertation). Institut National Polytechnique de Toulouse - INPT. English. ⟨NNT: 2021INPT0079⟩. https://theses.fr/2021INPT0079.

Blanchard S, **Cazeres Q and Cuenot B** (2022) Chemical modeling for methane oxy-combustion in liquid rocket engines. *Acta Astronautica 190*, 98–111. https://doi.org/10.1016/j.actaastro.2021.09.039.

Bouhlel MA, **Hwang JT**, **Bartoli N**, **Lafage R**, **Morlier J and Martins JR** (2019) A Python surrogate modeling framework with derivatives. *Advances in Engineering Software 135*, 102662. https://doi.org/10.1016/j.advengsoft.2019.03.005.

Brunton SL, **Noack BR and Koumoutsakos P** (2020) Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics 52*(1), 477–508. https://doi.org/10.1146/annurev-fluid-010719-060214.

Cai S, **Mao Z**, **Wang Z**, **Yin M and Karniadakis GE** (2021) *Physics-Informed Neural Networks (PINNs) for Fluid Mechanics: A Review* (tech. rep.). *arXiv.*.

Celano MP, **Silvestri S**, **Schlieben G**, **Kirchberger C**, **Haidn OJ**, **Dawson T**, **Ranjan R and Menon S** (2014) Numerical and experimental investigation for a GOX-GCH4 shear-coaxial injector element. In *Space Propulsion Conference,* Cologne, Germany. https://www.researchgate.net/publication/263601323_Numerical_and_Experimental_Investigation_for_a_GOX-GCH4_Shear-Coaxial_Injector_Element.

Chemnitz A, **Sattelmayer T**, **Roth C**, **Haidn O**, **Daimon Y**, **Keller R**, **Gerlinger P**, **Zips J and Pfitzner M** (2018) Numerical investigation of reacting flow in a methane rocket combustor: turbulence modeling. *Journal of Propulsion and Power 34*(4), 864–877. https://doi.org/10.2514/1.B36565.

Chen L, **Cakal B**, **Hu X and Thuerey N** (2021) Numerical investigation of minimum drag profiles in laminar flow using deep learning surrogates. *Journal of Fluid Mechanics 919*. https://doi.org/10.1017/jfm.2021.398.

Cheng P (2017) Transfer learning with convolutional neural networks for classification of abdominal ultrasound images. *J Digit Imaging 30*, 234–243. https://doi.org/10.1007/s10278-016-9929-2.

Choi H-S (2023) Symmetry-informed surrogates with data-free constraint for real-time acoustic wave propagation. *Applied Acoustics*.

Chollet F (2017) *Deep Learning with Python*, 1st Edn. Manning Publications Co.

Chung WT, **Mishra AA**, **Perakis N and Ihme M** (2021) Data-assisted combustion simulations with dynamic submodel assignment using random forests. *Combustion and Flame 227*, 172–185. https://doi.org/10.1016/j.combustflame.2020.12.041.

Cicirello A (2024) Physics-enhanced machine learning: A position paper for dynamical systems investigations (tech. rep.). *arXiv.*.

Cueto E and Chinesta F (2023) *Thermodynamics of learning physical phenomena* (tech. rep. arXiv:2207.12749). arXiv.

Deng J, **Dong W**, **Socher R**, **Li L-J**, **Li K and Fei-Fei L** (2009) ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. https://doi.org/10.1109/CVPR.2009.5206848.

Eldred MS and Dunlavy DM (2012) Formulations for surrogate-based optimization with data fit, multi-fidelity, and reduced-order models. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.* https://doi.org/10.2514/6.2006-7117.

Farimani AB, **Gomes J and Pande VS** (2017) *Deep Learning the Physics of Transport Phenomena* (tech. rep. arXiv:1709.02432). arXiv.

Fernandez-Godino MG (2023) Review of multi-fidelity models. *Advances in Computational Science and Engineering*, *1*(4): 351–400. https://doi.org/10.3934/acse.2023015.

Goswami S, **Bora A**, **Yu Y and Karniadakis GE** (2022) *Physics-Informed Neural Operators* (tech. rep. arXiv:2207.05748). arXiv.

Goswami S, **Jagtap AD**, **Babaee H**, **Susi BT and Karniadakis GE** (2023) *Learning Stiff Chemical Kinetics Using Extended Deep Neural Operators* (tech. rep. arXiv:2302.12645). arXiv.

Guo X, **Li W and Iorio F** (2016) Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 481–490. https://doi.org/10.1145/2939672.2939738.

Hasegawa K, **Fukami K**, **Murata T and Fukagata K** (2020) Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. *Theoretical and Computational Fluid Dynamics 34*(4), 367–383. https://doi.org/10.1007/s00162-020-00528-w.

He K, **Zhang X**, **Ren S and Sun J** (2016) Deep residual learning for image recognition. In 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90.

Hemmasian A and Barati Farimani A (2023) Reduced-order modeling of fluid flows with transformers. *Physics of Fluids 35*(5), 057126. https://doi.org/10.1063/5.0151515.

Hernandez Q, **Badias A**, **Gonzalez D**, **Chinesta F and Cueto E** (2021) Deep learning of thermodynamics-aware reduced-order models from data. *Computer Methods in Applied Mechanics and Engineering 379*, 113763. https://doi.org/10.1016/j.cma.2021.113763.

Hu Z, **Daryakenari NA**, **Shen Q**, **Kawaguchi K and Karniadakis GE** (2024). *State-Space Models are Accurate and Efficient Neural Operators for Dynamical Systems* (tech. rep. arXiv:2409.03231). arXiv.

Hwang JT and Martins JRRA (2018) A fast-prediction surrogate model for large datasets. *Aerospace Science and Technology 75*, 74–87. https://doi.org/10.1016/j.ast.2017.12.030.

Jeon J, **Lee J**, **Vinuesa R**, **Kim SJ** (2024). Residual-based physics-informed transfer learning: A hybrid method for accelerating long-term CFD simulations via deep learning. *International Journal of Heat and Mass Transfer 220*. https://doi.org/10.1016/j.ijheatmasstransfer.2023.124900.

Jin R, **Chen W and Sudjianto A** (2005) An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference 134*(1), 268–287. https://doi.org/10.1016/j.jspi.2004.02.014.

Karniadakis GE, **Kevrekidis IG**, **Lu L**, **Perdikaris P**, **Wang S and Yang L** (2021) Physics-informed machine learning. *Nature Reviews Physics*, *3*(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5.

Kim Y, **Choi Y**, **Widemann D and Zohdi T** (2022) A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. *Journal of Computational Physics 451*, 110841. https://doi.org/10.1016/j.jcp.2021.110841.

Kingma DP and Ba J (2014) Adam: A method for stochastic optimization. *CoRR, abs/1412.6980*.

Krügener M, **Zapata Usandivaras JF**, **Bauerheim M and Urbano A** (2022) Coaxial-injector surrogate modeling based on Reynolds averaged Navier stokes simulations using deep learning. *Journal of Propulsion and Power 38*(5), 783–798. https://doi.org/10.2514/1.B38696.

Lapenna PE, **Amaduzzi R**, **Durigon D**, **Indelicato G**, **Nasuti F and Creta F** (2018) Simulation of a single-element GCH4/GOx rocket combustor using a non-adiabatic flamelet method. In 2018 *Joint Propulsion Conference.* American Institute of Aeronautics; Astronautics. https://doi.org/10.2514/6.2018-4872.

Lapeyre CJ, **Misdariis A**, **Cazard N**, **Veynante D and Poinsot T** (2019) Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame 203*, 255–264. https://doi.org/10.1016/j.combustflame.2019.02.019.

Lax P and Wendroff B (1960) Systems of conservation laws. *Communications on Pure and Applied Mathematics 13*(2), 217–237. https://doi.org/10.1002/cpa.3160130205.

Le Clainche S, **Ferrer E**, **Gibson S**, **Cross E**, **Parente A and Vinuesa R** (2023) Improving aircraft performance using machine learning: A review. *Aerospace Science and Technology 138*, 108354. https://doi.org/10.1016/j.ast.2023.108354.

Li X, **Pang T**, **Xiong B**, **Liu W**, **Liang P and Wang T** (2017) Convolutional neural networks based transfer learning for diabetic retinopathy fundus image classification. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Shanghai, China, pp. 1–11. https://doi.org/10.1109/CISP-BMEI.2017.8301998.

Li Z, **Kovachki N**, **Azizzadenesheli K**, **Liu B**, **Bhattacharya K**, **Stuart A and Anandkumar A** (2021) *Fourier Neural Operator for Parametric Partial Differential Equations* (tech. rep.). arXiv.

Liu D and Wang Y (2019) Multi-Fidelity Physics-Constrained Neural Network and its Application in Materials Modeling. In *Proceedings of the ASME 2019 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Volume *2A*: 45th Design Automation Conference. Anaheim, California, USA. August 18–21, 2019. V02AT03A007. ASME. https://doi.org/10.1115/DETC2019-98115.

Lu L, **Jin P**, **Pang G**, **Zhang Z and Karniadakis GE** (2021) Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence 3*(3), 218–229. https://doi.org/10.1038/s42256-021-00302-5.

Maestro D, **Cuenot B and Selle L** (2019) Large eddy simulation of combustion and heat transfer in a single element gch 4/gox rocket combustor. *Flow, Turbulence and Combustion 103*(3), 699–730.

Malik MR, **Obando Vega P**, **Coussement A and Parente A** (2021) Combustion modeling using Principal Component Analysis: A posteriori validation on Sandia flames D, E and F. *Proceedings of the Combustion Institute 38*(2), 2635–2643. https://doi.org/10.1016/j.proci.2020.07.014.

McKay MD, **Beckman RJ and Conover WJ** (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics 21*(2), 239–245.

Mendez MA, **Ianiro A**, **Noack BR and Brunton SL** (eds.) (2023) *Data-Driven Fluid Mechanics: Combining First Principles and Machine Learning*, 1st Edn. Cambridge University Press. https://doi.org/10.1017/9781108896214.

Minisci E and Vasile M (2013) Robust design of a reentry unmanned space vehicle by multifidelity evolution control. *AIAA Journal 51*(6), 1284–1295. https://doi.org/10.2514/1.J051573.

Nicoud F, **Toda HB**, **Cabrit O**, **Bose S and Lee J** (2011) Using singular values to build a subgrid-scale model for large eddy simulations. *Physics of Fluids 23*(8), 085106. https://doi.org/10.1063/1.3623274.

Pan SJ and Yang Q (2010) A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering 22*(10), 1345–1359. https://doi.org/10.1109/TKDE.2009.191.

Parente A, **Swaminathan N** (2024). Data-driven models and digital twins for sustainable combustion technologies. *iScience 27*(4). https://doi.org/10.1016/j.isci.2024.109349.

Perakis N, **Celano MP and Haidn OJ** (2017) Heat flux and temperature evaluation in a rectangular multi-element GOX/GCH4 combustion chamber using an inverse heat conduction method. In *7th European Conference in for Aeronautics and Aerospace Sciences* (EUCASS), Milan, Italy. https://www.eucass.eu/doi/EUCASS2017-366.pdf.

Poinsot T and Lelef S (1992) Boundary conditions for direct simulations of compressible viscous flows. *Journal of Computational Physics 101*(1), 104–129. https://doi.org/10.1016/0021-9991(92)90046-2.

Regazzoni F, **Pagani S**, **Salvador M**, **Dede' L and Quarteroni A** (2024) Learning the intrinsic dynamics of spatio-temporal processes through latent dynamics networks. *Nature Communications 15*(1), 1834. https://doi.org/10.1038/s41467-024-45323-x.

Ribani R and Marengoni M (2019) A Survey of transfer learning for convolutional neural networks. In 2019 *32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, 47–57. https://doi.org/10.1109/SIBGRAPI-T.2019.00010.

Schonfeld T and Rudgyard M (1999) Steady and unsteady flow simulations using the hybrid flow solver avbp. *AIAA Journal 37* (11), 1378–1385. https://doi.org/10.2514/2.636.

Schroeder W, **Martin K and Lorensen B** (2006) *The Visualization Toolkit*, 4th Edn. Kitware.

Solera-Rico A, **Sanmiguel Vila C**, **Gómez-López M**, **Wang Y**, **Almashjary A**, **Dawson ST M and Vinuesa R** (2024) β-Variational autoencoders and transformers for reduced-order modelling of fluid flows. *Nature Communications 15*(1), 1361. https://doi.org/10.1038/s41467-024-45578-4.

Stimpson CJ, **Ernst CD**, **Knupp P**, **Pébay PP and Thompson D**. (2007). *The Verdict Library Reference Manual* (tech. rep.). Sandia National Laboratories.

Swaminathan N and Parente A (2023) Machine learning and its application to reacting flows: ML and combustion. Springer International Publishing. https://doi.org/10.1007/978-3-031-16248-0.

Tan C, **Sun F**, **Kong T**, **Zhang W**, **Yang C and Liu C** (2018) A survey on deep transfer learning. In Kurková V, Manolopoulos Y, Hammer B, Iliadis L and Maglogiannis I (eds.), *Artificial Neural Networks and Machine Learning – Icann 2018*. Springer International Publishing, pp. 270–279. https://doi.org/10.1007/978-3-030-01424-7_27.

Thuerey N, **Weissenow K**, **Prantl L and Hu X** (2020) Deep learning methods for reynolds-averaged navier-stokes simulations of airfoil flows. *AIAA Journal 58*(1), 25–36. https://doi.org/10.2514/1.j058291 Comment: Code and data available at: https://github.com/thunil/Deep-Flow-Prediction.

Van Driest ER (1951) Turbulent boundary layer in compressible fluids. *Journal of the Aeronautical Sciences 18*(3), 145–160. https://doi.org/10.2514/8.1895.

Vinuesa R and Brunton SL (2021) *The Potential of Machine Learning to Enhance Computational Fluid Dynamics* (tech. rep. arXiv:2110.02085).

Wang Y, **Solera-Rico A**, **Vila CS and Vinuesa R** (2024) Towards optimal β-variational autoencoders combined with transformers for reduced-order modelling of turbulent flows. *International Journal of Heat and Fluid Flow 105*. https://doi.org/10.1016/j.ijheatfluidflow.2023.109254.

Wang Z, **Meng X**, **Jiang X**, **Xiang H and Karniadakis GE** (2023) *Solution Multiplicity and Effects of Data and Eddy Viscosity on Navier-Stokes Solutions Inferred by Physics-Informed Neural Networks* (tech. rep. arXiv:2309.06010). arXiv.

White C, **Ushizima D and Farhat C** (2019) *Fast Neural Network Predictions from Constrained Aerodynamics Datasets* (tech. rep. arXiv:1902.00091). arXiv.

Xing V, **Lapeyre CJ**, **Jaravel T and Poinsot T** (2021) Generalization capability of convolutional neural networks for progress variable variance and reaction rate subgrid-scale modeling. *Energies 14*(16), 5096. https://doi.org/10.3390/en14165096.

Yu J and Hesthaven JS (2019) Flowfield reconstruction method using artificial neural network. *AIAA Journal 57*(2), 482–498. https://doi.org/10.2514/1.J057108.

Zapata Usandivaras JF, **Urbano A**, **Bauerheim M and Cuenot B** (2022) Data driven models for the design of rocket injector elements. *Aerospace 9*(10). https://doi.org/10.3390/aerospace9100594.

Zdybal K, **Sutherland J and Parente A** (2022) Manifold-informed state vector subset for reduced-order modeling. *Proceedings of the Combustion Institute 39*, 1–10. https://doi.org/10.1016/j.proci.2022.06.019.

Zhang Y, **Sung W.-J. and Mavris D** (2018) *Application of Convolutional Neural Network to Predict Airfoil Lift Coefficient* (tech. rep. arXiv:1712.10082). arXiv.

Zhao Z, **Ding X and Prakash BA** (2024) *PINNsFormer: A Transformer-Based Framework For Physics-Informed Neural Networks* (tech. rep. arXiv:2307.11833). arXiv.

Zips J, **Müller H and Pfitzner M** (2017) Non-adiabatic tabulation methods to predict wall-heat loads in rocket combustion. In *55th AIAA Aerospace Sciences Meeting.* American Institute of Aeronautics; Astronautics. https://doi.org/10.2514/6.2017-1469.

Cite this article: Zapata Usandivaras JF, Bauerheim M, Cuenot B and Urbano A (2025). Data-driven multifidelity surrogate models for rocket engines injector design. *Data-Centric Engineering*, 6, e2. doi:10.1017/dce.2024.56