

Ultra-scaled deep learning temperature reconstruction in turbulent airflow ventilation

Filippos Sofos (Φίλιππος Σοφός),^{1,2} Dimitris Drikakis (Δημήτρης Δρικόκης),^{1,a)} and Ioannis William Kokkinakis (Ιωάννης Κοκκινάκης)¹

¹*Institute for Advanced Modeling and Simulation, University of Nicosia, Nicosia CY-2417, Cyprus*

²*Condensed Matter Physics Laboratory, Department of Physics, University of Thessaly, Lamia 35100, Greece*

A deep learning super-resolution scheme is proposed to reconstruct a coarse, turbulent temperature field into a detailed, continuous field. The fluid mechanics application here refers to an airflow ventilation process in an indoor setting. Large eddy simulations are performed from a dense simulation grid and provide temperature data in two-dimensional images. The images are fed to a deep learning flow reconstruction model after being scaled down to 100 times. Training and testing are performed on these images, and the model learns to map such highly coarse fields to their high-resolution counterparts. This computational, super-resolution approach mimics the process of employing sparse sensor measurements and trying to upscale to a dense field. Notably, the model achieves high performance when the input images are scaled down by 5 – 20 times their original dimension, acceptable performance when 30, and poor performance at higher scales. The peak signal-to-noise ratio, the structure similarity index, and the relative error between the original and the reconstructed output are given and compared to common image processing techniques, such as linear and bicubic interpolation. The proposed super-resolution pipeline suggests a high-performance platform that calculates spatial temperature values from sparse measurements and can bypass the installation of a wide sensor array, making it a cost-effective solution for relevant applications.

I. INTRODUCTION

The choice of an efficient sensing mechanism is essential for many science and engineering principles, such as heat transfer, fluid flow control, astrophysical data, environmental processes, biological and medical applications^{1–5}, to mention a few. Of equal importance is the post-processing of the discrete measurements and their upscaling to a nearly continuous field, i.e., from a coarse to a fine grid, with the aid of a minimum number of input data.⁶ In this direction, sparse sensor measurement reconstruction has emerged and found fertile ground in fluid mechanics, especially concerning highly turbulent flow fields.

The super-resolution (SR) method has been widely applied in sparse sensor measurement reconstruction. This data-driven approach maps measurements from an array of sensors to a computational grid of discrete values. This applies to 2D or 3D image data processing, where pixels correspond to sensor positions. A sparse array of sensors maps to a low-resolution image. The key point here is to upscale the low-resolution image to a fine-resolution one, providing a continuous field and practically substituting the need to impose a dense sensor array. Peng *et al.*⁷ has managed to obtain full temperature field reconstruction with a (200 × 200) resolution from 16 temperature observations. Carter *et al.*⁸ indicated several unresolved issues related to the reconstruction parameters, the number of detailed fields needed to train the models, and the number of sensors in the investigated region. Furthermore, sensor position optimization is another critical and longstanding open issue requiring further investigation as it could have a significant impact.⁹ Other issues that impose

difficulties include noisy environmental phenomena and processing power demands.¹⁰

In addition to SR, reduced-order models (ROMs) have been well-suited to complex fluid mechanics problems to reduce the huge computational cost imposed.¹¹ Proper orthogonal decomposition (POD) methods¹² have been exploited for dimension reduction, which may be considered as a case of unsupervised machine learning (ML) method in the form of a single-layer neural network (NN).^{8,13,14} Hybrid schemes have also emerged to ensure computational accuracy and efficiency, combining POD with coupling reconstruction methods based on the particle swarm optimization technique.¹⁵ Nevertheless, the most common methods in fluid mechanics are based on shallow ML methods^{16–18} and deep learning (DL) architectures, mainly with convolutional neural networks (CNNs).^{19–22}

Convolutional layers are well-posed to imaging operations due to their ability to capture and reveal features from big data with advanced computational approaches that demand only a fraction of the time and resources needed by standard fully connected neural networks (FCNNs).²³ Along with sub-grid simulation data upscaling²⁴ or data assimilation methods that combine actual measurements with simulations²⁵, the emphasis is on obtaining fine mesh quantities by minimizing the error between the received data and given model predictions. The choice of an appropriate method depends on parameters such as the amount of available data and the desired accuracy. This is driven by cost issues and technical considerations of the sensors. In an SR architecture, CNN layers (i.e., convolutional, deconvolutional, pooling and activation layers) connect appropriately. More accurate architectures are built by further incorporating recurrent neural networks (RNNs)²⁶, generative adversarial neural networks (GANs)^{27,28}, and auto-encoders/decoders.²⁹ After the introduction of physics-informed neural networks (PINNs)³⁰, the concept of physics-driven approaches has also emerged.^{31–33}

^{a)}Electronic mail: Author to whom correspondence should be addressed: drikakis.d@unic.ac.cy

Furthermore, accurate results in heat and flow field reconstruction from sparse observations were achieved by embedding Fourier neural operators in the neural network.³⁴

This paper invokes high-resolution temperature fields from implicit Large Eddy Simulations (ILES) and scales down to ultra-low resolution fields, down to 100 times compared to the original dimensions. It is a fact that in computer vision applications, to our knowledge, standard coarsening scales for training an SR framework range from 2 to 4 times.^{35–37} The aim is to theoretically map an ultra-low resolution image to an array of sensors that measures temperature values inside a closed rectangle region. The Deep Learning Flow Image (DELFI) architecture, which has been previously applied for reconstructing the velocity field in an incompressible sudden expansion application and the pressure field in a shock-wave turbulent boundary layer interaction^{38,39}, is also employed here. DELFI has been compared to popular architectures used in fluid mechanics SR applications and found to perform equally well. However, being lightweight and fast (the reader can refer to the comparison made in Sofos, Drikakis, and Kokkinakis³⁹).

The temperature field is considered for investigating the model's performance, as it is an essential parameter for indoor ventilation and highly important for thermal and energy management.^{40–43} Therefore, temperature maps (2D images) are taken from high-resolution computational fluid dynamics (CFD) simulations of typical room air ventilation. The high-resolution fields are used to compare and train DELFI, along with ultra-low-resolution counterparts. As mentioned before, these counterparts can be seen as sparse sensor values, and the inherent difficulty lies in reconstructing these discrete points into a meaningful, continuous field. We show that the proposed SR method is superior to typical image processing techniques like linear or bicubic interpolation and offers high performance in ultra-scaled scenarios. After being appropriately trained, it can be fast, compared to other SR models,³⁸ and easy to run for training or validation on standard hardware, adding to its appeal and potential in the field.

The paper is organized as follows. Section II presents the simulation details and data acquisition and describes the implied SR architecture. The proposed model reconstruction ability is compared to common image processing methods and presented in terms of qualitative and quantitative results in Section III. Finally, Section IV summarizes the main conclusions and provides suggestions for future work.

II. MODELS AND METHODS

A. Equations

The three-dimensional Navier-Stokes (NS) equations for a Newtonian viscous fluid are considered. For a finite control volume, Ω , and ignoring body forces, the NS can be written in conservative variables and Cartesian co-ordinates form as follows:

$$\frac{\partial}{\partial t} \int_V \mathbf{U} dV + \oint_S (\mathbf{F}_a + \mathbf{F}_s + \mathbf{F}_q) dS = 0, \quad (1)$$

where $\mathbf{U}$ represents the vector of the conservative variables, $\mathbf{F}_a$ is the vector of the advective fluxes, $\mathbf{F}_s$ is the vector of the total (pressure and viscous) stress fluxes, and $\mathbf{F}_q$ is the vector of the heat conduction fluxes:

$$\begin{aligned} \mathbf{U} &= [\rho, \rho u, \rho v, \rho w, e_t]^T \\ \mathbf{F}_a &= [\rho, \rho u, \rho v, \rho w, e_t]^T \mathbf{u}_\perp \\ \mathbf{F}_s &= [0, S_1, S_2, S_3, \mathbf{S} \cdot \mathbf{u}]^T \\ \mathbf{F}_q &= [0, 0, 0, 0, \mathbf{q} \cdot \hat{n}]^T \end{aligned} \quad (2)$$

where $\mathbf{S} = [S_1, S_2, S_3]^T = \hat{n} \cdot (\rho \mathbf{I} - \mathbf{T})$ represents the pressure and viscous (shear and normal) stresses in each corresponding Cartesian direction, $e_t = e_i + \rho(\mathbf{u} \cdot \mathbf{u})/2$ is the total energy density, ρ is the density; $\mathbf{u}$ is the velocity vector; p is the static pressure; $\mathbf{u}_\perp = (\mathbf{u} \cdot \hat{n})$ is the velocity magnitude normal to the surface element dS ; $\hat{n}$ is the outward pointing unit normal of a surface element dS belonging to the closed boundary $\partial\Omega$ of the control volume Ω ; e_i is the internal energy density, which for a calorically perfect gas is given by:

$$e_i = \rho c_v T = \frac{p}{\gamma - 1} \quad (3)$$

where T is the temperature, c_v is the specific heat capacity at constant volume, and γ is the heat capacity ratio (or adiabatic index) defined as $\gamma = c_p/c_v$, where c_p is the specific heat capacity at constant pressure and $R_s = c_p - c_v = c_v(\gamma - 1)$ is the specific gas constant.

The viscous stress tensor is given by:

$$\mathbf{T} = \lambda_b (\nabla \cdot \mathbf{u}) \mathbf{I} + \mu [\nabla \otimes \mathbf{u} + (\nabla \otimes \mathbf{u})^T] \quad (4)$$

where $\mathbf{I}$ is the identity tensor. The bulk viscosity is given by $\lambda_b = -4\mu/3$ according to Stokes' hypothesis, and μ is the dynamic viscosity obtained according to Sutherland's law:

$$\mu(T) = \mu_{\text{ref}} \left(\frac{T}{T_{\text{ref}}} \right)^{1/2} \frac{T_{\text{ref}} + T_s}{T + T_s} \quad (5)$$

where the free-stream values are used as the reference and the Sutherland temperature is $T_s = 110.4$ K.

The heat flux is calculated according to Fourier's Law of heat conduction:

$$\mathbf{q} = -\kappa \nabla T \quad (6)$$

where κ is the heat conductivity given by:

$$\kappa(T) = \frac{c_p}{\text{Pr}} \mu(T) \quad (7)$$

and $\text{Pr} = 0.72$ is the Prandtl number.

B. Numerical solver

The simulations were performed using the block-structured grid code CNS3D, which solves the above Navier-Stokes

equations using the finite-volume method (FVM).^{44,45} CNS3D can be used for implicit Large Eddy Simulations (ILES) and Direct Numerical Simulations (DNS). CNS3D has been extensively validated for low- and high-speed flows, including turbulent mixing.^{44–48} The advective terms are solved using the Godunov-type (upwind) method. The Riemann solver is solved at the cell interfaces. The primitive variables are reconstructed at the cell interfaces and used to calculate the numerical fluxes. For the spatial reconstruction, a one-dimensional swept unidirectional stencil is used. The numerical simulations herein were obtained using an augmented⁴⁴ 11th-order weighted essentially non-oscillatory (WENO) scheme^{49–51} for interpolation in conjunction with the “Harten, Lax, van Leer, and (the missing) Contact” (HLLC) approximate Riemann solver by Toro, Spruce, and Speares⁵². Further details of the implemented WENO scheme and HLLC Riemann solver are provided in Appendix A and B, respectively. The solution is advanced in time using an explicit five-stage (fourth-order accurate) optimal strong-stability-preserving Runge-Kutta method of Spiteri and Ruuth.⁵³

C. Flow setup

The room is modelled as rectangular with dimensions of $6.0 \times 2.4 \times 2.8 \text{ m}^3$ (length $\times$ height $\times$ width). A standard air conditioning (AC) unit is placed at the cabin's center with a square outlet (48 cm^2) and four rectangular inlets ($48 \times 4 \text{ cm}^2$ each) that expel air at an angle of 45° degrees, as per the schematic in Fig. 1.

The initial room temperature is 25°C with a 60% relative humidity. The simulation is carried out using two components (multi-component flow), where the first component is considered to be dry-air (0% moisture) with an adiabatic index of $\gamma_{da} = 7/5$ and molar mass of $\mathcal{M}_{da} = 28.964 \text{ kg/kmol}$, and the second component is water vapor with an adiabatic index of $\gamma_{wv} = 4/3$ and molar mass of $\mathcal{M}_{wv} = 18.015 \text{ kg/kmol}$. The velocity is initially assumed to be equal to zero everywhere in the room, and the total energy varies in the normal (y) direction due to gravity, i.e.,

$$e_t = e_i + \rho_m g_y y, \quad (8)$$

where the Earth's gravitational acceleration is taken as constant $g_y = -9.81 \text{ m/s}^2$. The internal energy, e_i is calculated according to:

$$e_i = \frac{p_{da} + p_{wv}}{\gamma - 1} \quad (9)$$

where p_{da} is the room pressure of the dry air component and p_{wv} is the water vapor pressure. The former is obtained according to $p_{da} = p_0 - p_{wv}$, where p_0 is taken here as the stagnation pressure at ground/sea level, i.e., $p_0 = 101,325 \text{ Pa}$. The water vapor pressure is calculated according to $p_{wv} = x_{wv} p_0$, where the molar-fraction is $x_{wv} = w_{wv} (\mathcal{M} / \mathcal{M}_{wv})$, and the total molar mass of a miscible mixture is obtained according to:

$$\mathcal{M} = \frac{1}{\sum_i (w_i / \mathcal{M}_i)} \quad (10)$$

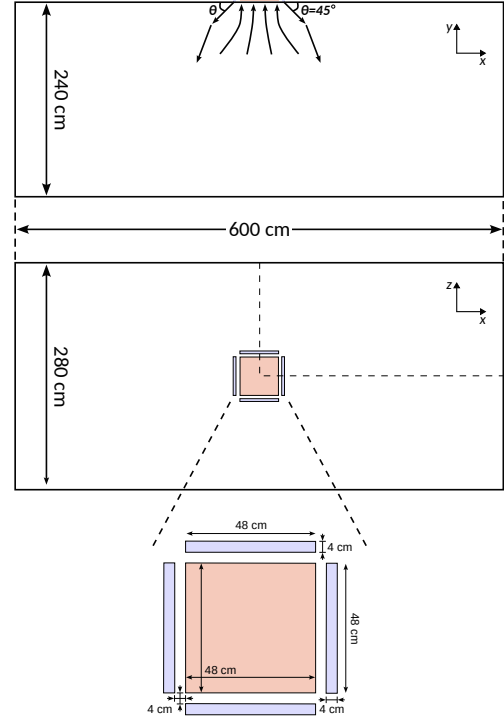


FIG. 1. Shape and size of the room and AC unit considered.

where w_i is the mass-fraction of the i -th component. The initial air mixture density is taken to be constant and is calculated based on the prescribed relative humidity, i.e., the fraction of water vapour and dry air:

$$\rho_m = \rho_{da} + \rho_{wv} \quad (11)$$

where the individual component density is calculated from $\rho_i = p_i / (R_s^i T_0)$. Here, R_s^i is the specific gas constant of the i -th component, while T_0 is the initial (stagnation) room temperature.

The light-blue shaded regions in Fig. 1 are the AC unit inlets, pushing cold air ($\sim 18^\circ \text{C}$) into the room at 4 m/s with relative humidity (RH) of 40% and at an angle of $\theta = 45^\circ$ degrees outwards. Finally, it is mentioned that ventilation rates in a room may vary depending on the application.^{54–56}

The total mass-flux of the AC inlets is (counter-)balanced by the light-red color-shaded square AC outlet located precisely in the middle of the room ceiling, surrounded by the AC inlets. A stagnation pressure outflow condition is used of $\simeq 101,324 \text{ Pa}$. As mentioned, the near-wall boundary layer is not resolved in the present simulations to help reduce the overall computational expense. Therefore, the room walls are modelled as slip instead of no-slip; these include all remaining

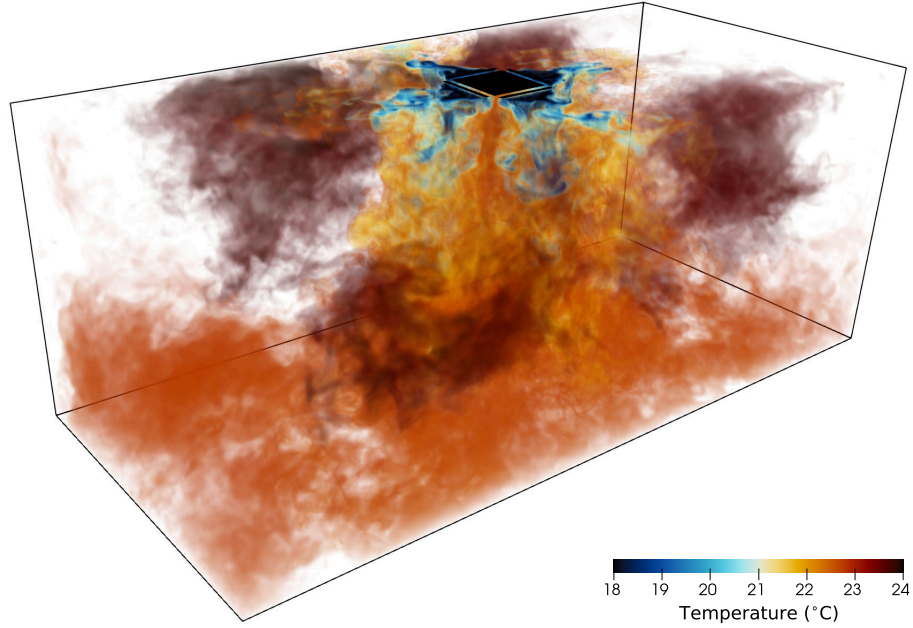


FIG. 2. Three-dimensional contours of the flow temperature field in a rectangular room approximately ~ 72 seconds after the air-conditioning (AC) unit is turned on. Initial room temperature at 25°C , AC inflow temperature at 18°C . Pockets of warm air can still be seen as illustrated by the dark-red colored regions (color bar max temperature value chosen to better visualize the hot air regions $\in [24, 25]^\circ\text{C}$).

TABLE I. Mesh convergence of the mean room temperature at $t = 90$ seconds.

	Mesh 1	Mesh 2	Mesh 3
$\bar{T}_{\text{mean}} (^\circ\text{C})$	22.6	22.9	23.7
Difference (%)	0	1.3	5

boundary surfaces, i.e., the room floor, side walls, and ceiling.

The computational results are obtained on a block-structured hexahedral orthogonal Cartesian mesh comprising of cells with an edge length of 2 cm ($\Delta\mathbf{x}^+ \simeq 700$), resulting in a mesh resolution of $300 \times 120 \times 140$, giving a total of 5,040,000 cells. In essence, the mesh comprises cells of the same size in all directions, i.e., $\Delta x = \Delta y = \Delta z = 2$ cm.

Figure 2 illustrates the temperature iso-surfaces of resolved three-dimensional turbulent flowfield approximately ~ 72 seconds after the AC unit began cooling the room. The geometry of the AC units and inlet conditions is such that the resulting room flow circulation forms four hot-spot regions symmetric about the room's floor plan (xz -plane) midsections. Regarding height (y -axis), these hot spots are located above 1.2 meters, mainly around the room's ceiling, and indicate the regions where the AC units affect the least.

Table I compares the mean room temperature between the

high-resolution mesh (mesh 1) and two coarser meshes (mesh 2 and 3). The coarser meshes comprise hexahedral cells with double the edge length (i.e., 4 and 8 cm) compared to mesh 1. After 90 seconds, the results exhibit less than $< 1.3\%$ differences. The smaller turbulent scales resolved on finer mesh resulted in a slightly greater mixing between the dry, cool air-conditioning air and the warm, humid, still air in the room. Consequently, the obtained mean temperature and relative humidity are slightly lower on the finer mesh. Though an even finer mesh would resolve smaller turbulence flow scales, it would require a significantly greater computational cost while providing diminishing returns for this study. For example, discretizing the present room size using hexahedral cells with an edge length of 1 cm would increase the computational cost by $\times 16$ while providing an even smaller incremental difference, $< 1.3\%$, to the mean room temperature obtained.

Based on the AC inlet height (4 cm) and inlet velocity (4 m/s), an inlet Reynolds number of $Re_j \simeq 11,000$ is obtained. Compared to wall-bounded turbulent flows, free-shear turbulent flows are typically much easier to resolve since the largest turbulent scales are of the order of the thickness of the shear layer. Nonetheless, for turbulent jet flows, the axial mesh resolution near the inlet is of the order of $\mathcal{O}(-3)$ of the inlet's diameter (round jet) or height (planar jet). However, applying such a fine resolution would make the computational

requirements prohibitively large, even if restricted to the flow regions surrounding the AC inlets. However, the air velocity varies at around 0.1 – 0.4 m/s along the walls in the rest of the domain. The purpose of the present study is not to elucidate the turbulence properties of the suddenly expanded turbulent flow emitted from the AC jet inlets. The mesh convergence study demonstrates that the largest and most energetic turbulent scales driving most of the mixing between the initially hot air in the room and the cold air injected by the AC inlets have been reasonably resolved.

D. Data acquisition

From a technical point of view, the model processes a dataset that contains a time sequence of $\sim 10,000$ images (precisely 9,986 time instants) in the xz -plane and another pool of 10,000 images in the yz -plane (see Fig. 3). The data is gathered over 90 sec (one and a half minutes), resulting in a sampling frequency of ~ 0.009 Hz. Images correspond to temperature contours ranging between 18 to 25 °C. The images are in 3-channel mode (red - R, green - G, and blue - B). Colored images can generally represent the range of intensities better than grayscale images for the human eye.⁵⁷ Actual temperature values can be taken after the 3 channel pixel values are averaged in one layer. The extracted images are randomly divided into training (80%) and validation (20%) datasets.

E. Super-resolution architecture

The employed super-resolution architecture is shown in Fig. 3. It is based on the DELFI model, which has been extensively tested on shock-boundary layer interaction (pressure fields)³⁸ and suddenly-expanded flows (velocity fields)³⁹, and found to achieve excellent performance, with minimum resource and computational time. Similar architectures based on the prototype U-Net⁵⁸ architecture have also been incorporated to reconstruct temperature fields.⁵⁹

In this study, DELFI reaches its limit of reconstruction of ultra-low scale fields, up to 100 times lower than the input images. The cases tested are presented in Table II. It is pointed out that the specific application is not directed at obtaining a highly accurate temperature field. The focus is on assigning image pixel values to construct a computational framework resembling a sparse sensor network. In such a way, the computational platform can draw a finer temperature field from sparse measurements.

Training data enters the scale-down stage (Fig. 3) to resemble a sparse sensor network. As the amount of input data is significant and memory demands are increased to process all images simultaneously, the images are fed sequentially into the pipeline in batches of 1000 images per training instance. This process allows to train the model faster, without increased memory demands⁶⁰ so that it can run even in standard computer hardware. An alternative way would be to employ transfer learning techniques after pre-training the model

with a similar image set and fine-tuning the model weights to new image data.⁶¹

For every batch of 1000 image samples, we use 800 for training and 200 for validation, continuing this process until the entire dataset is utilized. This allows us to monitor the model's performance during training. The current approach does not include a separate test dataset for final evaluation; the performance is monitored using validation data throughout the training process. We have trained the model in three computer architectures to assess its performance. Table III shows the time needed to train DELFI on this dataset for every 1000 image samples. It takes about 29hrs to run the training procedure on 8 cores of an AMD CPU, about 10hrs on 40 cores of an Intel Xeon architecture, while, for the same task, 1hr only is needed on a computer with 2 NVIDIA Graphical Processing Units (GPUs).

After the scaling-down stage, low-resolution images (according to the scale factor, s , shown in Table II) enter the DELFI framework and are paired to their satisfactory resolution, i.e., the ground truth counterpart. The reconstructed image is given at the final stage of the process.

III. RESULTS

To qualitatively assess the proposed method, the investigation starts from an original high-resolution temperature field (i.e., the ground truth image), and the reconstruction ability of the proposed architecture is assessed over six different input resolutions. Furthermore, the DL reconstruction performed here is compared with the result using linear interpolation (LI) and bicubic interpolation (BI), standard methods used in image processing tasks. Both these methods interpolate the values of pixels of a lower-resolution image to create a higher-resolution image based on the values of neighboring pixels.

To reach an ultra-scaled field resembling an array of sensors, the resolution of the image used as input to the DELFI pipeline gradually decreases. In applications where the objective is to upscale a sparse spatial field - composed of a variety of discrete measurements coming from a sparse sensor array - to a fine field, an alternative method that reaches satisfactory accuracy is based on first dividing the field into sub-regions, usually with Voronoi tessellation, and applying reconstruction techniques on each sub-region.⁶²

The present study expands the applicability of DELFI in ultra-low scaling conditions, as seen from the scale factors and the respective dimensions in Table II. Figure 4a presents the ground truth image from a time instance near the beginning of the simulation, as the cold air flows out of the AC inlets at the ceiling and gradually fills and cools the room. The white rectangle area, chosen for its highly-unstable behavior, is magnified for comparison with the following reconstructed fields. In Fig. 4b, the scaled-down image ($s = 5$) is shown in the leftmost column, followed by the LI output in the second column, the BI output in the third column, and the SR result derived from DELFI in the rightmost column. All three presented reconstruction methods, i.e., the LI, the BI, and the SR, have performed well in reconstructing the scaled

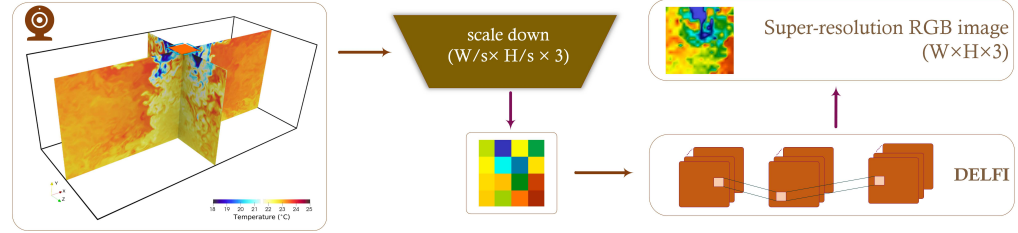


FIG. 3. The implied super-resolution architecture, based on DELFI³⁹, with s the scale factor that leads to a coarser resolution of the input, high-resolution field. The scaled image is loaded to DELFI and paired with the ground truth image. The training process involves learning the model how to map from a coarse into a finer, reconstructed field.

TABLE II. Image dimensions for every scale investigated. The value $s = 1$ refers to the original image dimension.

s	1	5	10	20	30	50	100
$W \times H$	2240 × 896	448 × 179	224 × 89	112 × 44	74 × 29	44 × 17	22 × 8

TABLE III. Comparison of time required to train the DELFI model on various computer architectures.

Hardware	Cores	Images	epochs	time (mins/hrs)
AMD Ryzen (3970X)	8	1000	30	1750/29.17
Intel Xeon (E5-4650v2)	40	1000	30	615/10.25
NVIDIA GPU (Tesla K40 × 2)	5760	1000	30	62.5/1.04

input to a close-to-the-ground truth output. Among them, the SR-reconstructed output seems the closest to the ground truth, with clear edges and no blurring. The BI image is also close, while the LI has improved the pixelated field without achieving a clear output.

The SR-reconstruction results for $s = 10$ (in the rightmost column of Fig. 4c) are also very close to the original region. At the same time, satisfactory reconstruction has also been obtained for SR for $s = 20$, in the rightmost column of Fig. 4d. On the other hand, this is not the case for the LI and the BI reconstruction methods. Highly pixelated outputs for $s = 10$ and $s = 20$ are given (in Figs. 4c-d, respectively), denoting that these standard computer vision techniques have reached their upper performance limit and can no longer perform well at such highly scaled resolutions. Moreover, their reconstruction ability is further decreased for $s=30, 50$, and 100 , as shown in Figs. 4e, f, and g, respectively.

For higher scale factors, $s = 30$ in Fig. 4e, DELFI has performed marginally well, with blurring appearing in the result. When the scaled-down factor increases to $s = 50$ in Fig. 4f, the model reconstructs the blurred input to a finer SR field, which captures only the primary (large-scale) details of the turbulent temperature field. Practically, the input field here is non-continuous, and it would be difficult for an SR method to

upscale it to a meaningful, continuous field that approaches the actual temperature values. Finally, the ultra-low scale input of dimensions (22×8) shown in the leftmost column of Fig. 4g enters the DELFI pipeline and upscales to a finer grid that captures, to some degree, the temperature field behavior. Although the SR result is far from the ground truth, it can still transform the general features hidden in the "squared"-pixel values into a continuous field.

After the cold air has been circulated inside most of the room space, near the end of the simulation ($t = 90$ sec), an additional time instance has been taken for investigation in Fig. 5. The black rectangle in Fig. 5a is retained for comparison with the reconstructed parts. The same qualitative performance is obtained as in Fig. 4 for all scale factors shown. Reconstruction is acceptable for the SR model, especially for $s \leq 30$ (see the rightmost columns of Figs. 5b-d). At the same time, the LI reconstruction images remain acceptable only for $s = 5$ (second column in Fig. 5b), failing to reproduce the temperature field at the higher scales examined. The BI method performs slightly better than LI since its reconstruction performance remains acceptable up to $s = 10$. However, after that ($s > 10$), its performance quickly deteriorates and becomes poor.

Three widely used quantitative metrics that can give a clear view of the reconstruction performance in SR tasks are the PSNR, the SSIM, and the relative error (RE). The PSNR is a widely used metric in image processing to quantify the quality of a reconstructed image compared to its original version. The RE covers common error analysis, and it is based on pixel differences, which are independent of the underlying structures. However, in some cases, images of physical properties might present highly structured behavior, i.e., strong dependencies between the pixels which carry information about the structured regions, such as eddies in turbulent flows. The SSIM measure can capture these variations.⁶³

The PSNR is calculated from the pixel differences between

This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1063/5.0213085

Preprint

7

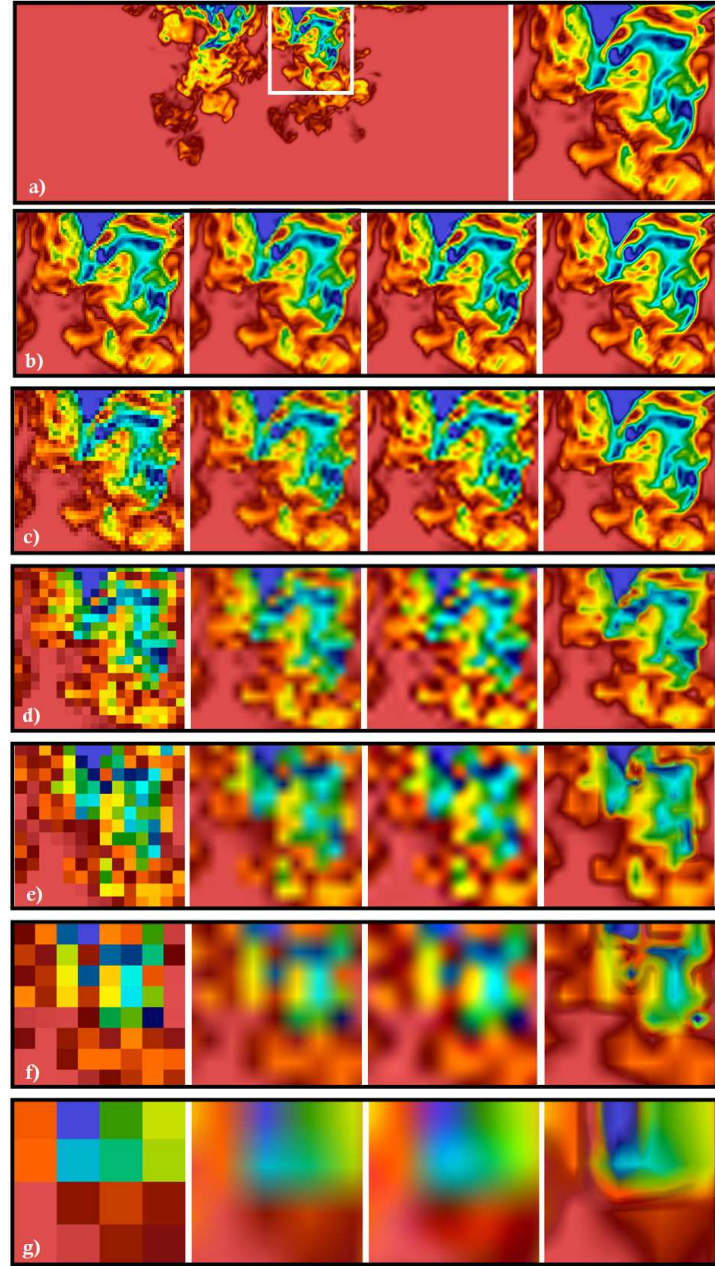


FIG. 4. Airflow temperature field reconstruction, from a time instance near the beginning of the simulation. (a) The ground truth image, with a zoomed region (the white rectangle), shows the field's details. Next, the scaled-down images are shown on the left column, along with their reconstructed counterparts, i.e., the linear interpolated, the bicubic interpolated, and the DELFI output, respectively, for (b) $s = 5$, (c) $s = 10$, (d) $s = 20$, (e) $s = 30$, (f) $s = 50$, and (g) $s = 100$.

This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1063/5.0213085

Preprint

8

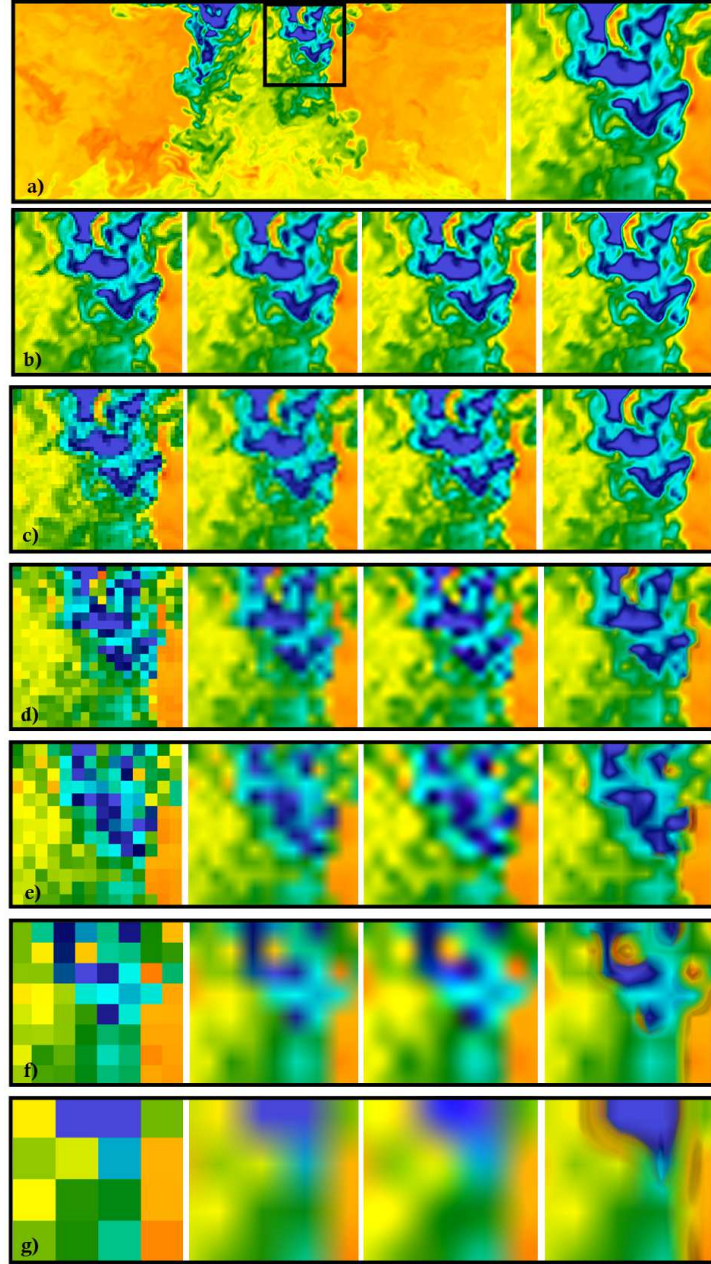


FIG. 5. Airflow temperature field reconstruction, from a time instance near the end of the simulation. (a) The ground truth image, with a zoomed region (the black rectangle), shows the field's details. Next, the scaled-down images are shown on the left column, along with their reconstructed counterparts, i.e., the linear interpolated, the bicubic interpolated, and the DELFI output, respectively, for (b) $s = 5$, (c) $s = 10$, (d) $s = 20$, (e) $s = 30$, (f) $s = 50$, and (g) $s = 100$.

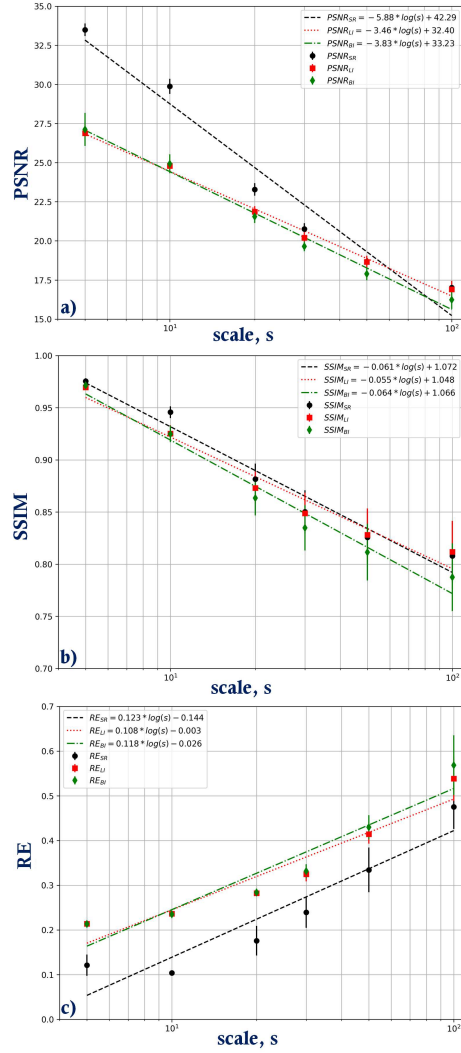


FIG. 6. Reconstruction metrics vs. the scale-down factor, s , (a) PSNR, (b) SSIM, and (c) RE. The error bars denote the standard deviation for each s . Lines are semilog linear fits.

the ground truth and the SR-derived images using^{19,64}:

$$PSNR(\mathbf{X}_{SR}, \mathbf{X}_{GT}) = 20 \log \left(\frac{1}{\sqrt{MSE(\mathbf{X}_{GT}, \mathbf{X}_{SR})}} \right), \quad (12)$$

where $\mathbf{X}_{SR}$ is the SR output, $\mathbf{X}_{GT}$ the ground truth image, and MSE the mean squared error. The SSIM refers to the struc-

tural similarity measures, such as luminance, contrast, and structures. The comparisons, C for each feature, are combined into the equation⁶³:

$$SSIM(\mathbf{X}_{SR}, \mathbf{X}_{GT}) = \frac{[C_l(\mathbf{X}_{GT}, \mathbf{X}_s)]^\alpha [C_c(\mathbf{X}_{GT}, \mathbf{X}_s)]^\beta}{[C_s(\mathbf{X}_{GT}, \mathbf{X}_s)]^\gamma}, \quad (13)$$

where $\mathbf{X}_s$ is the scaled image.

The RE is a measure of the average magnitude of the reconstruction error relative to the ground truth, and, as a percentage, it can be applied to data where large variations exist.⁶⁵ It is given by:

$$RE(\mathbf{X}_{SR}, \mathbf{X}_{GT}) = \left| \frac{\mathbf{X}_{GT} - \mathbf{X}_s}{\mathbf{X}_{GT}} \right| \times 100\%. \quad (14)$$

Reconstruction performance is calculated for each scale factor, s , based on the three metrics (PSNR, SSIM, RE) and compared to LI and BI reconstruction results. At first, as shown in Fig. 6a, the PSNR values for DELFI are consistently higher than the respective LI and BI values, especially for $s < 30$. For $s \leq 10$, $PSNR > 30.0$. The obtained PSNR values are close to the range of reported values for other architectures found in SR for fluid mechanics applications. To mention a few, the downsampled skip-connection/multi-scale (DSC/MS)⁶⁶, the autoencoder convolutional neural network (AE-CNN)⁶⁷ and the multi-scale temporal path U-Net (MST-UNET)⁶⁸ have reported maximum PSNR values in the range 30.17 – 31.18. Regarding generative adversarial networks, which employ a more complex architecture compared to U-Net based ones, the super-resolution GAN (SRGAN)⁶⁹ has given $PSNR=36.31$, while the enhanced super resolution GAN (ESRGAN) has reached $PSNR=37.05$.²⁷ A detailed comparison between various SR models can be found in Sofos, Drikakis, and Kokkinakis³⁹.

Nevertheless, we must remember that in this paper, DELFI is incorporated for ultra-scaled reconstruction tasks, starting from $s = 5$ and reaching $s = 100$. In this direction, an acceptable PSNR value is obtained for $s = 20$, while the performance is lower, and a poorer reconstruction result is obtained for $s = 30, 50, 100$ (Fig. 6a).

Of importance is the fact that the PSNR for our SR model, along with the respective LI and BI values, has shown a decreasing logarithmic dependence on the scale factor values. At the limit of $s \rightarrow 1$ (i.e., the original resolution, without scaling), DELFI achieves $PSNR \approx 42.0$, LI reaches $PSNR \approx 32.0$ and BI has $PSNR \approx 33.0$. Another point is the calculated PSNRs for LI and BI approach DELFI-derived values for $s > 30$, where high deviations from the ground truth image exist.

On the other hand, the SSIM for DELFI presents similar logarithmic behavior (Fig. 6b), which converges to 1 as $s \rightarrow 1$. For all scales investigated here, $SSIM > 0.80$. There are no significant differences in statistical accuracy between the SSIM calculated for DELFI and LI, BI. On the other hand, the relative errors are significantly lower for DELFI compared to LI and BI. In Fig. 6c, it is observed that $RE_{SR} \approx 10\%$ for $s = 5$ and $s = 10$, $RE_{SR} < 20\%$ for $s = 20$, while, a maximum of $RE_{SR} \approx 50\%$ is taken for $s = 100$. In the range investigated

here, the LI and BI start from $RE > 20\%$ at small scales and reach $RE_{SR} \approx 50 - 60\%$ for $s = 100$.

It is obvious that typical image processing techniques, such as LI and BI, are limited in their applicability to highly complex problems and large values of s . The need to introduce DL methods emerges, and the proposed DELFI model can be a reliable choice while being faster than other SR architectures.

IV. CONCLUSIONS

A super-resolution method is applied in this paper to reconstruct an ultra-scaled temperature field into continuous to reach a dense temperature map as close to the ground truth as possible. Discrete temperature values from such highly pixelated images can be seen as sparse sensor values, and their reconstruction to a fully detailed temperature map is the main focus. The application concerns a CFD-simulated air ventilation case with an A/C unit on the ceiling of a prototype closed rectangle room.

Similar super-resolution methods have been widely incorporated in fluid mechanics applications, where 2D or 3D images being processed correspond to property fields of velocity, pressure, vorticity, and temperature, most of the time. Low-resolution variants of the original fields have been used to train the deep learning networks. However, we incorporate ultra-scaled-down fields to train and validate the implied model, DELFI. The coarse fields have been scaled 100 times from the original dimensions. Nonetheless, these ultra-coarse fields have produced performance metrics in the same order as in other studies in the literature where the scaling factor ranges only between 2 and 5. Here, we present increased reconstruction performance with a scaling factor of 20, while acceptable performance is obtained for 30-time scaled-down images.

The performance measures shown include the peak signal-to-noise ratio, the structural similarity index, and the relative error. Of interest is that these measures follow a semi-logarithmic behavior vs the scale-down factor. Results have shown that DELFI is superior to standard computer vision techniques, such as linear and bicubic interpolation, in all cases investigated. Considering that DELFI has been successfully employed in reconstructing velocity and pressure fields in other fluid mechanics applications, we believe it could pose a fast and reliable scheme to tackle complex, ill-posed problems like turbulent flow field estimation.

Future research should consider combining the model with actual data obtained from sensors. As a step further, experimental validation with images from a thermal camera would enhance the present research. The final aim is to exploit a low-resolution camera in place of an array of sensors to accurately provide a detailed temperature map in buildings or even exploit satellite images to monitor a property of interest on Earth's surface.

ACKNOWLEDGMENTS

This paper is supported by the European Union's Horizon Europe Research and Innovation Actions programme under grant agreement No 101069937, project name HS4U (HEALTHY SHIP 4U). The views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Climate, Infrastructure, and Environment Executive Agency. Neither the European Union nor the granting authority can be held responsible. This work was supported by computational time granted by the National Infrastructures for Research and Technology SA (GRNET SA) in the National HPC facility - ARIS under Project No. pr013008-AL_PHYSICS.

DATA AVAILABILITY

The data supporting this study's findings are available on request.

REFERENCES

- ¹H. Wu, H. Zhang, G. Hu, and R. Qiao, "Deep learning-based reconstruction of the structure of heterogeneous composites from their temperature fields," *AIP Advances* **10**, 045037 (2020).
- ²K. Fukami, K. Fukagata, and K. Taira, "Assessment of supervised machine learning methods for fluid flows," *Theoretical and Computational Fluid Dynamics* **34**, 497–519 (2020).
- ³F. J. Yu, J. Lazar, and C. A. Argüelles, "Trigger-level event reconstruction for neutrino telescopes using sparse submanifold convolutional neural networks," *Phys. Rev. D* **108**, 063017 (2023).
- ⁴J. Ray, J. Lee, V. Yadav, S. Lefantzi, A. M. Michalak, and B. van Bloemen Waanders, "A sparse reconstruction method for the estimation of multi-resolution emission fields via atmospheric inversion," *Geoscientific Model Development* **8**, 1259–1273 (2015).
- ⁵S. Zheng and Y. Xiangyang, "Image reconstruction based on compressed sensing for sparse-data endoscopic photoacoustic tomography," *Computers in Biology and Medicine* **116**, 103587 (2020).
- ⁶E. Franz, B. Solenthaler, and N. Thuerey, "Global transport for fluid reconstruction with learned self-supervision," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021) pp. 1632–1642.
- ⁷X. Peng, X. Li, Z. Gong, X. Zhao, and W. Yao, "A deep learning method based on partition modeling for reconstructing temperature field," *International Journal of Thermal Sciences* **182**, 107802 (2022).
- ⁸D. W. Carter, F. De Voogt, R. Soares, and B. Ganapathisubramani, "Data-driven sparse reconstruction of flow over a stalled aerofoil using experimental data," *Data-Centric Engineering* **2**, e5 (2021).
- ⁹K. Manohar, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Data-driven sparse sensor placement for reconstruction: Demonstrating the benefits of exploiting known patterns," *IEEE Control Systems Magazine* **38**, 63–86 (2018).
- ¹⁰J. E. Santos, Z. R. Fox, A. Mohan, D. O'Malley, H. Viswanathan, and N. Lubbers, "Development of the Senseiver for efficient field reconstruction from sparse observations," *Nature Machine Intelligence* **5**, 1317–1325 (2023).
- ¹¹E. H. Krath, F. L. Carpenter, and P. G. A. Cizmas, "Prediction of unsteady flows in turbomachinery cascades using proper orthogonal decomposition," *Physics of Fluids* **36**, 037108 (2024).
- ¹²G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual Review of Fluid Mechanics* **25**, 539–575 (1993).

- ¹³M. Frank, D. Drikakis, and V. Charissis, "Machine-learning methods for computational science and engineering," *Computation* **8**, 15 (2020).
- ¹⁴N. Christakis and D. Drikakis, "Unsupervised learning of particles dispersion," *Mathematics* **11** (2023), 10.3390/math11173637.
- ¹⁵Z. Zhang, X. Gao, Q. Chen, and Y. Yuan, "A novel thermal turbulence reconstruction method using proper orthogonal decomposition and compressed sensing coupled based on improved particle swarm optimization for sensor arrangement," *Physics of Fluids* **36**, 055124 (2024).
- ¹⁶Y. Sha, Y. Xu, Y. Wei, and C. Wang, "Prediction of pressure fields on cavitation hydrofoil based on improved compressed sensing technology," *Physics of Fluids* **36**, 013321 (2024).
- ¹⁷P. Dubois, T. Gomez, L. Planckaert, and L. Perret, "Machine learning for fluid flow reconstruction from limited measurements," *Journal of Computational Physics* **448**, 110733 (2022).
- ¹⁸A. G. Ozbay and S. Laizet, "Deep learning fluid flow reconstruction around arbitrary two-dimensional objects from sparse sensors using conformal mappings," *AIP Advances* **12**, 045126 (2022).
- ¹⁹C. Kong, J.-T. Chang, Y.-F. Li, and R.-Y. Chen, "Deep learning methods for super-resolution reconstruction of temperature fields in a supersonic combustor," *AIP Advances* **10**, 115021 (2020).
- ²⁰A. Khan, A. Sohail, U. Zahoor, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artificial Intelligence Review* **53**, 5455–5516 (2020).
- ²¹K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016) pp. 770–778.
- ²²T. Falk, D. Mai, R. Bensch, O. Cicek, A. Abdulkadir, Y. Marrakchi, A. Bohm, J. Deubner, Z. Jackel, K. Seiwald, A. Dovzhenko, O. Tietz, C. Dal Bosco, S. Walsh, D. Saltukoglu, T. L. Tay, M. Prinz, K. Palme, M. Simons, I. Diester, T. Brox, and O. Ronneberger, "U-Net: Deep learning for cell counting, detection, and morphometry," *Nature Methods* **16**, 67–70 (2019).
- ²³M. Morimoto, K. Fukami, K. Zhang, A. G. Nair, and K. Fukagata, "Convolutional neural networks for fluid flow analysis: toward effective metamodeling and low dimensionalization," *Theoretical and Computational Fluid Dynamics* **35**, 633–658 (2021).
- ²⁴K. Duraisamy, "Perspectives on machine learning-augmented Reynolds-averaged and large eddy simulation models of turbulence," *Phys. Rev. Fluids* **6**, 050504 (2021).
- ²⁵V. Mons, J.-C. Chassaing, T. Gomez, and P. Sagaut, "Reconstruction of unsteady viscous flows using data assimilation schemes," *Journal of Computational Physics* **316**, 255–280 (2016).
- ²⁶W. Weng, Y. Zhang, and Z. Xiong, "Boosting event stream super-resolution with a recurrent neural network," in *Computer Vision – ECCV 2022*, edited by S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner (Springer Nature Switzerland, Cham, 2022) pp. 470–488.
- ²⁷X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Computer Vision – ECCV 2018 Workshops*, edited by L. Leal-Taixé and S. Roth (Springer International Publishing, Cham, 2019) pp. 63–79.
- ²⁸J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)* (2017) pp. 2242–2251.
- ²⁹Z. Luo, L. Wang, J. Xu, M. Chen, J. Yuan, and A. C. C. Tan, "Flow reconstruction from sparse sensors based on reduced-order autoencoder state estimation," *Physics of Fluids* **35**, 075127 (2023).
- ³⁰M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics* **378**, 686–707 (2019).
- ³¹M. Lino, S. Fotiadis, A. A. Bharath, and C. D. Cantwell, "Current and emerging deep-learning methods for the simulation of fluid dynamics," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **479**, 20230058 (2023).
- ³²H. Gao, L. Sun, and J.-X. Wang, "Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels," *Physics of Fluids* **33**, 073603 (2021).
- ³³L. Sun, H. Gao, S. Pan, and J.-X. Wang, "Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data," *Computer Methods in Applied Mechanics and Engineering* **361**, 112732 (2020).
- ³⁴X. Zhao, X. Chen, Z. Gong, W. Zhou, W. Yao, and Y. Zhang, "RecFNO: A resolution-invariant flow and heat field reconstruction method from sparse observations via fourier neural operator," *International Journal of Thermal Sciences* **195**, 108619 (2024).
- ³⁵J. Liu, C. Chen, J. Tang, and G. Wu, "From coarse to fine: Hierarchical pixel integration for lightweight image super-resolution," *Proceedings of the AAAI Conference on Artificial Intelligence* **37**, 1666–1674 (2023).
- ³⁶C. Tian, Y. Xu, W. Zuo, B. Zhang, L. Fei, and C.-W. Lin, "Coarse-to-fine CNN for image super-resolution," *IEEE Transactions on Multimedia* **23**, 1489–1502 (2021).
- ³⁷K. Zhang, D. Tao, X. Gao, X. Li, and J. Li, "Coarse-to-fine learning for single-image super-resolution," *IEEE Transactions on Neural Networks and Learning Systems* **28**, 1109–1122 (2017).
- ³⁸F. Sofos, D. Drikakis, I. W. Kokkinakis, and S. M. Spottswood, "A deep learning super-resolution model for turbulent image upscaling and its application to shock wave–boundary layer interaction," *Physics of Fluids* **36**, 025117 (2024).
- ³⁹F. Sofos, D. Drikakis, and I. W. Kokkinakis, "Deep learning architecture for sparse and noisy turbulent flow data," *Physics of Fluids* **36**, 035155 (2024).
- ⁴⁰C. Jiang, Y. C. Soh, and H. Li, "Two-stage indoor physical field reconstruction from sparse sensor observations," *Energy and Buildings* **151**, 548–563 (2017).
- ⁴¹S. D. Prasetyo, E. P. Budiana, A. R. Prabowo, and Z. Arifin, "Modeling finned thermal collector construction nanofluid-based Al2O3 to enhance photovoltaic performance," *Civil Engineering Journal* **9**, 2989–3007 (2023), number: 12.
- ⁴²P. I. Oforji, E. J. Mba, and F. O. Okeke, "The effects of rhythm on building openings and fenestrations on airflow pattern in tropical low-rise residential buildings," *Civil Engineering Journal* **9**, 2062–2084 (2023), number: 8.
- ⁴³Y. A. Sazonov, M. A. Mokhov, A. V. Bondarenko, V. V. Voronova, K. A. Tumanyan, and E. I. Konyushkov, "Interdisciplinary studies of jet systems using Euler methodology and computational fluid dynamics technologies," *HighTech and Innovation Journal* **4**, 703–719 (2023), number: 4.
- ⁴⁴I. W. Kokkinakis, D. Drikakis, K. Ritos, and S. M. Spottswood, "Direct numerical simulation of supersonic flow and acoustics over a compression ramp," *Physics of Fluids* **32**, 066107 (2020).
- ⁴⁵I. Kokkinakis and D. Drikakis, "Implicit large eddy simulation of weakly-compressible turbulent channel flow," *Computer Methods in Applied Mechanics and Engineering* **287**, 229–261 (2015).
- ⁴⁶K. Karantonis, I. W. Kokkinakis, B. Thornber, and D. Drikakis, "Compressibility in suddenly expanded subsonic flows," *Physics of Fluids* **33**, 105106 (2021).
- ⁴⁷A. Mosedale and D. Drikakis, "Assessment of Very High Order of Accuracy in Implicit LES models," *Journal of Fluids Engineering* **129**, 1497–1503 (2007), https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/129/12/1497/5729131/1497_1.pdf.
- ⁴⁸D. Drikakis and P. K. Smolarkiewicz, "On spurious vortical structures," *Journal of Computational Physics* **172**, 309–325 (2001).
- ⁴⁹X.-D. Liu, S. Osher, and T. Chan, "Weighted essentially non-oscillatory schemes," *Journal of Computational Physics* **115**, 200–212 (1994).
- ⁵⁰G.-S. Jiang and C.-W. Shu, "Efficient implementation of weighted ENO schemes," *Journal of Computational Physics* **126**, 202–228 (1996).
- ⁵¹D. S. Balsara and C.-W. Shu, "Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy," *Journal of Computational Physics* **160**, 405–452 (2000).
- ⁵²E. F. Toro, M. Spruce, and W. Speares, "Restoration of the contact surface in the HLL-Riemann solver," *Shock Waves* **4**, 25–34 (1994).
- ⁵³R. Spiteri and S. Ruuth, "A new class of optimal high-order strong-stability-preserving time discretization methods," *SIAM Journal on Numerical Analysis* **40**, 469–491 (2002).
- ⁵⁴R. Yang, C. S. Ng, K. L. Chong, R. Verzicco, and D. Lohse, "Do increased flow rates in displacement ventilation always lead to better results?" *Journal of Fluid Mechanics* **932**, A3 (2022).
- ⁵⁵K. Ritos, D. Drikakis, and I. W. Kokkinakis, "Virus spreading in cruiser cabin," *Physics of Fluids* (2023), <https://doi.org/10.1063/5.0169992>.
- ⁵⁶K. Ritos, D. Drikakis, and I. W. Kokkinakis, "The effects of ventilation conditions on mitigating airborne virus transmission," *Physics of Fluids* **36**, 013322 (2024), https://pubs.aip.org/aip/pot/article-pdf/doi/10.1063/5.0185296/18626326/013322_1_5.0185296.pdf.

- ⁵⁷S. Vidas, P. Moghadam, and M. Bosse, "3d thermal mapping of building interiors using an rgb-d and thermal camera," in *2013 IEEE International Conference on Robotics and Automation* (2013) pp. 2311–2318.
- ⁵⁸O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, edited by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi (Springer International Publishing, Cham, 2015) pp. 234–241.
- ⁵⁹X. Peng, X. Li, Z. Gong, X. Zhao, and W. Yao, "A deep learning method based on partition modeling for reconstructing temperature field," *International Journal of Thermal Sciences* **182**, 107802 (2022).
- ⁶⁰P. Liao, W. Song, P. Du, and H. Zhao, "Multi-fidelity convolutional neural network surrogate model for aerodynamic optimization based on transfer learning," *Physics of Fluids* **33**, 127121 (2021).
- ⁶¹B. Neyshabur, H. Sedghi, and C. Zhang, "What is being transferred in transfer learning?" in *Advances in Neural Information Processing Systems*, Vol. 33, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin (Curran Associates, Inc., 2020) pp. 512–523.
- ⁶²K. Fukami, R. Maulik, N. Ramachandra, K. Fukagata, and K. Taira, "Global field reconstruction from sparse sensors with Voronoi tessellation-assisted deep learning," *Nature Machine Intelligence* **3**, 945–951 (2021).
- ⁶³Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing* **13**, 600–612 (2004).
- ⁶⁴F. Sofos, D. Drikakis, I. W. Kokkinakis, and S. M. Spottswood, "Convolutional neural networks for compressible turbulent flow reconstruction," *Physics of Fluids* **35**, 116120 (2023).
- ⁶⁵J. Demongeot, A. Hamie, A. Laksaci, and M. Rachdi, "Relative-error prediction in nonparametric functional statistics: Theory and practice," *Journal of Multivariate Analysis* **146**, 261–268 (2016), special Issue on Statistical Models and Methods for High or Infinite Dimensional Spaces.
- ⁶⁶K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," *Journal of Fluid Mechanics* **870**, 106–120 (2019).
- ⁶⁷K. Fukami, K. Hasegawa, T. Nakamura, M. Morimoto, and K. Fukagata, "Model order reduction with neural networks: Application to laminar and turbulent flows," *SN Computer Science* **2**, 467 (2021).
- ⁶⁸K. Bao, X. Zhang, W. Peng, and W. Yao, "Deep learning method for super-resolution reconstruction of the spatio-temporal flow field," *Advances in Aerodynamics* **5**, 19 (2023).
- ⁶⁹A. Gumes, S. Discetti, A. Ianiro, B. Sirmacek, H. Azizpour, and R. Vinuesa, "From coarse wall measurements to turbulent velocity fields through deep learning," *Physics of Fluids* **33**, 075121 (2021).
- ⁷⁰C.-W. Shu, "Numerical experiments on the accuracy of ENO and modified ENO schemes," *Journal of Scientific Computing* **5**, 127–149 (1990).
- ⁷¹C.-W. Shu, "Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws," in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations: Lectures given at the 2nd Session of the Centro Internazionale Matematico Estivo (C.I.M.E.) held in Cetraro, Italy, June 23–28, 1997*, edited by A. Quarteroni (Springer Berlin Heidelberg, Berlin, Heidelberg, 1998) pp. 325–432.
- ⁷²C. Shu, "High order weighted essentially nonoscillatory schemes for convection dominated problems," *SIAM Review* **51**, 82–126 (2009).
- ⁷³A. K. Henrick, T. D. Aslam, and J. M. Powers, "Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points," *Journal of Computational Physics* **207**, 542–567 (2005).
- ⁷⁴E. M. Taylor, M. Wu, and M. P. Martin, "Optimization of nonlinear error for weighted essentially non-oscillatory methods in direct numerical simulations of compressible turbulence," *Journal of Computational Physics* **223**, 384–397 (2007).
- ⁷⁵O. Zanotti and M. Dumbser, "Efficient conservative ADER schemes based on WENO reconstruction and space-time predictor in primitive variables," *Computational Astrophysics and Cosmology* **3**, 1–32 (2016).
- ⁷⁶R. H. Nichols, R. W. Tramel, and P. G. Buning, "Evaluation of two high-order weighted essentially nonoscillatory schemes," *AIAA Journal* **46**, 3090–3102 (2008).
- ⁷⁷M. Wu and M. P. Martin, "Direct numerical simulation of supersonic turbulent boundary layer over a compression ramp," *AIAA Journal* **45**, 879–889 (2007).
- ⁷⁸H. Feng, C. Huang, and R. Wang, "An improved mapped weighted essentially non-oscillatory scheme," *Applied Mathematics and Computation* **232**, 453–468 (2014).
- ⁷⁹A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, "Uniformly high order accurate essentially non-oscillatory schemes, III," *Journal of Computational Physics* **131**, 3–47 (1997).
- ⁸⁰V. Titarev and E. Toro, "Finite-volume WENO schemes for three-dimensional conservation laws," *Journal of Computational Physics* **201**, 238–260 (2004).
- ⁸¹E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics, A Practical Introduction*, 3rd ed. (Springer-Verlag Berlin Heidelberg, 2009).
- ⁸²E. F. Toro, "A linearized Riemann solver for the time-dependent Euler equations of gas dynamics," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* **434**, 683–693 (1991).
- ⁸³E. F. Toro, "Direct Riemann solvers for the time-dependent Euler equations," *Shock Waves* **5**, 75 (1995).

Appendix A: WENO implementation

To address potential numerical instabilities due to the process of choosing an essentially non-oscillatory (ENO) stencil⁷⁰, Weighted ENO (WENO) methods were introduced^{49,50}. WENO schemes use a convex combination of all the ENO candidate stencils such that the numerical flux is approximated with the higher order of accuracy in smooth regions while still retaining the ENO property in the flow regions near discontinuities; see^{71,72} for an overview and further references. For WENO implementations on structured grids, when the solution is locally smooth enough, the convex combination of the stencils of a r^{th} -order ENO scheme results in a $(2r - 1)^{\text{th}}$ -order WENO scheme⁴⁹.

Aiming to achieve a balance between accuracy and stability, we enhance the WENO schemes of 3rd and 5th-order of Jiang and Shu⁵⁰ ($r = 2, 3$) and 7th, 9th and 11th-order of Balsara and Shu⁵¹ ($r = 4, 5, 6$) by combining the mapped WENO approach of Henrick, Aslam, and Powers⁷³ (WENO-M) and the relative total variation limiter approach of Taylor, Wu, and Martín⁷⁴ (WENO-RLTV). WENO-M recovers the loss of accuracy occurring near smooth critical points. WENO-RLTV reduces the numerical dissipation by using the optimal linear weights in sufficiently smooth regions instead of nonlinear linear smoothness-indicator-based weights. The numerical reconstruction can be performed at conservative, characteristic, or primitive variables. The reconstruction of the conservative variables is more common in the literature. However, past research has shown that such practice can lead to inaccuracies in capturing shock waves; see Zanotti and Dumbser⁷⁵ and references therein. Similar to other authors⁷⁶, we have opted to use the primitive variables in the high-order numerical reconstruction. The characteristics-based variables would be more expensive computationally.

We present below a detailed description of the WENO procedure implemented to obtain the left reconstruction, $q_{i+1/2}^L$, of the primitive variables, $q = [\rho, u, p]^T$, at cell face $i + 1/2$:

1. The *full* (left and right reconstruction) stencil ($S_{i+1/2}^G$) is normalized, per variable, according to the transform

mation function:

$$S_{i+1/2}^{Gz} = \frac{S_{i+1/2}^G - g_{\min}}{g_{\max}} \quad (\text{A1})$$

where

$$S_{i+1/2}^G = (q_{i-r+1}, \dots, q_{i+r})$$

and

$$g_{\min} = \min(S_{i+1/2}^G) - 1$$

$$g_{\max} = \max(S_{i+1/2}^G - q_{\min})$$

and the new k^{th} candidate stencil for the left reconstruction, containing r cell center values, is given by:

$$S_{i+1/2;k}^L = S_{i+1/2}^{Gz} [i-r+1+k, \dots, i+k]$$

where $k = 0, \dots, r-1$. Eq. (A1) normalizes the values of the candidate stencils before the estimation of the smoothness indicators (IS) in such a way that (i) the maximum value of the entire stencil becomes equal to one, i.e. $\max(S_{i+1/2}^{Gz}) = 1$, (ii) the minimum value takes a positive and nonzero value, i.e. $\min(S_{i+1/2}^{Gz}) > 0$, and (iii) the value range scales as initially relative to the maximum. By definition $g_{\max}$ is always positive and non-zero; hence Eq. (A1) will never result in an undefined operation and cause an exception. Using the above *normalization of the total stencil values*, per variable, is found to (i) prevent negative WENO smoothness indicator values, (ii) reduce numerical dissipation, and (iii) simplify the application of the proceeding step. The stencil normalization was found not to affect the MUSCL-type slope limiters.

- Next, a modified version of the relative total variation (TV) limiting procedure of Taylor, Wu, and Martín⁷⁴ is implemented. The TV of each k^{th} candidate stencil is calculated according to:

$$\text{TV}_k(S_{ik}) = \sum_{l=1}^{r-1} |q_{i-r+k+l+1} - q_{i-r+k+l}| \quad (\text{A2})$$

Eq. (A2) is then used to obtain the maximum TV ratio between the candidate stencils:

$$R(\text{TV}) = \frac{\max(\text{TV}_k)}{\min(\text{TV}_k) + \varepsilon} \quad (\text{A3})$$

If all of the stencils contain significant discontinuities, then the value of $R(\text{TV})$ can be incorrectly small, i.e. $R(\text{TV}) \approx 1$. Thus, an additional criterion is introduced to avoid such an occurrence. The linear weights are used provided the following two conditions are satisfied:

$$\text{if } [R(\text{TV}) < A_{\text{RL}}^{\text{TV}} \ \& \ \max(\text{TV}_k) < B_{\text{RL}}^{\text{TV}}] \text{ then}$$

$$\omega_k^r = C_k^r \quad (\text{A4})$$

According to Taylor, Wu, and Martín⁷⁴, $A_{\text{RL}}^{\text{TV}} = 5$, while for the second condition, $B_{\text{RL}}^{\text{TV}} = 0.2(r-1)$, where r is the order of the polynomials used in the $2(r-1)^{\text{th}}$ -order WENO. In practice, however, $r = 2$ is typically used. Note that the equation for $B_{\text{RL}}^{\text{TV}}$ is applicable only if the preceding pre-treatment/re-scaling of the candidate stencils is carried out; otherwise, it must be multiplied by $q_{\max}$. In essence, the second condition allows for an average TV of 20% between two neighbouring cells of the local stencils (S_i^G) maximum variable value, but this value can be modified if necessary. Wu and Martín⁷⁷ used a value of $B_{\text{RL}}^{\text{TV}} = 0.2$ for their 4th-order bandwidth-optimized WENO implementation in their DNS study.

Eq. (A4) assumes that for the linear weights the condition $\sum_{l=0}^{r-1} C_l^r = 1$ is always satisfied.

- If condition Eq. (A4) is not satisfied, nonlinear near weights based on the smoothness indicators of each candidate stencil are computed according to the following two steps:

$$\Omega_k^r = \frac{C_k^r}{(IS_k^r)^p + \varepsilon}, \quad \omega_k^r = \frac{\Omega_k^r}{\sum_{l=0}^{r-1} \Omega_l^r} \quad (\text{A5})$$

where $p = r$ and $\varepsilon = 10^{-41}$.

The standard WENO weights obtained in Eq. (A5) are modified according to the mapped WENO (WENO-M) approach of Henrick, Aslam, and Powers⁷³ as:

$$\tilde{\omega}_k^r = \frac{\tilde{\Omega}_k^r}{\sum_{l=0}^{r-1} \tilde{\Omega}_l^r} \quad (\text{A6})$$

where, using the alternate formulation of Feng, Huang, and Wang⁷⁸, the mapped weights are given by:

$$\tilde{\Omega}_k^r = C_k^r + \frac{(\Omega_k^r + C_k^r)^{K+1}A}{(\Omega_k^r - C_k^r)^KA + \Omega_k^r(1 - \Omega_k^r)} \quad (\text{A7})$$

and setting $A = 1$ and $K = 2$ results in the original mapping function⁷³.

- The reconstructed scaled variable value at the left-side of cell-face $i+1/2$ is given by:

$$q_{i+1/2}^L = \sum_{k=0}^{r-1} [\tilde{\omega}_k^r f(q)_k^r] \quad (\text{A8})$$

where

$$f(q)_k^r = \sum_{l=0}^{r-1} \alpha_{k,l}^r q_{i-r+k+l+1} \quad (\text{A9})$$

- Finally, due to the initial “normalizing” of the stencil in step 1, the reconstructed values obtained using Eq. (A8) need to be “re-scaled” according to:

$$q_{i+1/2}^L = q_{i+1/2}^L g_{\max} + g_{\min} \quad (\text{A10})$$

WENO reconstruction can lead to spurious oscillations if two or more shocks are too close to each other and WENO cannot choose a single smooth stencil. To remedy this problem, a procedure first introduced by Harten *et al.*⁷⁹ is adopted. If the reconstructed density and pressure values differ too drastically from their cell-center average values, the order of the WENO reconstruction is reduced. After completion of the left and right reconstruction procedures at cell-face $i + 1/2$, the left and right reconstructed density and pressure values are compared against their respective left and right cell-center values:

$$\begin{aligned} |\rho_{i+1/2}^L - \rho_i| &> C_{\bar{\rho}}^- \quad \text{or} \\ |\rho_{i+1/2}^R - \rho_{i+1}| &> C_{\bar{\rho}}^- \end{aligned} \quad (\text{A11})$$

where the order reduction threshold constant is set equal to $C_{\bar{\rho}}^- = 0.7$. If the condition in Eq. (A11) is met, then the order of the WENO scheme is reduced according to $(r - 1)$. The reconstruction procedure is then repeated for all variables, and the condition is rechecked. The process is repeated until Eq. (A11) is no longer satisfied. For example, assuming the condition is repeatably met, a 9th-order WENO would first reduce to 7th-order, then to 5th, 3rd, and finally to the 2nd-order MC MUSCL scheme. Titarev and Toro⁸⁰ showed that using the above procedure does not degrade the high order of accuracy for sufficiently smooth solutions.

Appendix B: HLLC approximate Riemann solver implementation

The Riemann problem is solved here using the so-called ‘‘Harten, Lax, van Leer, and (the missing) Contact’’ (HLLC) approximate Riemann solver of Toro, Spruce, and Speares⁵². More specifically, the adaptive non-iterative Riemann solver (ANRS) variant proposed by Toro⁸¹ (see §9.5.2) is implemented. The following sequence details the approximate HLLC Riemann solver procedure implemented:

1. To ensure high order near the boundaries for high-order FVM codes, typically, the ghost-cell method is used to apply the boundary conditions (BC). However, even after carefully programming the boundary conditions and reconstruction procedures, computer rounding errors can persist and give rise to differences between the left and right reconstructed states. Therefore, to ensure the appropriate flux, we modify the left and right reconstructed states for the following BCs: symmetry plane (inviscid wall), heated (constant temperature) wall, and adiabatic (zero heat-flux) viscous (no-slip) wall.

For a symmetry plane, the no penetration condition is implemented for both advective and acoustic waves using the procedure described by Algorithm 1.

In the case of a viscous wall, Algorithm 2 is used instead. For an isothermal wall, the temperature at the ghost cells is linearly interpolated from the interior domain and the wall. In this case, it is advisable to restrict

```

if Left BC Symmetry then
     $\rho_L = \rho_R;$ 
     $p_L = p_R;$ 
     $\mathbf{u}_L = \mathbf{u}_R - 2(\mathbf{u}_R \cdot \hat{\mathbf{n}})\hat{\mathbf{n}};$ 
else if Right BC Symmetry then
     $\rho_R = \rho_L;$ 
     $p_R = p_L;$ 
     $\mathbf{u}_R = \mathbf{u}_L - 2(\mathbf{u}_L \cdot \hat{\mathbf{n}})\hat{\mathbf{n}};$ 

```

Algorithm 1: Ensure symmetry BC flux in HLLC.

the interpolated temperature range of values to be only positive ($T \in \mathbb{R}_{>0}$), i.e. $T_{\text{ghosts}} > 10^{-15}$, which reduces the likelihood of a non-physical solution from manifesting.

```

if Left BC Viscous Wall then
     $p_L = p_R;$ 
    if Wall Temperature then  $\rho_R = p_R / (R_s T_W);$ 
     $p_L = p_R;$ 
     $\mathbf{u}_L = \mathbf{u}_R = 0;$ 
else if Right BC Viscous Wall then
     $p_R = p_L;$ 
    if Wall Temperature then  $\rho_L = p_L / (R_s T_W);$ 
     $p_R = p_L;$ 
     $\mathbf{u}_L = \mathbf{u}_R = 0;$ 

```

Algorithm 2: Ensure viscous wall BC flux in HLLC; if *Wall Temperature* true isothermal, else adiabatic.

2. An initial estimate of the pressure in the *Star Region*, that is, the region defined in-between the nonlinear near convective wave-speeds (or characteristics), can be obtained according to⁸²:

$$p_* = \max(0, p_{\text{pvrs}}) \quad (\text{B1})$$

which for curvilinear coordinates p_{pvrs} is obtained according to:

$$\begin{aligned} p_{\text{pvrs}} &= \frac{1}{2} \left[p_L + p_R + \left(u_L^\perp - u_R^\perp \right) \bar{\rho} \bar{s} \right] \\ \bar{\rho} &= (\rho_L + \rho_R) / 2, \quad \bar{s} = (s_L + s_R) / 2 \end{aligned} \quad (\text{B2})$$

where the speed of sound is defined as $s = \sqrt{\gamma p / \rho}$ and $u^\perp = \mathbf{u} \cdot \hat{\mathbf{n}}$ is the magnitude of the velocity normal to the cell-face.

The ‘‘averaged’’ value of p_* given by Eq. (B1) is enhanced by taking into account the local conditions. The ANRS approach⁸¹ introduces two conditions as a means to avoid unnecessary computations, i.e. updating the value of p_* obtained by Eq. (B1) with one that is more accurate. The first condition requires that the ratio between the maximum and minimum local reconstructed pressures is more significant than a predetermined constant, i.e.

$$Q = p_{\text{max}} / p_{\text{min}} > Q_{\text{user}}$$

where $p_{\min} = \min(p_L, p_R)$, $p_{\max} = \max(p_L, p_R)$ and it is recommended that $Q_{\text{user}} = 2$. The other condition requires that p_* does not lie between $p_{\min}$ and $p_{\max}$, i.e. $p_* < p_{\min}$ or $p_* > p_{\max}$. However, similarly to non-differentiable (reconstruction) limiters, they hinder convergence. Instead, the following relation is used immediately after Eq. (B1):

$$p_* = \begin{cases} \left[\frac{s_L + s_R - \frac{\gamma-1}{2}(u_R^\perp - u_L^\perp)}{s_L/p_L^z + s_R/p_R^z} \right]^{\frac{1}{z}} & \text{if } p_* \leq p_{\min} \quad (\text{TRRS}) \\ \frac{g_L p_L + g_R p_R - (u_R^\perp - u_L^\perp)}{g_L + g_R} & \text{if } p_* > p_{\max} \quad (\text{TSRS}) \end{cases}$$

where

$$z = \frac{\gamma-1}{2\gamma}, \quad A_K = \frac{2}{(\gamma-1)p_K}, \quad B = \left(\frac{\gamma-1}{\gamma+1} \right) p_K$$

$$g_K = \left(\frac{A_K}{p + B_K} \right)^{1/2}, \quad K = L, R$$

The abbreviations TRRS⁸¹ and TSRS^{81,83} stand for the

Two-Rarefaction Riemann Solver and Two-Shock Riemann Solver, respectively.

3. Next, we compute the wave-speed estimates according to Toro, Spruce, and Speares⁵²:

$$\begin{aligned} S_L &= u_L^\perp - s_L q_L \\ S_R &= u_R^\perp + s_R q_R \end{aligned} \quad (\text{B3})$$

where

$$q_K = \begin{cases} 1 & \text{if } p_* \leq p_K \\ \left[1 + \frac{\gamma+1}{2\gamma} \left(\frac{p_*}{p_K} - 1 \right) \right]^{1/2} & \text{if } p_* > p_K \end{cases} \quad (\text{B4})$$

Eq. (B3) slightly increases the numerical diffusion as it permits a greater range but has favourable stability in very high-speed flows and particularly near strong shock-waves.

Using the above, the intermediate “missing” wave-speed, S_* , and associated HLLC fluxes are computed according to Toro⁸¹.