Fast and accurate field reconstruction of Thermal-Large Eddy Simulation (T-LES) by Deep Learning

Yanis ZATOUT^{1,2*}, Adrien TOUTANT^{1}, Onofrio SEMERARO^{2}, Lionel MATHELIN^{2}, Françoise BATAILLE^{1}

^{1}PROMES-CNRS (UPR 8521), Université de Perpignan Via Domitia
Rambla de la thermodynamique, 66100 Perpignan (France)
^{2} LISN (UMR 9015), CNRS, Université Paris-Saclay
91405 Orsay (France)
*(Corresponding author: yanis.zatout@cnrs.fr)

Abstract - In this paper, we examine a machine learning-based method aimed at improving the accuracy of T-LES fields in the context of highly anisothermal flows. We compare this method with an already existing super-resolution method. We train our convolutional neural network by filtering Direct Numerical Simulation (DNS) snapshots into T-LES ones, and optimize our network to reconstruct DNS small scales from T-LES snapshots. Our results show that the neural network outperforms the classical reconstruction method in terms of the quality of the reconstructed coherent structures, but ends up increasing the Root Mean Square (RMS) values over the DNS ones.

Nomenclature

T	temperature, K
$\bar{\cdot}$	large eddy simulation filter
h	half height of the canal, m
Greek symbols	
Δ_i	mesh length in the i direction, m
θ	neural network weights
Index and exponent	
f	fine mesh
c	coarse mesh
r	reconstructed field

1. Introduction

Solar receivers used in gas-pressurized concentrated solar power plants are subject to extremely high temperatures and heat fluxes, as well as intense turbulence and asymmetrical heating. These conditions present a significant challenge when it comes to the simulation of the system using Direct Numerical Simulation (DNS) due to the large size of the receiver compared to the smallest turbulent structures. Thermal Large Eddy Simulation (T-LES) is a good alternative to overcome this challenge. T-LES models the effect of the smallest turbulence structures and reduces computational costs. However, the high heat fluxes present in solar receivers can affect the turbulence, and existing sub-grid models developed for isothermal or weakly anisothermal flows may not be suitable under such operating conditions [1]. Some types of sub-grid closure models use convolutions to filter local quantities like the scale similarity Bardina *et al.* model [2]. T-LES inherently does not contain enough information for a precise assessment of simulation quantities. To predict thermomechanical constraints within a solar receiver, it is crucial to enhance the accuracy of the filtered data by inverting the T-LES filter and enabling the recovery of RMS temperature values and RMS heat flux values, which are important for understanding the thermal behavior of the system. Some Deep Learning (DL)

https://doi.org/10.25855/SFT2023-087

approaches exist, most notably Bode *et al.* [3], Fukami *et al.* [4], Kim *et al.* [5], all with different types of convolutional neural networks (CNN) architectures to reconstruct the small scales from LES simulations. These techniques aim to be included in CFD solvers to more accurately estimate simulation quantities like velocity fields, or passive scalar quantities in reactive flows. In this work, we will adopt a neural network-based approach and revisit the architecture used in Lapeyre *et al.* [6] by adapting it and learning to reconstruct fine scale temperature fields. We will compare the results of the neural network-based reconstruction against the scale similarity method developed by Stolz and Adams [7].

2. Supporting data

The simulation used for this work assumes a constant Prandtl number of $Pr = 0.76$. The flow is charactarized by its friction Reynolds number defined as

$$Re_{\tau,w} = \frac{U_\tau h}{\nu_w}, \tag{1}$$

where h is the half-height of the channel, ν_w the wall kinematic viscosity and U_τ the friction velocity. In our anisothermal channel flow, the two walls have different friction Reynolds numbers. We define the mean friction Reynolds number at the hot and cold sides as

$$Re_\tau = \frac{1}{2}\Big(Re_{\tau,1} + Re_{\tau,2}\Big). \tag{2}$$

We consider the DNS data generated in Dupuy *et al.* [8] for a channel flow at $Re_\tau = 180$, periodic both in the streamwise (x) and spanwise (z) directions, with lower and upper wall temperatures fixed at $T_1 = 293\ K$ and $T_2 = 586\ K$; we filter the data using a top-hat filter. The mesh is regular both in the streamwise and spanwise directions, and grows in size according to a hyperbolic tangent law along the wall-normal direction. We use cubic spline interpolation to coarsen the DNS temperature data T to the LES grid. This operation shrinks the data from a $384 \times 266 \times 384$ grid to a $48 \times 50 \times 48$ grid. We then use a top-hat filter that spreads over 3 cells in each direction, and re-interpolate the filtered temperature field $\overline{T}^c$ to the DNS mesh $\overline{T}^f$. The following sketch summarizes the work flow

$$T^f \xrightarrow{\text{Coarsening}} T^c \xrightarrow{\text{Filtering}} \overline{T}^c \xrightarrow[\text{interpolation}]{\text{Cubic spline}} \overline{T}^f \xrightarrow{\text{Reconstruction method}} T_r. \tag{3}$$

The top-hat filter approximation for 3 cells of a given quantity ϕ is given by

$$\overline{\phi}(x_i, y_k, z_j, t) = \frac{1}{9(y_{k+1} - y_{k-1})} \sum_{i',j',k'=i-1,j-1,k-1}^{i+1,j+1,k+1} \phi(x_{i'}, y_{k'}, z_{j'}, t)(y_{k'+1} - y_{k'}). \tag{4}$$

This filter is weighted along the y axis because grid points in the wall-normal direction follow a hyperbolic tangent transformation.

3. Common reconstruction methods

We test our approach against an already existing method developed by Adams *et al.* [7, 9, 10]. This method uses an iterative deconvolution procedure based on the Van Cittert method which we now succinctly discuss. The inverse of a convolution filter G assumed invertible, writes

$$G^{-1} = (\mathcal{I} - (\mathcal{I} - G))^{-1}, \tag{5}$$

$$= \lim_{p \to \infty} \sum_{i=0}^{p} (\mathcal{I} - G)^i. \tag{6}$$

We take a $p = 6$ approximation to this converging Neumann series (Stolz and Adams [7] recommend $p = 5$).

3.1. Reconstruction task in the context of our data

There are multiple ways of reconstructing using the above mentioned method. We kept two of them:

- Reconstructing on the coarse grid, then interpolating

$$T \xrightarrow{\text{Coarsening}} T^c \xrightarrow{\text{Filtering}} \overline{T}^c \xrightarrow{\text{Reconstruction method}} T_r^c \xrightarrow[\text{interpolation}]{\text{Cubic spline}} T_r^f. \tag{7}$$

- Interpolating then reconstructing on the fine grid

$$T \xrightarrow{\text{Coarsening}} T^c \xrightarrow{\text{Filtering}} \overline{T}^c \xrightarrow[\text{interpolation}]{\text{Cubic spline}} \overline{T}^f \xrightarrow{\text{Reconstruction method}} T_r. \tag{8}$$

A visual representation using real data is provided in Figure 1.

In the case of process 8, because of the filtering and interpolation processes, we are unable to estimate an accurate filter size on the DNS mesh. On the other hand, we know the filter size on the coarse mesh. The T-LES mesh was chosen so that the homogeneous directions (x) and (z) are 8 times coarser on the T-LES, but the mesh coarsening coefficient in the wall normal direction (y) is not constant. Because of this, we have to estimate the filter size in this direction. This coarsening coefficient fluctuates between 5.2 and 6.1.

4. Deep Learning method

To improve upon the conditioning of the learning task, we make the neural network $f_{NN}(\cdot, \theta)$ learn the residual, and not the reconstruction itself

$$T_r = \overline{T}^f + f_{NN}(\overline{T}^f; \theta), \tag{9}$$

with θ the network weights. The architecture is detailed in Lapeyre *et al.* [6].

4.1. Data acquisition

As discussed in Section 2, DNS simulation data are sampled every $\Delta_t^+ = 7.76 \times 10^{-3}$. The training set is composed of 13 snapshots, and the validation set of 4 snapshots for a total of 17. These snapshots are taken into a contiguous interval $t_{max}^+ - t_{min}^+ = 1.32 \times 10^{-1}$ for training and validation. Full convergence of statistics is achieved prior to taking snapshots, which are spaced out in time to ensure that they are uncorrelated.

4.2. Learning procedure

We perform a learning procedure where data are taken from a fixed height in the canal close to the hot boundary, with a crop height of 16 pixels.

On this fixed height, we randomly sample one subdomain per training snapshot at each step of the mini-batch training. These subdomains are of size $16 \times 16 \times 16$, and we do this 500 times per epoch. This training procedure is done on the coarse mesh to have a similar structure as in process 7. The loss function is defined as

$$\mathcal{L}(T, T_r) = \sqrt{\sum_{i=1}^{N} (T[i] - T_r[i])^2},$$

where $[i]$ is the i^{th} training example, and N the total number of patches in the current batch, results are shown with $N = 13$.

We optimize our neural network using a stochastic gradient optimization technique (ADAM, Kingma and Ba [11]) for a total of 150 epochs with a learning rate fixed at 10^{-2} at the beginning and decreasing by 20% each 10 epoch. Data are augmented via random 90° flips and mirror operations, similarly as Lapeyre *et al.* [6]. We do not allow flips and mirroring that would change the asymmetry of the wall-normal direction. The training takes approximately 20 minutes on an Nvidia RTX A3000 GPU.

5. Results

In the following section, we will compare temperature profiles, RMS profiles $\sqrt{\langle T'^2 \rangle} = \sqrt{\langle T^2 \rangle - \langle T \rangle^2}$, where $\langle \cdot \rangle$ is the spacial averaging over periodic directions (x) and (z), and probability distributions functions (PDF) as obtained by running the DNS, LES and reconstruction methods for a given snapshot. This snapshot is taken from the same simulation as the training snapshots, and is filtered in the same way as is explained in Section 2, but at a much earlier point in the simulation, at $t^+_{\text{train}} - t^+_{\text{test}} = 25.36$ before the training snapshots. Mean temperature profiles correspond to the mean of the temperature over the periodic directions.

The filter and the cubic spline interpolation do not affect the mean temperature profile, on the other hand the profile of the RMS $\sqrt{\langle T'^2 \rangle}$, plotted in Figure 2 shows that the LES filter under–predicts the RMS values of the DNS temperature field. Visual slices of the domain are taken around the point of maximum RMS on planes parallel to the walls.

5.1. Scale similarity method

As discussed in Section 3.1, we consider two methods for the scale similarity, labeled *coarse* and *fine*, for the reconstruction respectively performed on the coarse grid (with the correct LES filter size), and the fine grid (for the estimated LES filter size). In Figure 3, we can see the RMS quantities closely line up with the DNS ones. Figure 4, on the two right most columns, demonstrates qualitatively the 2 types of reconstruction against the DNS and the LES. One can clearly see that the scale similarity method adds smaller structures. Finally, in Figure 5, at the slice where RMS is maximum along the wall-normal direction, the probability density function of the reconstructed temperature manages to get close to the DNS ones.

5.2. Machine learning method

The result of the learning procedure described in Section 4.2 shows the neural network's loss converges around 150 epochs as seen in Figure 6. While the mean temperature profile closely lines up with the DNS one, we can see in Figure 7 the neural network adds too much detail to the reconstructed snapshots the further we are from the boundary. This can also be qualitatively appreciated in Figure 4, although the added structures are the same size as the DNS ones, unlike the scale similarity. Finally, in Figure 5, it can be observed that the CNN aligns almost perfectly with the DNS, indicating its ability to accurately reproduce the temperature distribution.

6. Conclusion

In this work, we demonstrated the capability of a neural network residual-based formulation to mimic DNS fields from LES ones in terms of statistics and visual evaluation of the reconstructed fields and have compared its performances against an established reconstruction method. Results show that the scale similarity method works to recover statistics such as the RMS quantities or the PDF, but somewhat fails in qualitative assessments. On the other hand, the CNN shows good results with respect to the qualitative assessment and reasonable results on the RMS profile. The machine learning approach has yielded promising results. Future work will aim at reconstructing the whole domain, not just a fixed-height sub-domain, learn over distribution of outputs, as opposed to learning a point to point image, and finally, be trained to accurately invert unknown filters from simulated T-LES, not just filtered DNS snapshots.

Acknowledgements

The authors acknowledge the funding received from the Agence nationale de la recherche (ANR) under the SOLAIRE ANR project (ANR-21-CE50-0031). This work was granted access to the HPC resources of CINES under the allocations 2017-A0022A05099 2018-A0042A05099 and 2022-A0112A05099 made by GENCI.

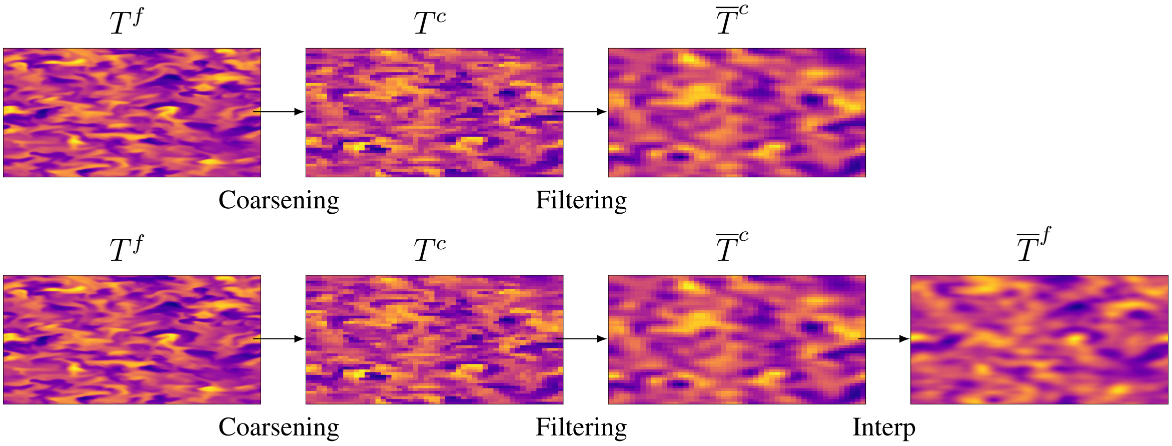

Figure 1: *Data preparation process for reconstruction algorithm assessment. The top process corresponds to equation 8, and the bottom one corresponds to equation 7.*

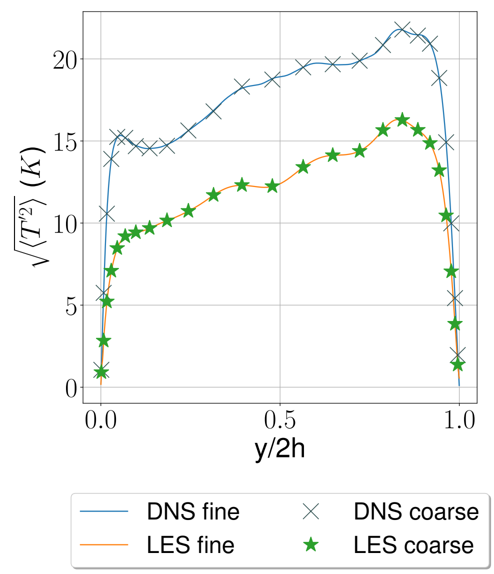

Figure 2: *RMS temperature profiles of T^f, $\overline{T}^f$, T^c and $\overline{T}^c$*

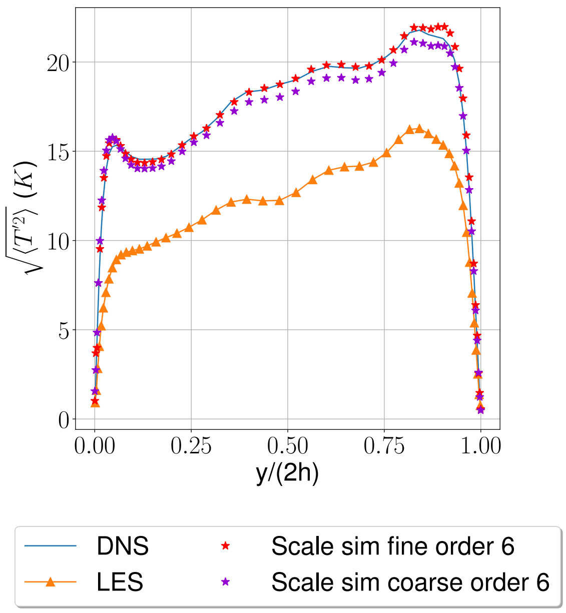

Figure 3: *RMS temperature profiles of T^f, $\overline{T}^f$, T_r scale sim coarse, and T_r scale sim fine*

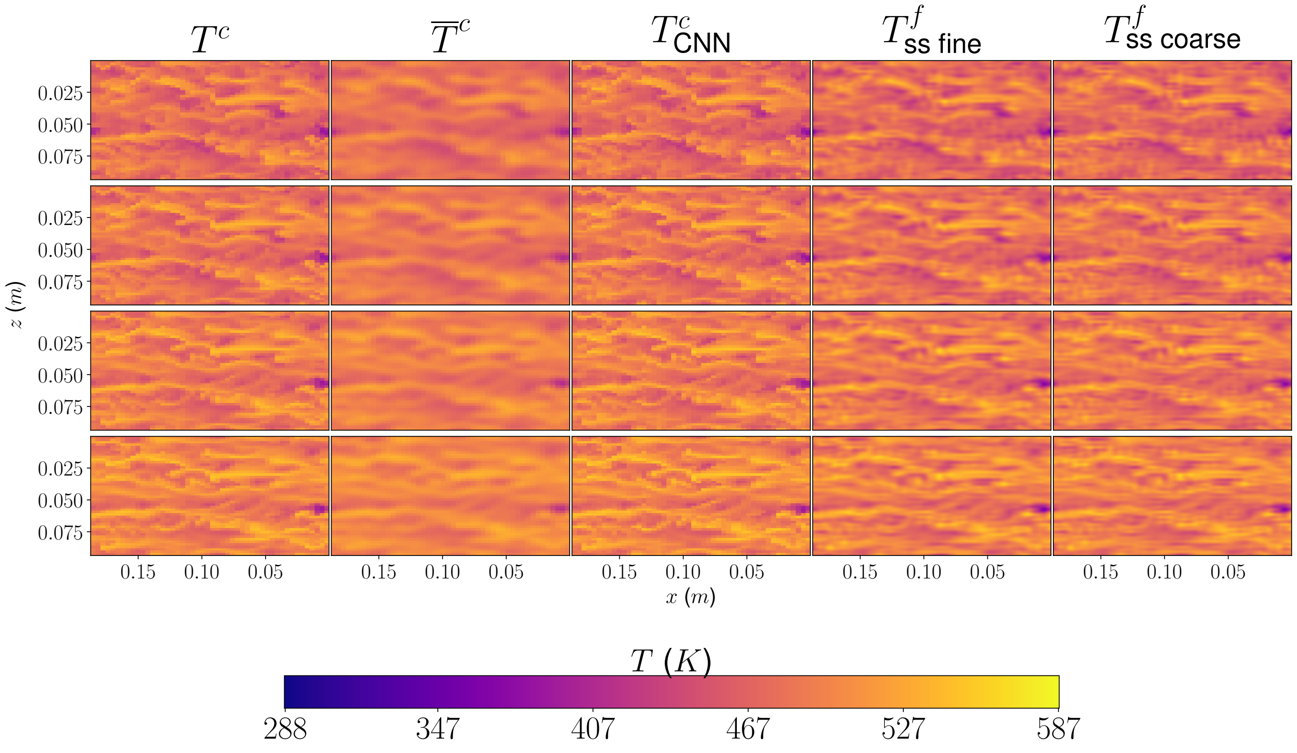

Figure 4: *Qualitative assessment of reconstruction with T^c, $\overline{T}^c$, T^c_r neural network , T^f_r scale sim fine, and T^f_r scale sim coarse*

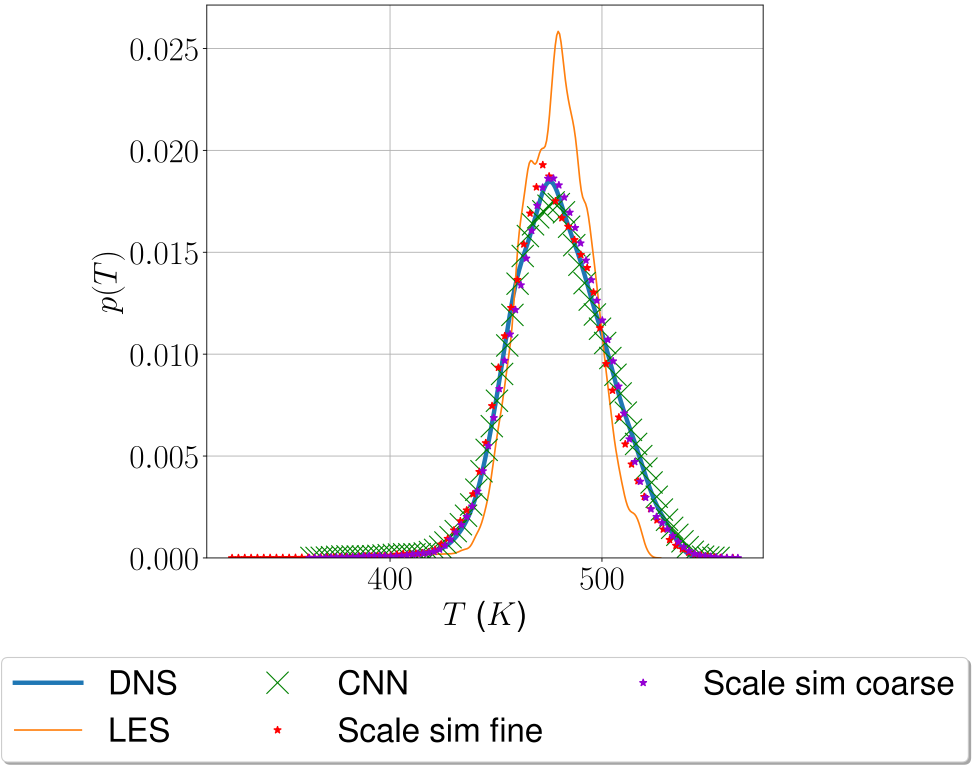

Figure 5: *PDF of T^c, $\overline{T}^c$, T^c_r neural network , T^f_r scale sim fine, and T^f_r scale sim coarse*

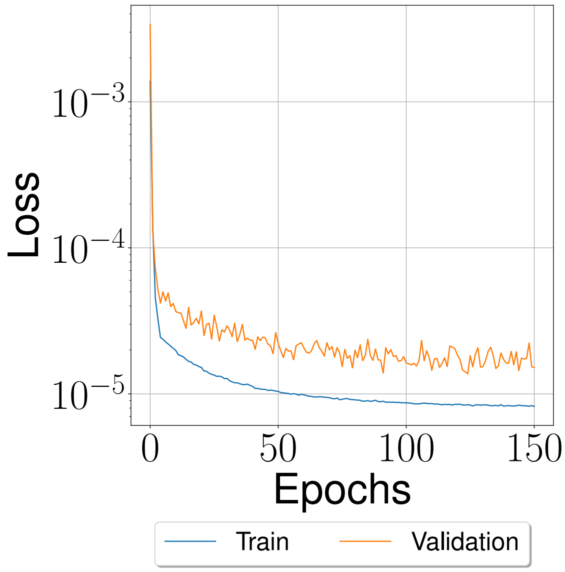

Figure 6: *Normalized training loss*

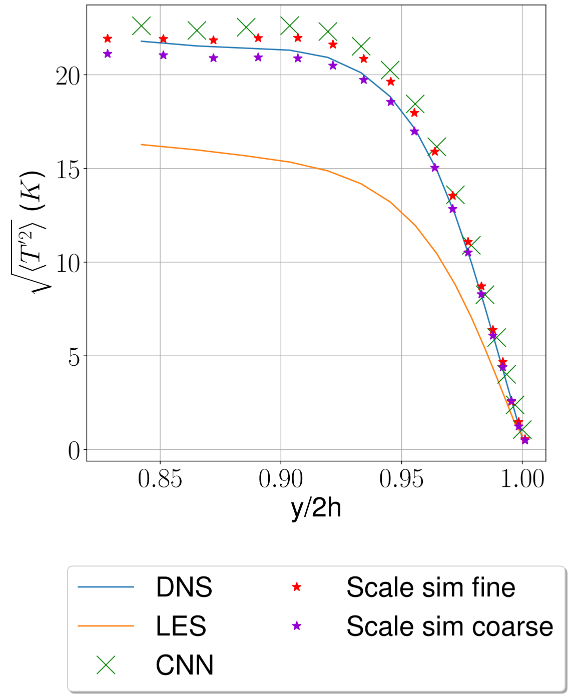

Figure 7: *RMS profile with T^c, $\overline{T^c}$, T^c_r neural network , T^f_r scale sim fine, and T^f_r scale sim coarse*

References

[1] D. Dupuy, A. Toutant, and F. Bataille. A priori tests of subgrid-scale models in an anisothermal turbulent channel flow at low mach number. *International Journal of Thermal Sciences*, 145:105999, 2019. ISSN 1290-0729. doi: https://doi.org/10.1016/j.ijthermalsci.2019.105999. URL `https://www.sciencedirect.com/science/article/pii/S1290072918306719`.

[2] J. Bardina, J. Ferziger, and W. Reynolds. Improved subgrid-scale models for large-eddy simulation. In *13th Fluid and Plasma Dynamics Conference*, Fluid Dynamics and Co-Located Conferences. American Institute of Aeronautics and Astronautics, 1980.

[3] M. Bode, M. Gauding, K. Kleinheinz, and H. Pitsch. Deep learning at scale for subgrid modeling in turbulent flows, 2019. URL `https://arxiv.org/abs/1910.00928`.

[4] K. Fukami, K. Fukagata, and K. Taira. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *Journal of Fluid Mechanics*, 909:A9, 2021. doi: 10.1017/jfm.2020.948.

[5] H. Kim, J. Kim, S. Won, and C. Lee. Unsupervised deep learning for super-resolution reconstruction of turbulence. *Journal of Fluid Mechanics*, 910:A29, 2021. doi: 10.1017/jfm.2020.1028.

[6] C. J. Lapeyre, A. Misdariis, N. Cazard, D. Veynante, and T. Poinsot. Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame*, 203:255–264, May 2019. doi: 10.1016/j.combustflame.2019.02.019. URL `https://hal.archives-ouvertes.fr/hal-02072920`.

[7] S. Stolz and N.A. Adams. An approximate deconvolution procedure for large-eddy simulation. *Phys. Fluids*, 11(7):1699–1701, 1999.

[8] D. Dupuy, A. Toutant, and F. Bataille. Study of the large-eddy simulation subgrid terms of a low Mach number anisothermal channel flow. *International Journal of Thermal Sciences*, 135:221–234, 2019. ISSN 1290-0729.

[9] R von Kaenel, N.A Adams, L Kleiser, and J.B Vos. The approximate deconvolution model for large-eddy simulation of compresible flows with finite volume schemes. *J. Fluids Engng*, 125:375–381, 2003.

[10] R. von Kaenel, N.A. Adams, L. Kleiser, and J.B. Vos. Effect of artificial dissipation on large-eddy simulation with deconvolution modeling. *AIAA J*, 41(8):1606–1609, 2003.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.