Weakly supervised learning for a priori reconstruction of Thermal Large Eddy Simulations (T-LES) using two-point correlations

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 work, we develop a deep learning-based method aimed at reconstructing Thermal Large Eddy Simulations (T-LES) data in flows with high temperature gradients, in a weakly supervised manner. We compare our method with an already existing super-resolution method. We train our neural network from filtered Direct Numerical Simulation (DNS) fields, and we make our network learn to reconstruct the statistics of the flow, using two point correlations. The neural network is demonstrated to reconstruct the statistics of the flow, including the two point correlation.

Nomenclature

T	temperature, K
$\bar{\cdot}$	large eddy simulation filter
C_2	two point correlation
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

Gas-pressurized solar receivers used in concentrated solar power plants are characterized by extremely high temperatures and heat fluxes, as well as intense turbulence and asymmetrical heating. Modeling techniques for such flows include Direct Numerical Simulation (DNS), but because of the difference in size between the receiver and the smallest scales of turbulence, this modeling technique is prohibitively expensive. On the other hand, Thermal Large Eddy Simulation (T-LES) is a good alternative. T-LES only simulates the large structures, and models the effects of the small structures. Nevertheless, turbulence in solar receivers may be impacted by the high heat fluxes, making the sub-grid models designed for isothermal or weakly anisothermal flows inaccurate for these conditions [1]. The proposed deep learning architecture is fully convolutional. Some types of T-LES closure models use convolutions to filter local quantities like the scale similarity Bardina *et al.* model [2]. Because T-LES does not simulate all turbulent scales, it does not contain information to have a precise assessment of simulation quantities, and is thus imprecise. 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, which are important for understanding the thermal behavior of the system. Some Deep Learning (DL) approaches exist, most notably Bode *et al.* [3], Fukami *et al.* [4], Kim *et al.* [5], all with different types

https://doi.org/10.25855/SFT2024-051

of convolutional neural networks (CNN) architectures to reconstruct the small scales from LES simulations, all with supervised algorithms. These techniques aim to be included in Computational Fluid Dynamics solvers to better estimate simulation quantities like velocity fields, or passive scalar quantities in reactive flows. In this work, we adopt a convolutional neural network-based approach and extent the architecture proposed by Lapeyre *et al.* [6], adapting it and learning to reconstruct unfiltered data, more specifically second order statistics, using a two point correlation cost function. We will then compare the results of the neural network-based reconstruction against the scale similarity method developed by Stolz and Adams [7].

2. Supporting data

This work uses a DNS database at a constant Prandtl number of $Pr = 0.76$. The flow defined by its friction Reynolds number given as

$$Re_{\tau,\omega} = \frac{U_\tau h}{\nu_\omega}, \tag{1}$$

with h, the channel half-height, ν_ω the wall kinematic viscosity in m^2/s and $U_\tau = \sqrt{\nu_\omega \partial U_x/\partial y}$ the friction velocity in m/s. 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

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

DNS data was produced by Dupuy *et al.* [8] for a channel flow at $Re_\tau = 180$, with two periodic directions (x) and (z) and boundary temperatures fixed at $T_{\text{cold}} = 293\ K$ and $T_{\text{hot}} = 586\ K$; a representation of the geometry of the problem can be found in Figure 1. We filter the data using a top-hat filter. The mesh is regular both in the streamwise (x) and spanwise (z) directions, and grows in size according to a hyperbolic tangent law along the wall-normal direction. We interpolate the data using a second order interpolation scheme, from the DNS mesh to the LES one, so that the mesh size is reduced from $384 \times 384 \times 266$ cells to $48 \times 52 \times 48$ cells. The number of mesh points is reduced to match the finest mesh used for T-LES in Dupuy *et al.* [9]. A top-hat filter spreading over 3 cells in each direction is used. We summarize our data generating algorithm by

$$T^f \xrightarrow{\text{Coarsening}} T^c \xrightarrow{\text{Filtering}} \overline{T}^c. \tag{3}$$

For any quantity ϕ, we define the top-hat filter as

$$\overline{\phi}(x_i, y_k, z_j, t) = \frac{1}{9(y_{k+\frac{3}{2}} - y_{k-\frac{3}{2}})} \sum_{i',j',k'=i-1,j-1,k-1}^{i+1,j+1,k+1} \phi(x_{i'}, y_{k'}, z_{j'}, t)(y_{k'+\frac{1}{2}} - y_{k'-\frac{1}{2}}). \tag{4}$$

with $y_k + \frac{1}{2}$ being the coordinate of the cell face.

3. Reconstruction

To benchmark our algorithm, we use the already existing deconvolution method proposed and developed by Adams *et al.* [7, 10, 11]. This method is based on the Van Cittert deconvolution. This method assumes that for any given LES filter G, if G is invertible, then there exists a

converging Neumann series such that the inverse G^{-1} can be written as

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

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

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

The way of reconstructing using the Van Cittert deconvolution follows

$$T^f \xrightarrow{\text{Coarsening}} T^c \xrightarrow{\text{Filtering}} \overline{T}^c \xrightarrow{\text{Reconstruction method}} T_r^c. \tag{7}$$

4. Deep Learning method

Because it is easier to learn a residu than the reconstruction itself, we propose learning the reconstruction residual $f_{NN}(\cdot, \theta)$ from

$$T_r = \overline{T} + f_{NN}(\overline{T}; \theta), \tag{8}$$

with θ the network weights. The architecture is similar to that of Lapeyre *et al.* [6], with a different number of filters. In the first level of the U-Net architecture, we have 2 filters, the second level has 4 filters, and the 3rd level has 8 filters. This reduces the number of overall parameters to 5737 from an original 1.5 million.

4.1. Data acquisition

As detailed in Section 2, we have a DNS database at our disposal, with approximately 5000 time steps, regularly spaced by $\Delta_t^+ = \Delta_t U_\tau / h = 1.2 \times 10^{-3}$. To minimize the correlation between each time step, we sample our database with $\Delta_t^+ = 7.4 \times 10^{-2}$. We are thus left with 820 time steps, which we split into an 80%, 20% proportion for training and validation. All results shown are for the validation dataset. These time steps are all sampled after full statistical convergence of the flow.

4.2. Learning procedure

The learning procedure is only applied to 16 mesh points closest to the center of the canal to simplify the learning procedure.

On this fixed height, at each step of the mini-batch training, we run an inference on 150 time steps at the same time. This number of time steps was imposed by GPU memory limitations. We then compute the two point correlation at each height in the chosen subdomain. This procedure is done on the coarse mesh to have a similar structure as in process 7. The loss function is defined as

$$\mathcal{L}(T, T_r, \overline{T}) = \frac{MSE(C_2(T^{(1,\cdots,N)}), C_2(T_r^b))}{\|\langle T^2 \rangle_{x,z,t} - \langle T \rangle_{x,z,t}^2\|_2} + C \|\langle f_{NN}(\overline{T}) \rangle_{x,z,t}\|_2,$$

The term $\|\langle T^2 \rangle_{x,z,t} - \langle T \rangle_{x,z,t}^2\|_2$ corresponds to the Mean Squared value (MS), under the Euclidean norm. C_2 corresponds to the two point correlation, which expresses as

$$C_2(T)(X, y, Z) = \langle T(x, y, z, t)T(x + X, y, z + Z, t)\rangle_{x,z,t} - \langle T(x, y, z, t)\rangle^2_{x,z,t}. \quad (9)$$

C_2 is thus a 3D field, function of the shift in x, written X, y, and the shift in z, Z. The operator $\langle\cdot\rangle_{x,z,t}$ corresponds to an averaging over x, z, and t, our periodic directions. The component corresponding to $\|\langle f_{NN}(\overline{T})\rangle_{x,z,t}\|_2$ of the loss function is a known constraint, as when writing a reconstruction of T, we can write $T = \overline{T} + T''$, with T'' corresponding to random fluctuations, with constraint $\langle T''\rangle = 0$, we add a weighting $C = 10^{-4}$ to this constraint, as not weighting it results in neural network outputs that are of worse quality. This constant was picked through trial and error, for 11 values ranging from 10^{-6} to 10^{-1}, with a multiplicative step of $\sqrt{10}$. The superscript $\cdot^{(1,\cdots,N)}$ corresponds to the whole training data set, while the superscript $\cdot^b$ corresponds to the given batch of time steps.

The convolutional neural network learns using a stochastic gradient optimization technique (ADAM, Kingma and Ba [12]) over 150 optimization cycles, called epochs with a learning rate fixed at 10^{-3} at the beginning and decreasing by 2% each epoch. These values are chosen by taking values used in Lapeyre *et al.* [6], and adapting it through trial and error. This training procedure requires a high volume of memory, thus requiring to change the size of the input in the wall normal direction. We thus take no more than 16 mesh points in this direction, leaving us with a $48 \times 48 \times 16$ subdomain to reconstruct. We are not able to accommodate more than 150 time steps per epoch. The training takes approximately 30 minutes on an NVIDIA RTX A6000.

5. Results

In the following section, we will compare temperature profiles, RMS profiles

$$\langle T'^2\rangle = \langle T^2\rangle - \langle T\rangle^2,$$

where $\langle\cdot\rangle$ is the spatial averaging over periodic directions (x) and (z), and time (t), and two point correlations C_2 obtained by running the DNS, LES and reconstruction methods for our validation dataset, giving us a better appreciation of the generalisation capabilities of the network.

5.1. Scale similarity method

We now consider the literature reconstruction method as described in process 7. In Figure 6, we can see the RMS quantities are very close to the DNS ones. Figure 4, the right most column shows the two point correlation C_2 against the DNS and LES. It is clear that the Van Cittert reconstruction method changes and improves upon the two point correlation of the LES. Finally, in Figures 2 and 3, we observe the two point correlation C_2 at a fixed height, corresponding to the center of the canal. We see this method lines up with the DNS very closely for the streamwise, and spanwise directions, with a degradation in the streamwise direction.

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 5. At the end of this learning process, the network produces RMS values close to DNS ones, as can be seen in Figure 6. The network also manages to reconstruct the two point correlation C_2 as shown in Figure 4. Finally,

in Figures 2 and 3, we observe the network staying consistently close to the DNS two point correlation, indicating its capacity to generalize at the task of reconstructing statistics. While the neural network manages to correct the statistics of the reconstructed flow, it is important to note that the network lacks the capacity to spatially reconstruct.

6. Conclusion

The present work demonstrates the ability of a deep learning based reconstruction method to learn to correct the statistics of *a priori* LES. This method was also compared to the well established Van Cittert method. The latter method showed excellent results when it comes to reconstructing the RMS, with results close to the DNS, as well as the reconstruction of the two point correlation, with slight degradations in the streamwise direction. On the other hand, the CNN shows results close to the DNS when it comes to the reconstructing of the RMS, and very close two point correlations, indicating this machine learning procedure is well suited to correct flow statistics. Future work will consist in reconstructing *a posteriori* LES, using the Van Cittert reconstruction as a baseline method.

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.

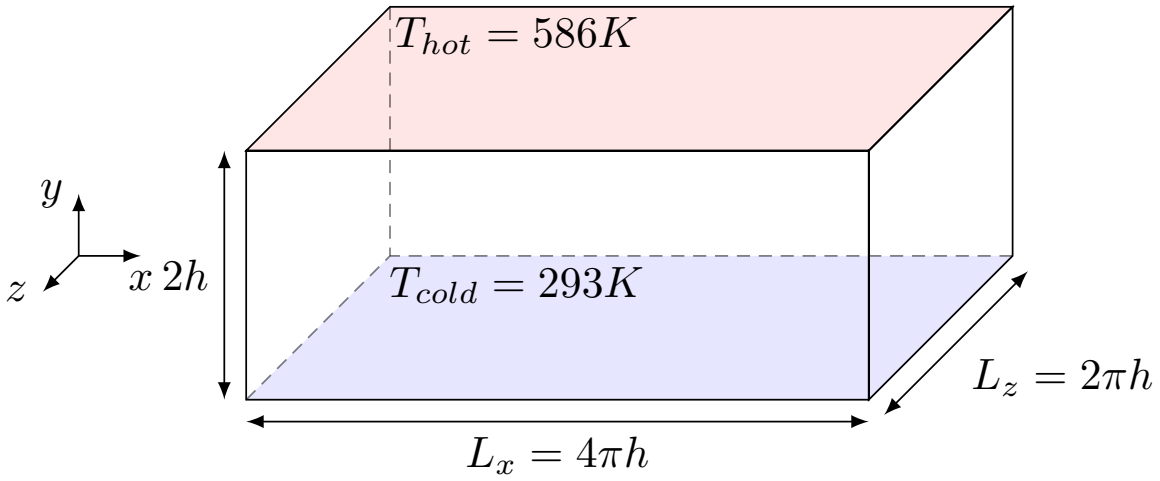

Figure 1: *Configuration geometry*

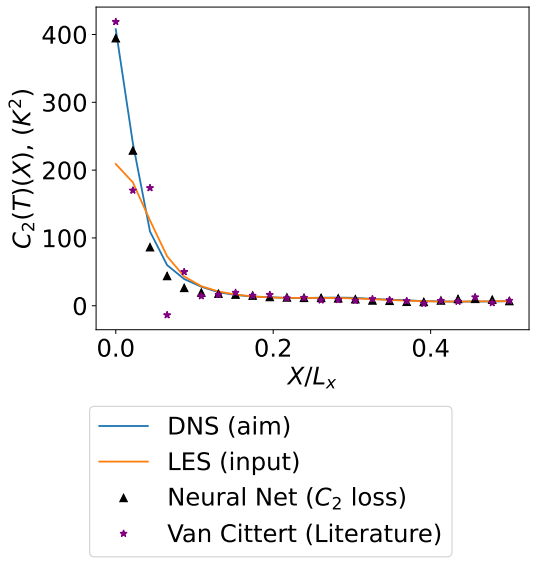

Figure 2: *Two point correlation at channel center $C_2(T)(X, y = h, Z = 0)$ as a function of the streamwise shift for the DNS, LES, the neural network, and the Van Cittert reconstruction method*

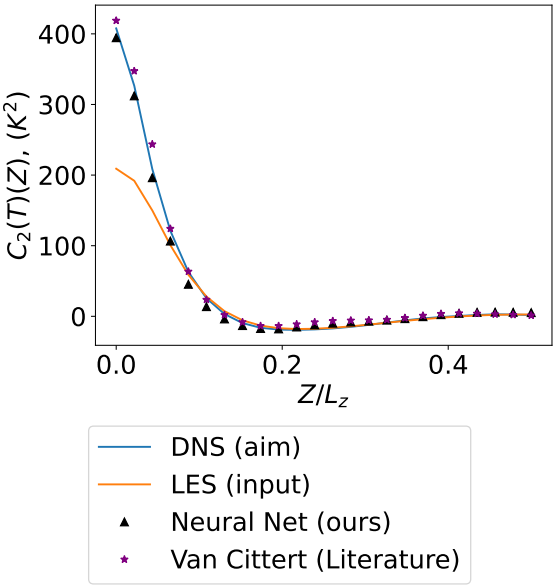

Figure 3: *Two point correlation at channel center $C_2(T)(X = 0, y = h, Z)$ as a function of the spanwise shift for the DNS, LES, the neural network, and the Van Cittert reconstruction method*

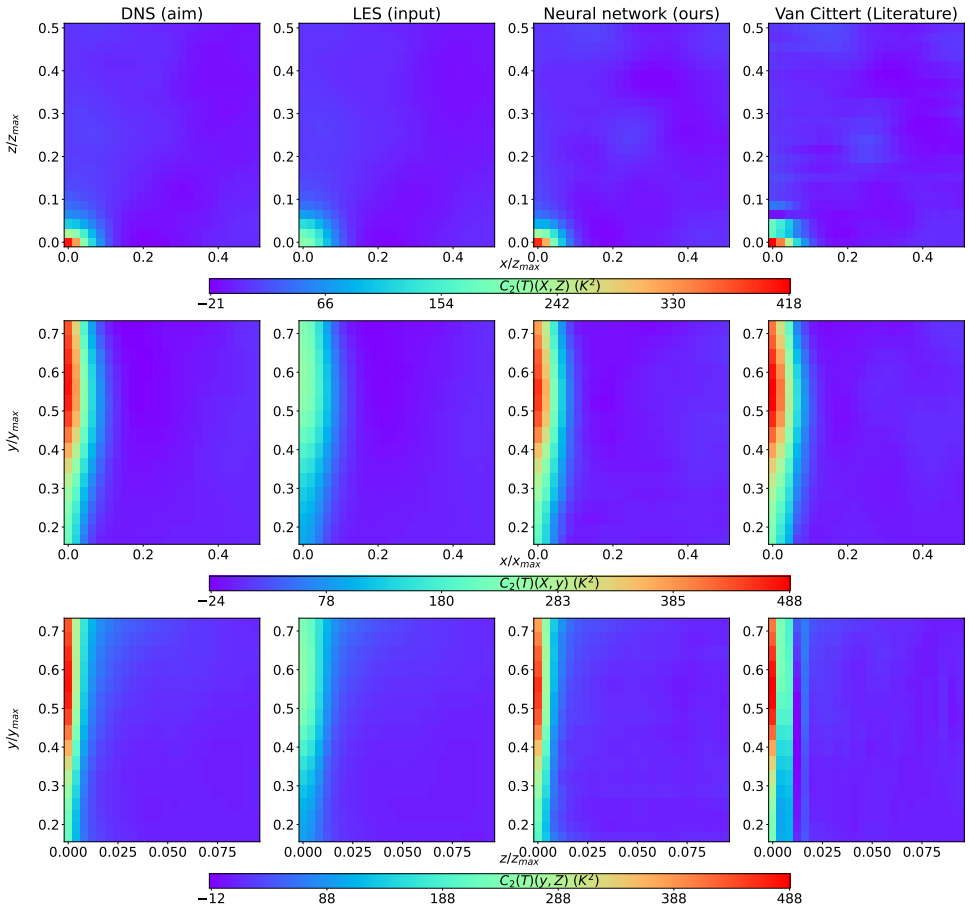

Figure 4: *Qualitative assessment of reconstruction two point correlation for different planes of the close to the channel center. First line corresponds to (X, y=h, Z) plane, second one corresponds to the (X, y, Z=0) plane, and last line is (X=0, y, Z) plane*

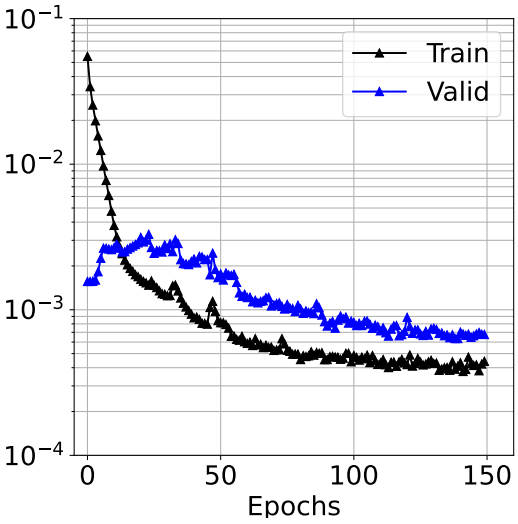

Figure 5: *Training loss*

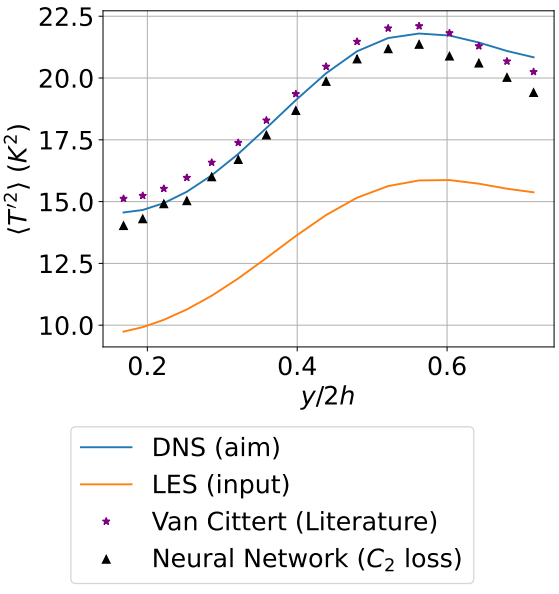

Figure 6: *RMS profile with DNS, LES, Neural Network and Van Cittert method*

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] D. Dupuy, A. Toutant, and F. Bataille. A posteriori tests of subgrid-scale models in strongly anisothermal turbulent flows. *Physics of Fluids*, 31(6):065113, 2019. ISSN 1070-6631.

[10] 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.

[11] 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.

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