Machine learning for the prediction of the local skin friction factors and Nusselt numbers in turbulent flows past rough surfaces

Sanhueza, Rafael Diez; Akkerman, Ido; Peeters, Jurriaan W.R.

DOI
10.1016/j.ijheatfluidflow.2023.109204

Publication date
2023

Document Version
Final published version

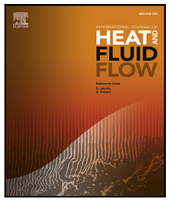
Published in
International Journal of Heat and Fluid Flow

International Journal of Heat and Fluid Flow 103 (2023) 109204

Contents lists available at ScienceDirect

International Journal of Heat and Fluid Flow

journal homepage: www.elsevier.com/locate/ijhff



Machine learning for the prediction of the local skin friction factors and Nusselt numbers in turbulent flows past rough surfaces

Rafael Diez Sanhueza *, Ido Akkerman, Jurriaan W.R. Peeters

Faculty of Mechanical, Materials and Maritime Engineering, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, The Netherlands

ARTICLE INFO

Keywords:
Turbulence
Rough surfaces
Machine learning

ABSTRACT

Turbulent flows past rough surfaces can create substantial energy losses in engineering equipment. During the last decades, developing accurate correlations to predict the thermal and hydrodynamic behavior of rough surfaces has proven to be a difficult challenge. In this work, we investigate the applicability of convolutional neural networks to perform a direct image-to-image translation between the height map of a rough surface and its detailed local skin friction factors and Nusselt numbers. Additionally, we propose the usage of separable convolutional modules to reduce the total number of trainable parameters, and PReLU activation functions to increase the expressivity of the neural networks created. Our final predictions are improved by a new filtering methodology, which is able to combine the results of multiple neural networks while discarding non-physical oscillations likely caused by over-fitting. The main study is based on a new DNS database formed by 80 flow cases at a friction Reynolds number of $Re_\tau = 180$ obtained by applying random shifts to the Fourier spectrum of the grit-blasted surface originally scanned by Busse et al. (2015). The results show that machine learning can accurately predict the skin friction values and Nusselt numbers for a rough surface. A detailed comparison with existing correlations in the literature revealed that the maximum errors generated by deep learning were only 8.1% for the global skin friction factors $\overline{C_f}$ and 2.9% for the Nusselt numbers $\overline{Nu}$, whereas the best classical correlations identified reached errors of 24.9% and 13.5% for $\overline{C_f}$ and $\overline{Nu}$ respectively. The deep learning results also proved stable with respect to rough surfaces with abrupt changes in their roughness elements, and only presented a minor sensitivity with respect to variations in the dataset size.

1. Introduction

Turbulent flows past rough surfaces can be found in a large variety of engineering applications. Irregular surfaces are often caused by external processes, such as bio-fouling, abrasion, machining, or corrosion. In most engineering applications, the presence of rough surfaces can substantially increase the drag resistance of transportation systems, and lower the efficiency of thermodynamic cycles. Despite these drawbacks, some rough surfaces under specific flow conditions can yield positive contributions. For example, certain rough surfaces can enhance heat transfer significantly, while having only a relatively small increase in pressure losses (Dipprey and Sabersky, 1963; Nilpueng and Wongwises, 2015; Ventola et al., 2014). Other surfaces were shown to reduce the total drag resistance of a turbulent flow (Golovin et al., 2016; Gong et al., 2021). One of the main challenges while working with rough surfaces is to predict the impact of a given surface topography on the drag resistance and heat transfer rates. Most correlations available in the literature predict the global skin friction factor $\overline{C_f}$ or the Stanton number $\overline{St}$ of a rough surface based on standard surface metrics, such as the root-mean-squared height variations, skewness, kurtosis, effective slope, forward-facing angles, or different auto-correlation functions (Napoli et al., 2008; Schultz and Flack, 2009; Thakkar et al., 2017; Jouybari et al., 2021). In this work, the operator $\overline{(...)}$ refers to the average across the whole surface. While traditional surface metrics offer a simplified framework to describe the geometry of rough surfaces, it has been shown in the literature that obtaining accurate predictions for the drag resistance of a generic rough surface remains an open challenge (Flack, 2018; Chung et al., 2021). The correlations found in the literature tend to be valid only for specific datasets, and the trends observed cannot be extrapolated to rough surfaces with widely varying characteristics. Moreover, several surface metrics, such as the skewness or the kurtosis, present mixed results across different studies (Flack and Schultz, 2010; Jouybari et al., 2021; De Marchis et al., 2019). As a result, a small group of surface features that produces a universal collapse of all known measurements for $\overline{C_f}$ or $\overline{St}$ with a low degree of dispersion has not been identified. One important reason behind the previous difficulties is that both

* Corresponding author.
E-mail address: R.G.DiezSanhueza-1@tudelft.nl (R.D. Sanhueza).

https://doi.org/10.1016/j.ijheatfluidflow.2023.109204
Received 12 January 2023; Received in revised form 14 June 2023; Accepted 23 July 2023
Available online 7 August 2023

Fig. 1. Example of two rough surfaces with identical roughness elements and averaged local quantities, such as their skewness or their kurtosis, but different hydrodynamic behavior (Forooghi et al., 2018).

the unique shape of roughness elements and their spatial organization affect turbulent flow fields differently (De Marchis et al., 2019; Yuan and Piomelli, 2014; Forooghi et al., 2018). The previous factors cannot be accounted for using traditional surface metrics, such as the skewness or the kurtosis, since these metrics do not encode information regarding the spatial organization of the roughness elements. An example of this issue can be found in Fig. 1, where two irregular surfaces with identical roughness elements located at different positions are presented; both rough surfaces will have identical skewness or effective slope, but they will have a substantially different hydrodynamic behavior due to the alignment of the roughness elements (Forooghi et al., 2018).

Machine learning (ML) may provide a robust alternative to traditional modeling for the hydrodynamic behavior of rough surfaces. Machine learning for fluid mechanics has already proven useful in other applications, such as correcting existing RANS turbulence models (Parish and Duraisamy, 2016) or predicting the Reynolds stress tensor (Sandberg and Zhao, 2022; Ling et al., 2016; Weatheritt and Sandberg, 2016). In the context of rough surface modeling, Jouybari et al. (2021) investigated the possibility to utilizing neural networks and Gaussian process regression to predict the equivalent Nikuradse sand-grain height k_s of rough surfaces based on traditional surfaces metrics. The parameter k_s corresponds to a length-scale that describes the hydrodynamic behavior of rough surfaces, and it can be used in combination with other correlations to predict the skin friction factor of a pipeline or channel flow (Moody, 1944; Nikuradse, 1933). However, correlations based on k_s can only predict the global behavior of a rough surface ($\overline{C_f}$ or $\overline{Nu}$), and thus they do not provide information regarding the local behavior of turbulence.

In this work, we investigate the applicability of machine learning systems based on convolutional neural networks to predict the local behavior of turbulent flows past irregular surfaces. These systems present several advantages over traditional correlations, such as being able to process the input height map H(x,z) of a rough surface directly, and to perform predictions without requiring traditional surface metrics. Deep learning systems can perform predictions taking into account both the detailed shape of every roughness element, as well as their spatial distribution. Furthermore, deep learning systems can be modified to not only predict global quantities, such as $\overline{C_f}$ or $\overline{St}$, but also distributed local quantities. Additionally, ML systems can perform direct predictions for the thermal behavior of rough surfaces, without requiring previous estimations for C_f that could lead to additional uncertainty.

This paper is organized as follows. In Section 2, the details of the DNS database are presented, including the methodology used to generate rough surfaces, the formulation of the CFD solver and all relevant physical parameters. Section 2.5 describes the numerical routine used to interpolate the wall forces and heat fluxes generated by a DNS solver using staggered grids and the immersed boundary method in CFD. In Section 3, the details of the machine learning systems developed are presented, as well as techniques used to improve the baseline architecture identified. In Section 4, the final results of the machine learning study are described, together with a comparison between the current results and traditional correlations found in the literature. In Section 5, the conclusions of the study are presented, along with future recommendations.

2. DNS database

2.1. Methodology for the DNS simulations

The methodology used to simulate turbulent flows past rough surfaces is based on the DNS framework described by Peeters and Sandham (2019). This formulation starts by considering a simple planar channel flow geometry to study the hydrodynamic behavior of rough surfaces. A schematic representation of the planar channel geometry with rough walls is shown in Fig. 2. Here, the (x, y, z) coordinates correspond to the streamwise, wall-normal and spanwise directions respectively. Only one half of the computational domain is presented, since the upper section of the channel corresponds to a reflection of the bottom part with respect to the symmetry plane located at the channel half-height $y = \delta$. The height function $H(x, z)$ shown in Fig. 2 is defined such that $\int_{x,z} H(x, z)\, dx\, dz = 0$ across the entire domain. Periodic boundary conditions are considered in the $x - z$ directions in order to close the computational domain. The incompressible Navier–Stokes equations and the energy equation for fluids are solved in dimensionless form,

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + Re_\tau^{-1} \nabla^2 \mathbf{u} + S_f, \tag{2}$$

$$\partial_t T + \mathbf{u} \cdot \nabla T = Pe_\tau^{-1} \nabla^2 T + S_q. \tag{3}$$

In Eqs. (1)–(3), the variables $\mathbf{u}$, p, T correspond to the dimensionless velocity, the pressure and the temperature fields respectively. The parameter $Re_\tau = u_\tau \delta / \nu$ found in the momentum equations is the friction Reynolds number, which is defined using the average friction velocity u_τ at the rough walls, the channel half-height δ and the kinematic viscosity ν. The scaling factor $Pe_\tau = Re_\tau Pr$ corresponds to the friction Peclet number, where Pr is the molecular Prandtl number. The parameter S_f found in Eq. (2) is a constant pressure-gradient term to force fluid motion, whereas S_q in Eq. (3) is a constant volumetric heat source term to induce thermal gradients in the fluids. The usage of constant source terms in turbulent channel flows, such as S_f or S_q, has been extensively validated during the last decades (Peeters and Sandham, 2019; Busse et al., 2015; Thakkar et al., 2017; Orlandi and Leonardi, 2006; Kim et al., 1987). In order to account for the presence of rough surfaces $H(x, z)$ inside the computational domain, the DNS solver uses an implementation of the immersed boundary method based on the direct-forcing approach proposed by Fadlun et al. (2000). In the first grid point inside the fluid domain, the velocity $\mathbf{u}$ and the temperature T are enforced by quadratic interpolation in the wall-normal direction. A detailed validation of this numerical method can be found in Peeters and Sandham (2019). While the DNS simulations are solved using dimensionless variables, it is important to note that the velocity and temperature fields are originally scaled by the friction velocity $u_\tau = \sqrt{\tau_w / \rho}$ and the friction temperature $T_\tau = q_w / (\rho c_p u_\tau)$. Here, the properties ρ and c_p refer to the fluid density and the specific heat capacity respectively, while the variables τ_w and q_w are the equivalent shear stresses and heat fluxes with respect to a smooth wall configuration.

During all DNS simulations, it is assumed that the reference variables Pr, S_f and S_q are equal to unity. Due to global momentum and energy conservation, this further implies that the equivalent shear stresses τ_w and heat fluxes q_w also have unitary values: $\tau_w = q_w = 1$. Therefore, the present DNS simulations have well-defined momentum and energy balances. The goals of the DNS simulations are to quantify the variations in the bulk flow properties of turbulent flows, the changes in their boundary layer parameters, and the distribution of their local skin friction factors $C_f(x, z)$ and Nusselt numbers $Nu(x, z)$

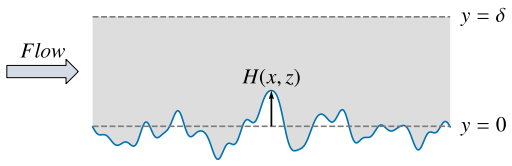


Fig. 2. Schematic representation of the planar channel flow geometry considered for the DNS simulations. The variable $H(x,z)$ corresponds to the height function of the rough surfaces, which is defined based on the average plane at $y = 0$. The parameter δ is the half-channel height: $\delta = L_y/2$.

across the rough surfaces. The formulas used to compute the values of $C_f(x,z)$ and $Nu(x,z)$ in Table 1 are the following,

$$C_f(x,z) = \frac{f_x(x,z)}{\frac{1}{2}\,\rho U_b^2}, \tag{4}$$

$$Nu(x,z) = \frac{q(x,z)}{\lambda\,(T_b - T_w)/L_y}. \tag{5}$$

In Eqs. (4)–(5), the variables U_b and T_b correspond to the bulk velocity of the fluid and its bulk temperature respectively. The bulk Reynolds number is defined as $Re_b = U_b L_y/\nu$, where L_y corresponds to the channel size and $\nu = 1/Re_\tau$ is the kinematic viscosity considered for all DNS simulations. The parameters $\lambda = 1/(Re_\tau\ Pr)$ and $T_w = 0$ are the thermal conductivity for each fluid and the Dirichlet temperature boundary condition imposed at the rough walls respectively. The variables $f_x(x,z)$ and $q(x,z)$ are the local forces and heat fluxes per unit of area distributed across the rough surfaces.

As a reference, the local Stanton numbers $St(x,y)$ distributed across the rough surfaces are defined as,

$$St(x,z) = \frac{Nu(x,z)}{Re_b\ Pr}. \tag{6}$$

In principle, it is possible to reconstruct the thermal behavior of turbulent flows past rough surfaces by considering either the local Nusselt numbers $Nu(x,z)$ or Stanton numbers $St(x,z)$. However, deep learning systems designed to reconstruct the local Nusselt numbers $Nu(x,z)$ can predict the bulk temperature of a fluid T_b directly if the average heat flux $\overline{q}$ acting over the boundaries is known. This marks a large contrast with respect to deep learning systems designed to predict the local Stanton numbers $St(x,z)$, since these systems require explicit information regarding the bulk velocity of a fluid U_b before predicting its bulk temperature T_b. Therefore, predictions for T_b can only be obtained after coupling deep learning systems trained to reconstruct the local Stanton numbers $St(x,z)$ with an additional ML system to estimate U_b, which increases the uncertainty of the final predictions. Due to the previous reason, the current study focuses on predicting the local skin friction factors $C_f(x,z)$ and Nusselt numbers $Nu(x,z)$ for all the rough surfaces considered.

For every DNS simulation, the domain size has dimensions $(L_x \times L_y \times L_z) = (5.63 \times 2 \times 2.815)$ in the streamwise, vertical and spanwise directions respectively. The half-channel size is thus assumed to be equal to unity in all simulations: $\delta = L_y/2 = 1$. The reference length scale chosen for the rough surfaces (k) has a value ranging from $k/\delta = 16.12\%$ to 17.44% across all the rough surfaces considered for simulation. This length scale corresponds to the mean-peak-to-valley height ($S_{z,5\times5}$) defined by Thakkar et al. (2017). Based on this configuration, the grid size considered by Peeters and Sandham (2019) was employed during the current study, which has dimensions $(N_x \times N_y \times N_z) = (280 \times 280 \times 140)$ for a friction Reynolds number of $Re_\tau = 180$. This grid size further ensures that approximately 12 grid points are used to discretize the smallest wavelength found in the definition of the height function $H(x,z)$ using a Fourier spectrum (Peeters and Sandham, 2019; Thakkar et al., 2017). The global statistics for the DNS simulations are presented later in Section 2.4, after the different types of rough surfaces considered in the study have been defined.

2.2. Categories of rough surfaces

In order to create machine learning models, a DNS database was generated with 80 flow cases simulated with a friction Reynolds number of $Re_\tau = 180$. Numerical simulations are used instead of experimental measurements, since the machine learning formulation requires detailed information regarding the local skin friction factors $C_f(x,z)$ and Nusselt numbers $Nu(x,z)$ distributed across the rough surfaces. All the rough surfaces considered are generated by introducing phase shift variations ϕ_i to the Fourier spectrum of the grit-blasted originally surface scanned by Busse et al. (2015). The height function $H(x,z)$ considered for the rough surfaces is the following,

$$H(x,z) = \sum_i R_i\ \cos\left(2\pi\left(x\,\frac{M_i}{L_x} + z\,\frac{N_i}{L_z}\right) - \phi_i\right). \tag{7}$$

In Eq. (7), the terms R_i, M_i and N_i correspond to constants extracted from the Fourier spectrum defined by Busse et al. (2015) in order to represent the grit-blasted surface scanned. Based on this formulation, the DNS database was generated by considering two categories of rough surfaces with a different methodology to define the phase shift component ϕ_i of every rough surface. The first group of rough surfaces, named Category I, was generated by considering 40 rough surfaces with random phase shift variations ϕ_i according to Eq. (7). Despite the simplicity of the methodology used, the rough surfaces found in Category I contain different types of morphological features and clusters of roughness elements, as it can be observed in the left side of Fig. 3. Therefore, predicting the local skin friction factors $C_f(x,z)$ and Nusselt numbers $Nu(x,z)$ generated by each rough surface is a challenging machine learning task, where it is necessary to predict the impact of each cluster of unique roughness elements in the local turbulent flow fields. The rough surfaces found in Category I are the main database used to train deep learning models during the current study.

The second group of rough surfaces, named Category II, corresponds to a special selection of rough surfaces with unusually high slope angles in the streamwise direction $\alpha_x = tan^{-1}(\partial H/\partial x)$. These rough surfaces serve as a challenging validation scenario to test the robustness of the deep learning systems trained, since they contain roughness elements with abrupt changes in their shape, as well as substantial modifications in their local turbulent flow fields. The methodology used to generate these rough surfaces is presented in Section 2.3.

A graphical comparison between the height maps $H(x,z)$ and the gradients $\partial H/\partial x$ for the rough surfaces contained in Categories I and II can be found in Fig. 3. The rough surfaces found in Category II contain regions with higher gradients than the examples shown from Category I. In Fig. 4, a detailed histogram is presented regarding the distribution of $|\partial H/\partial x|$ gradients for all rough surfaces belonging to Categories I and II of the DNS database. According to the distributions shown, the rough surfaces from Category II contain significantly steeper slopes than any sample from Category I. Therefore, the rough surfaces from Category II correspond to outlier cases with large distortions in their roughness elements, which can be used to test the robustness of the deep learning system to adverse conditions. Although it would be possible to include surfaces from Category II in the training sets, no significant improvements were observed in the validation performance during a preliminary study. Therefore, the surfaces from Category II are used exclusively to test the performance of the neural network for rough surfaces with features that fall outside Category I of the training sets.

2.3. Generation of rough surfaces with high slope in the streamwise direction

The rough surfaces found in Category II of the DNS database were generated by using a methodology specifically designed to detect rough surfaces with high slope in the streamwise direction $\partial H/\partial x$ across multiple regions of the DNS domain. The methodology starts by considering

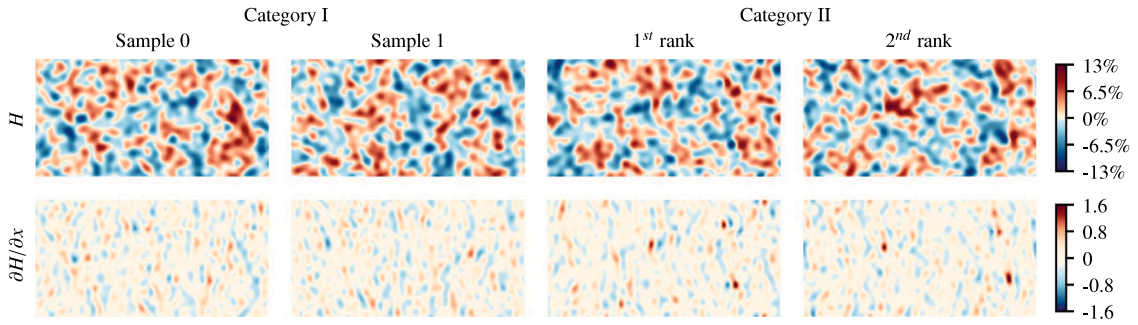


Fig. 3. Examples of rough surfaces belonging to Categories I and II of the DNS database. All height percentages are scaled with respect to the half-channel height $\delta = L_y/2$. The magnitude of the gradients $\partial H/\partial x$ is based on the reference length scales defined for the DNS simulations. Each height map presented corresponds to the entire DNS domain, which is periodic in the streamwise and spanwise directions.

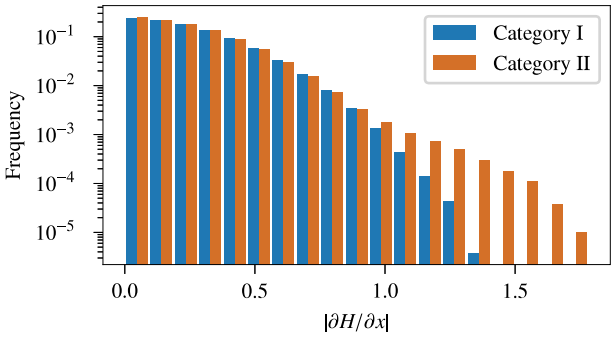


Fig. 4. Distribution of local $|\partial H/\partial x|$ gradients for rough surfaces belonging to Categories I and II of the DNS database.

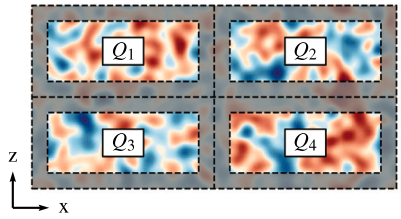


Fig. 5. Quadrants used to evaluate the absolute value of $|\partial H/\partial x|$ while selecting rough surfaces belonging to Category II of the DNS database. In the diagram shown, the horizontal and the vertical axis correspond to the streamwise and spanwise directions respectively.

a collection of 1,000,000 rough surfaces, which contain purely random phase shift variations ϕ according to the definition of the height function $H(x, z)$ back in Eq. (7). Then, all rough surfaces are ranked using an evaluation metric $\mathcal{M}$, which is defined using the highest values for $|\partial H/\partial x|$ found in every quadrant shown in Fig. 5. The gray zones between the quadrants in Fig. 5 correspond to regions where the values for $|\partial H/\partial x|$ are not evaluated. This helps the algorithm to avoid selecting rough surfaces where a single peak can represent the highest value for $|\partial H/\partial x|$ across multiple quadrants, since the idea is to obtain surfaces with multiple sharp roughness elements. The formula for the evaluation metric $\mathcal{M}$ is the following,

$$\mathcal{M} = \frac{\prod_i Q_i}{min(Q_i)}. \tag{8}$$

In Eq. (8), the variables Q_i correspond to the largest values for $|\partial H/\partial x|$ found in every quadrant of Fig. 5. The term $min(Q_i)$ found in the denominator of Eq. (8) is used to create an evaluation metric, which effectively only depends on the three highest values registered for Q_i. This allows the algorithm to select rough surfaces that might also contain one quadrant with smooth roughness elements, and thus it helps to increase the diversity of the surfaces selected. Despite the previous fact, the product term $\prod_i Q_i$ found in Eq. (8) implies that the remaining three quadrants of the surfaces must contain sharp roughness elements, or that at least one of the peaks found in the rough surfaces has very large $|\partial H/\partial x|$ values. Other formulas were also tested as a replacement for Eq. (8), such as $\mathcal{M} = \sum_i Q_i - min(Q_i)$. However, the empirical results proved that many of top-ranked surfaces found by both methodologies corresponded to the same samples, and that the remaining surfaces had a similar morphology.

2.4. Bulk flow properties and boundary layer parameters for the DNS database

In Table 1, the variations in the bulk flow properties and boundary layer parameters for the DNS simulations found in Categories I and II are presented. Here, it can be noted that all DNS cases for each category only present minor differences in the parameters being analyzed . However, the rough surfaces found in Category II present higher maximum values for all turbulent flow parameters, such as $\overline{C_f}$, $\overline{Nu}$, ΔU^+, ΔT^+. This implies that the distorted roughness elements found in these surfaces are able to produce minor changes in turbulence globally. The slightly lower values for the bulk Reynolds numbers Re_b found in rough surfaces from Category II can be explained since all DNS simulations are performed using a fixed momentum source term $S_f = 1$. This implies that rough surfaces which further enhance turbulence, and thus drag resistance, can reach the same target forces using slightly lower bulk velocities U_b, as it is observed in Table 1.

For more information, Appendix B presents detailed sub-plots for each quantity reported in Table 1, as well as the values of different traditional surface metrics. Regarding these results, it must be noted that all the rough surfaces found in our dataset have a constant the root-mean-squared height of 0.0358. This implies that the phase shift components ϕ_i of the height function $H(x, z)$ do not influence its root-mean-squared value when integrated inside a periodic domain. However, other surface metrics contain substantially different values. For example, the skewness of the rough surfaces varies up to 662% with respect to its median value within the dataset.

The parameter $k^+_{s,eq}$ listed in Table 1 corresponds to the dimensionless Nikuradse sand-grain roughness height for the irregular surfaces considered. Please note that the flows in the current DNS simulations are in the transitionally rough regime. Obtaining the equivalent sand-grain roughness size k_s for all DNS cases would require many more simulations in order to find the fully-rough asymptote for each surface. Instead, to find $k^+_s = k_s u_\tau/\nu$, we compare our results with Nikuradse's equations (Nikuradse, 1933) directly, using the time-averaged velocity $U^+_c = U_c/u_\tau$ obtained from the DNS simulations at the channel center ($y = \delta$):

$$U^+_c - 5.75\ \log_{10}\left(\delta/k_s\right) = A(k^+_{s,eq}), \tag{9}$$

Table 1
Variations in the bulk flow properties and boundary layer parameters for the DNS simulations present in Categories I and II. All variations are calculated with respect to the middle value for each range: $(max - min)/((min + max)/2)$.

Variable	Category I			Category II		
	Min.	Max.	Range	Min.	Max.	Range
$k^+_{s,eq}$	25.30	35.15	32.59%	25.49	37.12	37.15%
U_b	10.69	11.54	7.68%	10.49	11.53	9.44%
Re_b	3848	4156	7.68%	3776	4150	9.44%
$\overline{C_f}$	0.0150	0.0175	15.34%	0.0150	0.0182	18.83%
T_b	12.14	12.84	5.65%	12.04	12.81	6.15%
$\overline{Nu}$	28.02	29.66	5.65%	28.12	29.9	6.15%
$\overline{St}$	0.0068	0.0077	13.03%	0.0068	0.0079	15.54%
ΔU^+	4.09	5.07	21.46%	4.14	5.26	23.91%
ΔT^+	3.63	4.47	20.58%	3.68	4.55	20.95%

$$A(k^+_{s,eq}) = \begin{cases} 5.5 + 5.75\ \log_{10}(k^+_{s,eq}) & \text{if } k^+_{s,eq} < 3.55 \\ 6.59 + 3.5\ \log_{10}(k^+_{s,eq}) & \text{if } 3.55 \le k^+_{s,eq} < 7.08 \\ 9.58 & \text{if } 7.08 \le k^+_{s,eq} < 14.13 \\ 11.5 - 1.62\ \log_{10}(k^+_{s,eq}) & \text{if } 14.13 \le k^+_{s,eq} < 67.61 \\ 8.48 & \text{if } 67.61 \le k^+_{s,eq} \end{cases} \tag{10}$$

Eqs. (9)–(10) yield a 2.3% error when compared to the findings of Thakkar et al. (2018) for a grit-blasted surface with $k^+_s = 26.1$.

2.5. Wall force and heat flux interpolation

In order to calculate the local skin friction factors $C_f(x, z)$ and Nusselt numbers $Nu(x, z)$ acting over irregular surfaces, a post-processing routine was first developed to estimate the local forces $\mathbf{F}$ and heat fluxes Q distributed across the rough walls. This step is necessary, since the DNS solver uses a staggered grid to represent the velocity components u_x, u_y, u_z in the computational domain, and the immersed boundary method applied to CFD is used to account for the presence of the rough surfaces. Therefore, the local forces $\mathbf{F}$ and heat fluxes Q distributed across the rough surfaces are not readily available. The post-processing routine is mainly based on a Gauss integration scheme with face elements located over the rough walls. The integral equations are the following,

$$\mathbf{F} = \int_A \left(-P\ \mathbf{n} + \frac{1}{Re_\tau} \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \cdot \mathbf{n} \right)\ dA, \tag{11}$$

$$Q = \frac{1}{Pe_\tau} \int_A \nabla T \cdot \mathbf{n}\ dA. \tag{12}$$

Here, A refers to the area of the rough surfaces in contact with the fluid, and $\mathbf{n}$ denotes the normal of the rough surfaces. The normal for the bottom side of the channel can be determined using,

$$\mathbf{n} = \left(\sqrt{1 + \frac{\partial H}{\partial x}^2 + \frac{\partial H}{\partial z}^2} \right)^{-1} \left[-\frac{\partial H}{\partial x}\ ,\ 1\ ,\ -\frac{\partial H}{\partial z} \right]^T. \tag{13}$$

For the top side of the channel, this vector needs to be mirrored. The differential area dA for the face elements located over the rough surfaces is given by,

$$dA = \sqrt{1 + \frac{\partial H}{\partial x}^2 + \frac{\partial H}{\partial z}^2}\ dx\ dz. \tag{14}$$

When combining Eqs. (11)–(14), the expression $\sqrt{1+\partial H/\partial x^2+\partial H/\partial z^2}$ cancels. Therefore, a Gauss–Legendre scheme with a sufficient number of integration points can be used to find the solution of the resulting polynomial integrals for the drag forces and heat fluxes.

In order to perform integration, the variables $(\mathbf{u}, P, T, H)$ were reconstructed using shape functions based on a symmetric stencil of data points surrounding each face element. The degree of the polynomial terms chosen for every variable in each direction is listed in Table 2, whereas a graphical representation of original shape functions is given in Fig. 6. Here, it can be noted that all interpolation schemes considered

Table 2
Order of the polynomial terms considered for the shape functions in every direction while interpolating the local forces and heat fluxes.

Variable	Streamwise (x)	Vertical (y)	Spanwise (z)
P	2	1	2
T	2	1	2
u_x	1	1	2
u_y	2	1	2
u_z	2	1	1
H	2	–	2

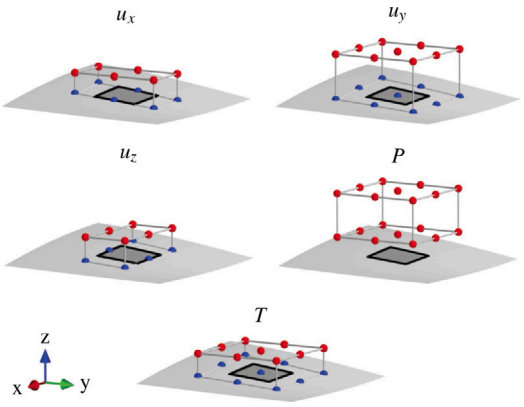


Fig. 6. Schematic representation of the shape functions described in Table 2 for every flow variable considered. The red nodes correspond to data points extracted from the staggered grid of the DNS simulations, whereas the blue nodes are fixed points (with a value of zero) considered over the rough surfaces. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.).

linear terms in the wall-normal direction (y), whereas mixed-order linear or quadratic terms were considered in the horizontal directions ($x - z$). The latter was necessary because the DNS solver employed a staggered discretization scheme to handle the velocity components $\mathbf{u} = (u_x, u_y, u_z)$. Therefore, the number of data points required to form a symmetric stencil was different in each direction. The usage of linear interpolation schemes in the wall-normal direction was validated empirically, since it was noted that quadratic terms yielded slightly higher errors with respect to both the global force and heat transfer balances. All integration areas considered for Eqs. (11)–(12) were centered around the original $x - z$ coordinates for the pressure and temperature fields (P, T). The pressure field was extrapolated in the wall-normal direction (y) by considering the first two data points available above the rough surfaces. However, the velocity and temperature fields $(\mathbf{u}, T)$ considered that the first layer of data points was located exactly over the rough surfaces at the corresponding $H(x, z)$ locations, with a value equal to the homogeneous Dirichlet boundary conditions ($\mathbf{u} = T = 0$).

The numerical implementation of the current post-processing routine was written in PyTorch, since the entire procedure can be expressed as a sequence of parallelizable array operations. The process

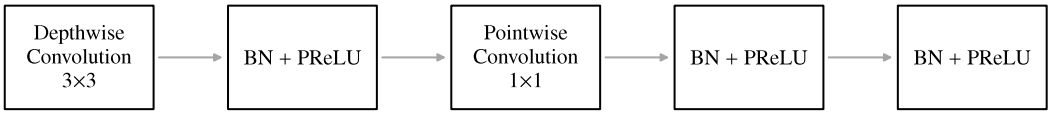


Fig. 7. Schematic representation of the depthwise separable convolution (DSC) modules employed in the machine learning study. The abbreviation BN refers to the 2-D batch normalization operations applied before each PReLU activation function.

of gathering data points located in neighboring $x-z$ locations can be expressed using standard array shift operations due to the presence of periodic boundary conditions for the rough surfaces.

Finally, it is important to note that the local skin friction factors $C_f(x,z)$, Nusselt numbers $Nu(x,z)$ and Stanton numbers $St(x,z)$ for each face element can be computed using the formulas described back in Eqs. (4)–(5). Within the context of these formulas, the local forces $f_x(x,z)$ and heat fluxes $q(x,z)$ per unit of area are given by,

$$f_x(x_i,z_j) = \frac{\mathbf{F}_{i,j}\cdot e_x}{\Delta x_i\ \Delta z_j}, \tag{15}$$

$$q(x_i,z_j) = \frac{Q_{i,j}}{\Delta x_i\ \Delta z_j}. \tag{16}$$

In Eqs. (15)–(16), the sub-indexes (i,j) refer to the 2-D array position of each finite element integration area considered. The variables $\mathbf{F}_{i,j}$ and $Q_{i,j}$ correspond to the results of the global integrals for the drag forces and heat transfer rates defined in Eqs. (11)–(12) for each face element. The term e_x corresponds to the unitary vector in the x-direction, whereas the variables Δx_i and Δz_j refer to the size of each integration area in the $x-z$ directions. Regarding the denominator term $(\Delta x_i\ \Delta z_j)$ found in Eqs. (15)–(16), this expression corresponds to the projected area of each finite element with respect to the $x-z$ plane, or a smooth wall configuration. This definition is consistent with the definitions given in Eqs. (4)–(5) for the skin friction values and Nusselt numbers, since in most engineering applications, the wetted area of a rough surface corresponds to an unknown quantity. Therefore, traditional definitions for the skin friction factor (C_f) or the Nusselt number (Nu) are given with respect to an equivalent smooth wall configuration.

3. Machine learning

In general, the relationship between the input height map $H(x,z)$ of a rough surface and its local skin friction values $C_f(x,z)$ or Nusselt numbers $Nu(x,z)$ corresponds to a complex non-linear function. This Section presents a detailed description of the machine learning systems employed during the current study. First, in Section 3.1, a detailed overview of the neural networks created is given, together with a description of the training procedure and its cost function. Additionally, a novel smoothing procedure is introduced in Section 3.2.

3.1. Neural networks and training procedure

During the current study, convolutional neural networks are employed since these models have been specifically designed to process images and to predict any target quantity of interest. The input height maps for rough surfaces $H(x,z)$ are processed as 2-D arrays formed by pixels, which correspond to grid points on the rough surface. These pixels are then translated into the corresponding local skin friction values $C_f(x,z)$ or Nusselt numbers $Nu(x,z)$ located at the center of each input image.

The deep learning systems employed during the study are formed by layers of depthwise separable convolution (DSC) modules (Chollet, 2017). The main benefits of this approach are that the total number of trainable parameters contained by convolutional neural networks is substantially reduced. Furthermore, an additional layer of non-linear activation functions is added between the depthwise and pointwise convolutional operators to increase the expressivity of the neural networks. For a convolutional operator with C internal channels and kernel size $K\times K$, the total number of parameters employed by DSC modules is reduced from C^2K^2 for a traditional kernel into only C^2+CK^2 parameters. Since $C \gg K$ inside the majority of deep learning systems, the previous analysis implies that the total number of trainable parameters is reduced by a factor of almost K^2.

The configuration of each DSC module employed during the study can be found in Fig. 7. Here, it can be noted that traditional RELU activation functions have been replaced by PReLU operators (He et al., 2015). PReLU operators are able to rescale negative input values by a trainable parameter α, whereas RELU operators simply cancel negative values ($\alpha = 0$). As a result, PReLU operators can increase the expressivity of neural networks with a minimal computational cost, and they can facilitate the convergence of the training procedure. Batch normalization is applied before each PReLU activation function to rescale the internal layers of the deep learning system and to train independent bias factors associated to every non-linear classifier. Furthermore, the usage of max-pooling operators was discarded during the current study. While max-pooling can extract physically relevant features generated by the internal layers of neural networks, we found during a preliminary study that neural networks with max-pooling or average-pooling operators did not yield higher accuracy. Due to this reason, we believe that max-pooling should only be considered as an alternative among other options while designing a neural network. In our final deep learning system, all reduction layers replaced max-pooling operators by additional convolutional modules.

The global configuration of the deep learning architectures used to predict the local skin friction values $C_f(x,z)$ and Nusselt numbers $Nu(x,z)$ are listed in Tables 3 and 4 respectively. In these Tables, the columns C_{in}, C_{out}, K, D, AF refer to the number of input channels, output channels, kernel size, dilation and the presence of activation functions respectively. The increasing dilation levels found in Tables 3 and 4 correspond to a modification introduced into the neural networks to improve their time and space complexity by orders of magnitude, while still producing the same output as a classical convolutional neural network. A detailed explanation of this modification can be found in Appendix A. Fig. 8 presents the total image size employed by the ML system described in Table 3 to predict $C_f(x,z)$ at a particular (x,z) location. As it can be observed, this ML system considers $0.48\ L_x \times 0.49\ L_z$ of the total domain size. Similarly, the ML system to predict $Nu(x,z)$ covers $0.91\ L_x \times 0.92\ L_z$ of the total domain. From a physical perspective, using the entire image size $(L_x \times L_z)$ as input would be ideal. However, increasing the image size can lead to neural networks using an increased number of parameters, which in turn could result in over-fitting. Therefore, a trade-off must be considered between the input image size of a convolutional neural network and its number of trainable parameters. The architectures presented in Tables 3 and 4 were discovered by a simple neural architecture search algorithm based on greedy optimization (Lupo Pasini et al., 2021) for the leading ML configuration. Our algorithm aimed at optimizing the total number of layers, the number of internal convolutional channels, and the configuration of the dilation levels inside the neural networks for both $C_f(x,z)$ and $Nu(x,z)$. The smaller image sizes used to predict $C_f(x,z)$ are potentially caused by the fact that modeling this quantity is a more challenging task than predicting heat transfer, due to the non-locality of pressure drag effects. Therefore, the optimizer may have allocated a large number of trainable parameters to process small input images

Table 3
Deep learning architecture to predict the local skin friction values $C_f(x,z)$. The parameters C_{in}, C_{out}, K and D refer to the number of input channels, output channels, kernel size, dilation and the presence of an activation function respectively.

Layer	C_{in}	C_{out}	K	D	AF
DSC	1	20	3	1×1	Yes
DSC	20	20	3	2×1	Yes
DSC	20	20	3	4×2	Yes
DSC	20	20	3	4×2	Yes
DSC	20	20	3	8×4	Yes
DSC	20	20	3	16×8	Yes
DSC	20	20	3	32×16	Yes
Conv. 1D	20	1	1	–	–

Table 4
Deep learning architecture to predict the local Nusselt numbers $Nu(x,z)$.

Layer	C_{in}	C_{out}	K	D	AF
DSC	1	20	3	1×1	Yes
DSC	20	20	3	2×1	Yes
DSC	20	20	3	4×2	Yes
DSC	20	20	3	8×4	Yes
DSC	20	20	3	16×8	Yes
DSC	20	20	3	32×16	Yes
DSC	20	20	3	64×32	Yes
Conv. 1D	20	1	1	–	–

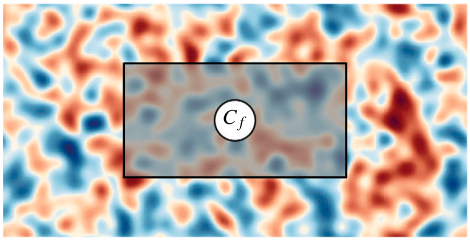


Fig. 8. Physical image size (shaded region) employed by the deep learning architecture described in Table 3 to predict $C_f(x,z)$ at a particular (x,z) location (black circle). The shaded region must always be centered with respect to the predicted point, and thus it must be translated to obtain predictions for other (x,z) locations.

for $C_f(x,z)$, and it avoided using larger images to prevent over-fitting. For future reference, both of the deep learning architectures described in Tables 3 and 4 have a total of 5621 trainable parameters. While the greedy optimizer contained multiple options to increase the number of trainable parameters in each neural network separately, the final architectures chosen by the optimizer contained the same number of layers and trainable parameters.

It is important to highlight that our convolutional neural network always considers input images which are centered with respect to the prediction location. Therefore, the relative position of every roughness element is clearly defined in the streamwise and spanwise directions. Moreover, each pixel in the input images always represents the same physical size (in length units) within the DNS domain, and the row and column indexes of every 2-D image are aligned with the streamwise and spanwise directions respectively.

The training procedure for the deep learning systems was based on a L1 cost function instead of a traditional L2 penalization scheme,

$$\mathcal{J} = \sum_{k,i,j} \left| Y_{i,j}^{[k]\ dns} - Y_{i,j}^{[k]\ ml} \right|. \tag{17}$$

In Eq. (17), the variables Y^{dns} and Y^{ml} correspond to the reference DNS data and the deep learning predictions respectively. Depending on the target of the study, the labels $Y_{i,j}$ can correspond to the local skin friction factors $C_f(x_i,z_j)$ or the local Nusselt numbers $Nu(x_i,z_j)$. The sub-index k refers to the identifier of the DNS case belonging to the training dataset that is being processed. The choice of a L1 cost function given by Eq. (17) allows optimizers to train deep learning models which more accurately focus on minimizing the average errors for the local skin friction factors $C_f(x,z)$ or Nusselt numbers $Nu(x,z)$ than L2 loss functions, which tend to over-penalize small outlier regions. Additionally, during the training procedure, physics-informed data augmentation is performed to account for the fact that mirrored rough surfaces should have identical local skin friction values $C_f(x,z)$ and Nusselt numbers $Nu(x,z)$. Moreover, since all rough surfaces are subject to periodic boundary conditions, the deep learning systems presented in Tables 3 and 4 utilize circular padding (Paszke et al., 2017) to account for the periodicity of the rough surfaces.

3.2. Smoothing procedure to combine the predictions of multiple neural networks

In order to avoid spurious oscillations in the final ML predictions, the results of the study were obtained by combining the results of two independently trained neural networks as shown in Fig. 9. All neural networks were trained by splitting the DNS cases from Category I of the database into 32 training cases and 8 test cases. For each DNS case, only neural networks which included such case in their test set were considered. To explain the smoothing procedure, we consider two different sets of predictions, $Y^{[1]}$ and $Y^{[2]}$, generated by independent neural networks as shown in Fig. 9. We imagine now that $Y^{[1]}$ and $Y^{[2]}$ can be combined such that a smoothed field β is obtained. In other words, β can be written as a weighted combination of multiple $Y^{[k]}$ fields,

$$\beta_{i,j} = \sum_k \alpha_{i,j}^{[k]}\, Y_{i,j}^{[k]}. \tag{18}$$

In Eq. (18), the variable $Y_{i,j}^{[k]}$ represents the predictions of each individual neural network with sub-index $[k]$. The multipliers $\alpha_{i,j}^{[k]}$ represent the weight factors. These factors are subject to the following constraints,

$$0 \le \alpha_{i,j}^{[k]} \le 1, \tag{19}$$

$$\sum_k \alpha_{i,j}^{[k]} = 1. \tag{20}$$

Eqs. (19)–(20) imply that the value of each point in the combined field $\beta_{i,j}$ must be obtained though interpolation between the existing sets of predictions $Y_{i,j}^{[k]}$. To comply with these conditions, the values of the $\alpha_{i,j}^{[k]}$ coefficients were expressed using the softmax function,

$$\alpha_{i,j}^{[k]} = \frac{\exp s_{i,j}^{[k]}}{\sum_k \exp s_{i,j}^{[k]}}. \tag{21}$$

In Eq. (21), the variable $s_{i,j}^{[k]}$ corresponds to an internal set of parameters employed by the softmax function. In order to create a smooth field $\beta_{i,j}$, the following cost function is used,

$$J_\beta = \sum_{i,j} \left(\frac{\partial^2 \beta_{i,j}}{\partial x^2} \right)^2 + \left(\frac{\partial^2 \beta_{i,j}}{\partial z^2} \right)^2, \tag{22}$$

$$\beta_{i,j} = \text{argmin}(J_\beta).$$

In Eq. (22), the second-order derivatives of the target $\beta_{i,j}$ distribution are minimized by the cost function J_β. This formulation has favorable numerical properties, such as being able to avoid isolated spikes in the target $\beta_{i,j}$ distributions, while still preserving physically meaningful peaks or gradients in the solutions generated by machine learning. All the ground-truth values for $C_f(x,z)$ and $Nu(x,z)$ in our DNS database contain smooth gradients, which implies that the previous formulation is well-suited to capture the trends observed in the physical data. Finally, the second-order expressions $\partial^2\beta_{i,j}/\partial x^2$ and $\partial^2\beta_{i,j}/\partial z^2$ in Eq. (22) were approximated using central finite difference expressions,

$$\frac{\partial^2 \beta_{i,j}}{\partial x^2} \approx \frac{\beta_{i+1,j} - 2\,\beta_{i,j} + \beta_{i-1,j}}{\Delta x_i^2}, \tag{23}$$

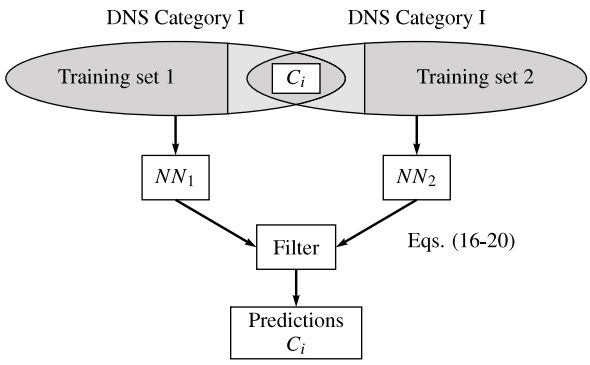


Fig. 9. Schematic representation of the filtering procedure considering two neural networks. The variable C_i refers to the DNS case being predicted, which must belong to the test set of both neural networks. Only DNS cases from Category I of the database are included in the training sets. However, the case C_i may belong to Category I or II.

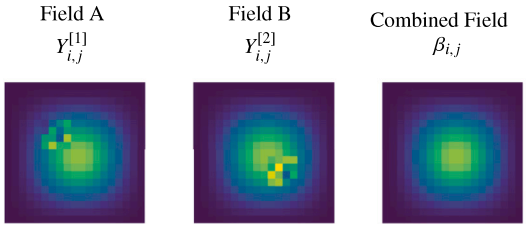


Fig. 10. Synthetic example presenting the results of the filtering methodology described in Section 3.2.

$$\frac{\partial^2 \beta_{i,j}}{\partial z^2} \approx \frac{\beta_{i,j+1} - 2\,\beta_{i,j} + \beta_{i,j-1}}{\Delta z_j^2}. \tag{24}$$

The optimizer chosen to find $\beta_{i,j}$ corresponds to a gradient-descent scheme with adaptive learning rates (Sanhueza et al., 2022; Battiti, 1989). This algorithm has several advantages, such as converging at optimal speeds using learning rates dynamically adjusted at run-time, or preventing divergence by performing line search until a smaller learning rate is found. From a practical perspective, another important property of the previous algorithm is that an early stopping criterion can be applied after a fixed number of iterations N_{max}, since the changes observed in $\beta_{i,j}$ decrease exponentially over time. An example of the results generated by the filtering methodology can be found in Fig. 10. Here, it can be seen that spurious predictions are filtered by the methodology and that a smooth combined field is obtained as a result.

Regarding the number of candidate predictions $Y^{[k]}$ passed to the algorithm, it is important to note that the current methodology is intended to work with only two prediction candidates $Y^{[1]}$ and $Y^{[2]}$. While more $Y^{[k]}$ fields could be considered in theory, the optimization algorithm will always try to choose the smoothest part of every input distribution. Therefore, using only two predictions candidates allows the algorithm to avoid numerical spikes in isolated regions, while still preserving physically meaningful gradients for $C_f(x,z)$ or $Nu(x,z)$.

4. Results

4.1. Local predictions

The local skin friction factors $C_f(x,z)$ and Nusselt numbers $Nu(x,z)$ were computed by combining the machine learning systems and filtering methodology described in Section 3. The results for the local skin friction factors $C_f(x,z)$ can be found in Fig. 11. Here, a comparison is presented between the machine learning predictions with the highest and the lowest accuracy with respect to the DNS data for rough surfaces belonging to Categories I and II of the DNS database. At it can be seen in Fig. 11, the deep learning predictions show physically realistic trends, with the majority of the errors located in areas where extreme values of $C_f(x,z)$ are found. The latter is especially evident for the DNS cases with the highest prediction errors. Moreover, the machine learning predictions for rough surfaces belonging to Categories I and II of the DNS database show similar patterns, which implies that the deep learning system is able to handle irregular surfaces with larger local height gradients in the streamwise direction. However, it can also be noted that the deep learning system tends to under-predict negative $C_f(x,z)$ values in recirculation zones, such as the region indicated by the pairs of markers (A, A') or (B, B') in Fig. 11. One possible reason for this behavior is that large negative $C_f(x,z)$ values rarely occur in our database, and thus the neural network can be biased towards predicting positive quantities.

The results of the deep learning predictions for the local Nusselt numbers $Nu(x,z)$ are presented in Fig. 12. As it can be observed, there is a high degree of similarity between all machine learning predictions and the reference DNS data. Even for the worst case scenarios, the differences between the predicted $Nu(x,z)$ values and the DNS data are much smaller than the differences calculated for $C_f(x,z)$. However, it should be noted that the deep learning predictions for $Nu(x,z)$ tend to be more diffusive near regions with extreme values; see the regions indicated by the pairs of markers (A, A') or (B, B') in Fig. 12, for instance.

The results of the sensitivity analysis for the deep learning predictions with respect to the dataset size can be found in Fig. 13. Here, histograms are presented comparing the distribution of local L1 errors for deep learning systems trained using 32, 16 and 8 rough surfaces. As it can be observed in the histograms, increasing the training set size from 8 to 16 or 32 rough surfaces has a positive effect on improving the accuracy of the momentum and heat transfer predictions. Moreover, it can be noted that predictions for the local Nusselt numbers $Nu(x,z)$ tend to have lower L1 errors than predictions for the local skin friction factors $C_f(x,z)$. One potential reason for the higher accuracy of the local Nusselt number $Nu(x,z)$ predictions is that heat transfer tends to occur mainly in peaks of the rough surfaces with high exposure to the incoming bulk flow, which can be easily identified by the deep learning system. In contrast, the local skin friction factors $C_f(x,z)$ are directly influenced by both pressure and viscous drag (Chung et al., 2021), which correspond to different physical processes. While viscous drag tends to occur close to the windward sides of local peaks, pressure drag can also occur in near stagnant regions due to the non-local pressure changes upstream or downstream with respect to large roughness elements. Therefore, predicting heat transfer may correspond to a simpler regression task than predicting momentum quantities.

It should also be noted that the magnitude of $C_f(x,z)$ can reach values up to 48.1 times higher than the average. This marks a large contrast with respect to the values for the local Nusselt numbers $Nu(x,z)$, which only reach magnitudes up to 14.0 times higher than the average in the datasets analyzed. Therefore, small relative differences between the deep learning predictions and the DNS data in regions where strong drag forces are expected will create much greater L1 errors than similar percentual differences in regions where large heat fluxes occur. The average error of the local ML predictions for $\overline{C_f}$ and $\overline{Nu}$ are 46.55% and 10.25% respectively. The worst 10% percentile of all local predictions have errors higher than 109.3% and 23.49% for $\overline{C_f}$ and $\overline{Nu}$ respectively. Note that these values are scaled with respect to the global values for $\overline{C_f}$ or $\overline{Nu}$ extracted from the DNS simulations. Due to this reason, the previous machine learning errors only correspond to a reference, and it is possible to reach values higher than 100% (of the average) in regions with large peaks for $C_f(x,z)$ or $Nu(x,z)$. For example, the peaks in the local skin frictions factors and Nusselt numbers can easily reach magnitudes of $C_f(x,z)/\overline{C_f} = 48.1$ and $Nu(x,z)/\overline{Nu} = 14.0$ within the DNS database. Under these conditions, a machine learning prediction with a magnitude of $C_f(x,z)/\overline{C_f} \approx 43.29$

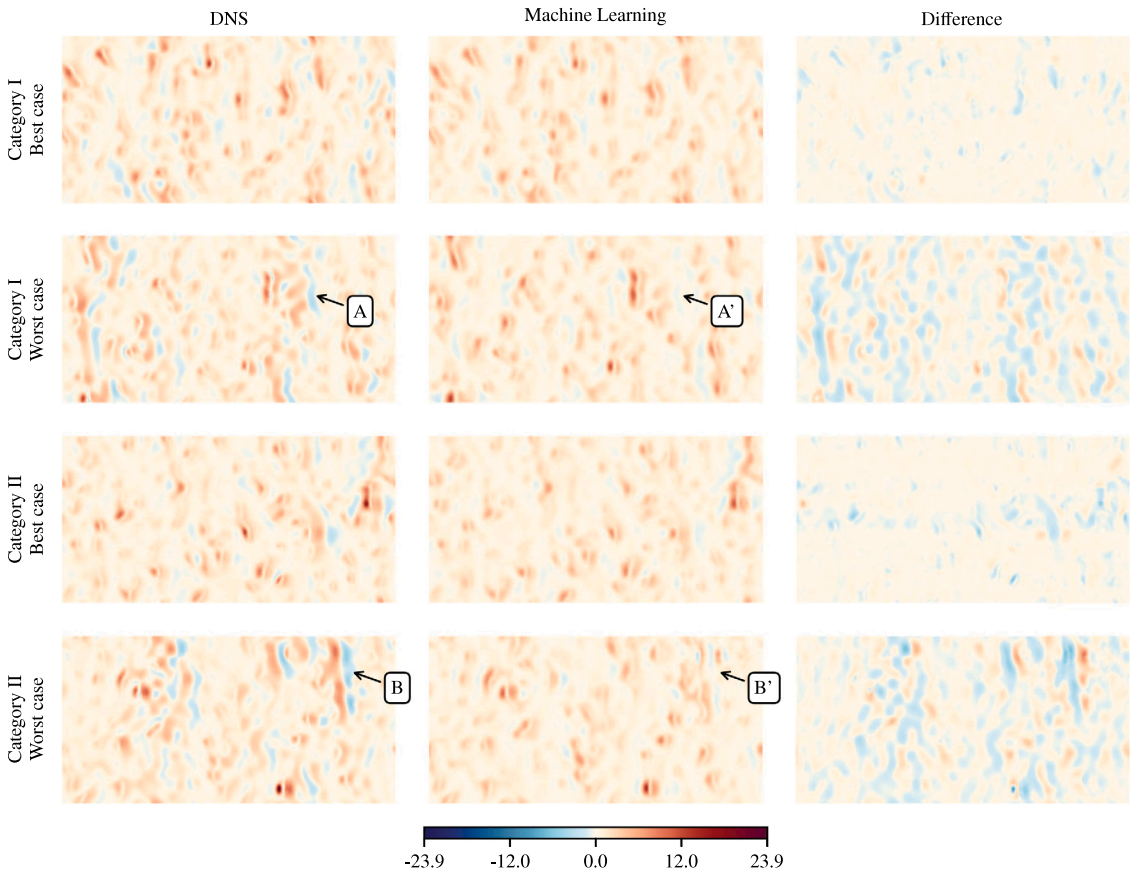


Fig. 11. Comparison between the local skin friction factors $C_f(x,z)$ predicted by deep learning and the DNS data. The cases presented correspond to the highest and the lowest values for the L1-norm of the local errors. The values in the colormap indicate the ratio between $C_f(x,z)$ at every spatial location and the average skin friction factor $\overline{C_f}$ calculated according to the DNS data.

in the region with the highest recorded values for the local skin friction factors would only have a 10% local error, yet a 481% error with respect to the average value of $\overline{C_f}$. While the previous analysis may suggest that using local relative errors would be a better alternative, the opposite case can also occur. For instance, in a region with low $C_f(x,z)/\overline{C_f} \approx 0.001$ values, a ML prediction of $C_f(x,z)/\overline{C_f} \approx 0.01$ would be off by a factor of 10 locally, but only have a 0.9% error with respect to the global $\overline{C_f}$ value. Due to this reason, it was decided to always report the ML errors performing a comparison with respect to the average $\overline{C_f}$ values.

After a detailed analysis, it was found that the local slope angles of the rough surface and the errors in the machine learning predictions were only weakly correlated. However, the highest errors in the machine learning predictions are correlated with the presence of large clusters of roughness elements blocking the path of the incoming flow. The machine learning errors can occur either upstream or downstream with respect to these elements, due to pressure drag effects. This observation is not directly quantified by traditional metrics, but it can be clearly observed in examples, such as Fig. 14. The cluster of roughness elements shown in this image creates multiple recirculation zones, which are not well-predicted by machine learning. Additionally, discrepancies can be observed at the top of the rough surfaces, and upstream of the cluster of roughness elements.

4.2. Global predictions

A study was conducted to analyze the errors observed for the global skin friction factors ($\overline{C_f}$) and Nusselt numbers ($\overline{Nu}$) for every rough surface. The results were compared with traditional correlations found in the literature, which are based on the dimensionless Nikuradse sand-grain roughness height $k^+_{s,eq}$. After performing a preliminary analysis, it was determined that the correlation developed by Flack and Schultz (2010) yielded the most accurate predictions,

$$k_{s,eq} = 4.43\ S_q\ (1 + S_{sk})^{1.37}. \tag{25}$$

In Eq. (25), the variables S_q and S_{sk} are the root-mean-squared height variations of the rough surface and its skewness, respectively (Thakkar et al., 2017),

$$S_q = \sqrt{\frac{1}{n_x n_z}\sum_{i,j}^{n_x,n_z} H^2_{i,j}}, \tag{26}$$

$$S_{sk} = S_q^{-3}\frac{1}{n_x n_z}\sum_{i,j}^{n_x,n_z} H^3_{i,j}. \tag{27}$$

In Eqs. (26)–(27), the variable $H_{i,j}$ corresponds to a 2-D array with the height of the rough surfaces at discrete $[i,j]$ locations in the $x-z$ directions respectively. After obtaining the Nikuradse sand-grain roughness height $k_{s,eq}$, the value of $k^+_{s,eq}$ can be computed from:

$$k^+_{s,eq} = k_{s,eq}\frac{u_\tau}{\nu}. \tag{28}$$

In order to obtain the skin friction factor $\overline{C_f}$ associated with $k^+_{s,eq}$ in the Nikuradse diagram (Nikuradse, 1933), the following formulas can be used (Peeters and Sandham, 2019; Thakkar et al., 2017):

$$\overline{C_f} = \frac{2}{\left(U_b^+\right)^2}, \tag{29}$$

$$U_b^+ = U^+_{b,s} - \Delta U^+, \tag{30}$$

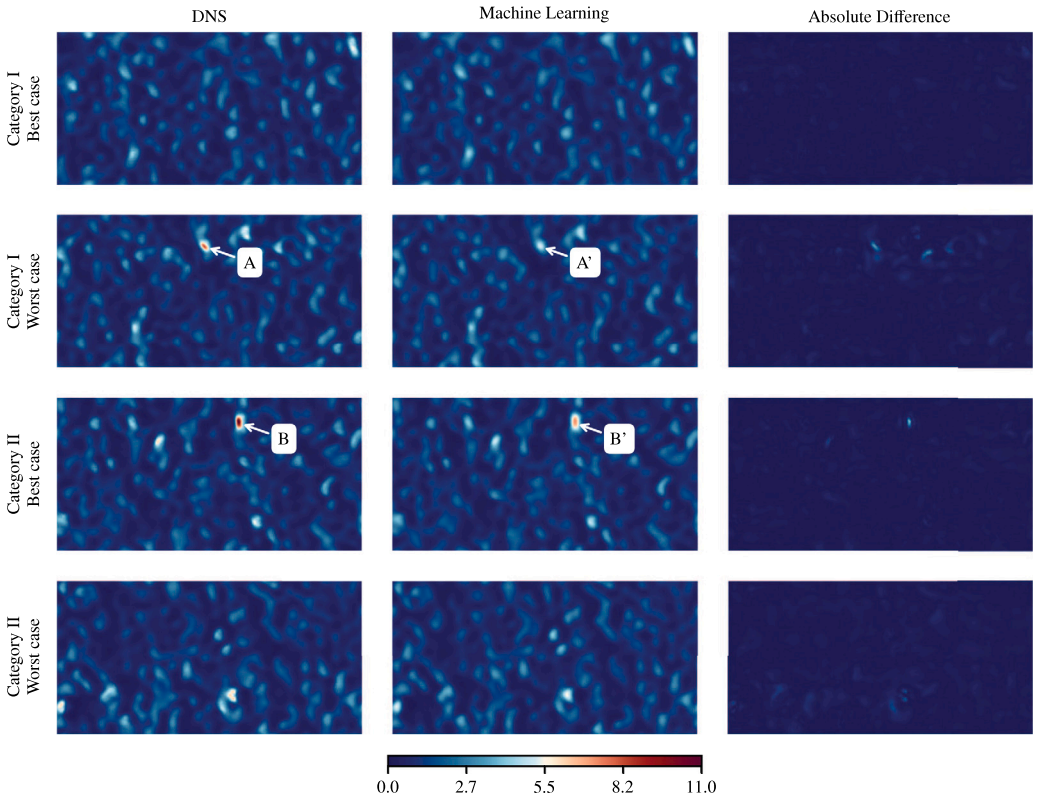


Fig. 12. Comparison between the local Nusselt numbers $Nu(x,z)$ predicted using machine learning and the reference DNS data. The colormap indicates the ratio between the local values of the Nusselt numbers $Nu(x,z)$ and the average Nusselt numbers $\overline{Nu}$ given by the DNS data.

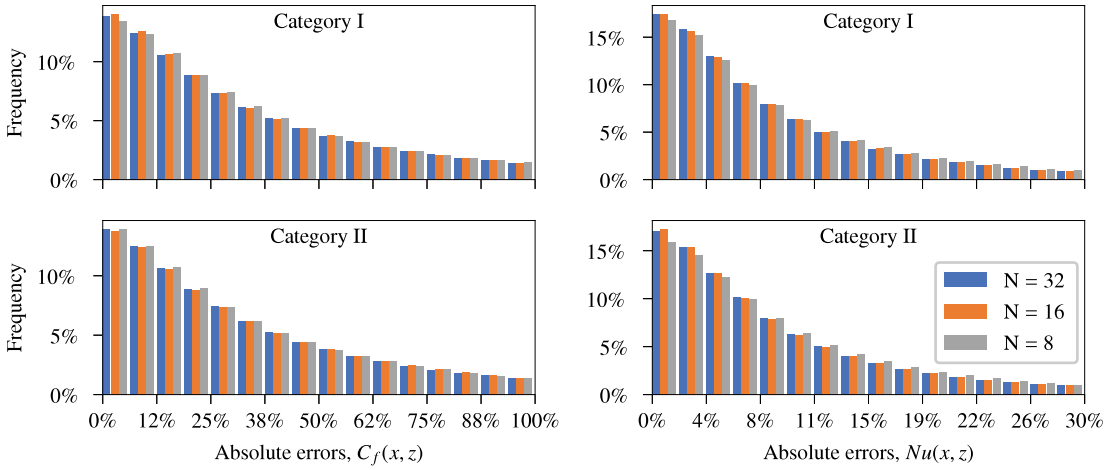


Fig. 13. Changes in the distribution of the absolute errors for the local skin friction factors $C_f(x,z)$ and the Nusselt numbers $Nu(x,z)$ with respect to the training dataset size. All errors are scaled using the average DNS values for $\overline{C_f}$ or $\overline{Nu}$ within each case. The legend indicates the number of DNS cases used to train the deep learning models, which were extracted from Category I of the database. The title of each subplot indicates to which category the test cases plotted in each histogram belong. For the DNS cases from Category I of the database, multiple deep learning models were trained, such that each sample could be considered as a held-back test case in a separate study.

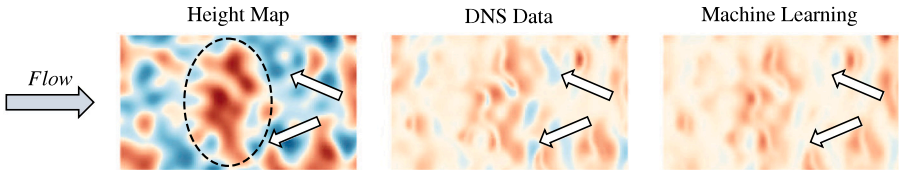


Fig. 14. Example of the correlation between large clusters of roughness elements blocking the incoming flow and large errors in the machine learning predictions distributed nearby.

$$\Delta U^+ = U^+_{c,s} - U^+_c. \tag{31}$$

In Eqs. (29)–(31), the variable U^+_b corresponds to the dimensionless bulk velocity. The parameters $U^+_{c,s}$ and $U^+_{b,s}$ correspond to the dimensionless time-averaged velocity of a turbulent channel flow with smooth walls at the channel center ($y = \delta$) and its bulk velocity respectively. Based on existing DNS data (Peeters and Sandham, 2019), these variables were assigned values of $U^+_{c,s} = 18.64$ and $U^+_{b,s} = 15.79$ for a turbulent channel flow at $Re_\tau = 180$. The dimensionless velocity at the channel center (U^+_c) can be calculated using Eqs. (9)–(10) from Section 2.4 based on the value of $k^+_{s,eq}$ in Eq. (28). The Nusselt number for the rough surfaces is then calculated from the correlation for the Stanton number given by Dipprey and Sabersky (1963),

$$\overline{St} = \frac{C_f/2}{1 + \sqrt{C_f/2}\left\{k_f \left[Re_b\sqrt{C_f/2}\,(\epsilon_s/D)\right]^{0.2} Pr^{0.44} - 8.48\right\}}. \tag{32}$$

In Eq. (32), the parameters k_f has a value of 5.19. The ratio ϵ_s/D found in Eq. (32) is given by,

$$\frac{\epsilon_s}{D} = \exp\left(\frac{3 - 1/\sqrt{C_f/2}}{2.5}\right). \tag{33}$$

The Nusselt number $\overline{Nu}$ associated to the correlation of Dipprey and Sabersky (1963) is then calculated from the Stanton number using the formula $\overline{Nu} = \overline{St}\ Re_b\ Pr$ given by Eq. (6). The bulk Reynolds number found in Eq. (32) can be calculated from the dimensionless bulk velocity U^+_b given by Eq. (29), since $u_\tau = 1$ in all DNS cases.

The results of the study regarding the accuracy of the global skin friction factors $\overline{C_f}$ and Nusselt number predictions $\overline{Nu}$ can be found in Fig. 15. As it can be observed, the deep learning system is substantially more accurate than traditional correlations while predicting both momentum $\overline{C_f}$ and heat transfer $\overline{Nu}$ parameters. The maximum errors for the skin friction coefficients $\overline{C_f}$ in the deep learning system reached values of 8.07%, whereas the system with traditional correlations reached errors up to 24.9%. In the case of heat transfer, the maximum errors for the local Nusselt numbers were 2.87% and 13.5% for the deep learning system and the traditional correlations respectively. Both of these results are satisfactory for machine learning. Moreover, only small differences were found between the accuracy of the deep learning system while making predictions for rough surfaces belonging to both Categories I and II of the DNS database.

Additionally, it is important to analyze the ratio between the errors in the machine learning predictions and the variations observed in the DNS database for $\overline{C_f}$ and $\overline{Nu}$. If the machine learning errors have an absolute value smaller than the variations observed for $\overline{C_f}$ and $\overline{Nu}$ in the DNS data, it means that machine learning is accurately predicting the physical changes observed. Given this context, the results of Table 1 indicate that the maximum variations observed in the DNS data for $\overline{C_f}$ and $\overline{Nu}$ are 18.8% and 6.5% respectively. Therefore, the maximum errors found in the machine learning predictions of 8.1% for $\overline{C_f}$ and 2.9% for $\overline{Nu}$ can be regarded as small. The ratio between the maximum variations observed in the DNS data and the deep learning predictions is 2.3 for both $\overline{C_f}$ (18.8%/8.1%) and $\overline{Nu}$ (6.5%/2.87%). Therefore, it can be concluded that deep learning constitutes a valid alternative to generate improved predictions for flow parameters, such as $\overline{C_f}$ or $\overline{Nu}$, if enough training data is collected.

To further put the neural network predictions into perspective, the maximum and the mean relative errors are compared with other predictive methods in Table 5, namely, the correlations of Flack and Schultz (2010) and Dipprey and Sabersky (1963), the re-calibrated versions of the latter, and simply considering the average values found in the training sets. The last method refers to averaging the $n_s - 1$ samples found in the training set of a predictive model for a target rough surface. The re-calibrated version of the correlation by Flack and Schultz (2010) has the following form: $k_{s,eq} = a\ S_q\ (1 + S_{sk})^b$, where a varies from 4.114 to 4.129 in every trained model, and b varies from 0.272 and 0.322. Additionally, in order to analyze heat transfer, the correlation from Dipprey and Sabersky (1963) was also re-calibrated to fit our training data, the parameterized version of the correlation has the form:

$$\overline{St} = \frac{C_f/2}{1 + \sqrt{C_f/2}\left\{a\left[Re_b\sqrt{C_f/2}\,(\epsilon_s/D)\right]^b Pr^{0.44} - c\right\}}, \tag{34}$$

$$\frac{\epsilon_s}{D} = \exp\left(\frac{d - 1/\sqrt{C_f/2}}{e}\right). \tag{35}$$

In Eqs. (34)–(35), the re-calibrated parameters have the following values after training: $a \in [5.213, 5.222]$, $b \in [0.230, 0.234]$, $c \in [8.453, 8.461]$, $d \in [3.011, 3.016]$ and $e \in [2.562, 2.589]$. The mean relative errors are calculated as follows:

$$L_1\,\overline{C_f} = \frac{1}{n_s}\sum_{k=1}^{n_s}\left|\frac{\overline{C_f}^{[k]\ pred} - \overline{C_f}^{[k]\ dns}}{\overline{C_f}^{[k]\ dns}}\right|, \tag{36}$$

$$L_1\,\overline{Nu} = \frac{1}{n_s}\sum_{k=1}^{n_s}\left|\frac{\overline{Nu}^{[k]\ pred} - \overline{Nu}^{[k]\ dns}}{\overline{Nu}^{[k]\ dns}}\right|. \tag{37}$$

In Eqs. (36)–(37), n_s refers to the number of samples in our dataset. As it can be observed in Table 5, our machine learning model significantly outperforms the traditional correlations from Flack and Schultz (2010) and Dipprey and Sabersky (1963). However, the machine learning model for $\overline{C_f}$ only displays a marginal improvement compared to the predictions of the re-calibrated correlation and the $n_s - 1$ averaged values. This result stems from the fact that the machine learning model was not trained to predict $\overline{C_f}$ and $\overline{Nu}$ directly, but it was rather trained to predict the local $C_f(x, z)$ and $Nu(x, z)$ values. Therefore, the deep learning predictions for $\overline{C_f}$ and $\overline{Nu}$ reported in Table 5 correspond to an emergent quantity, rather than an objective.

Regarding the generalizability of the model, we believe that the current convolutional neural network requires training data with similar flow characteristics, or boundary conditions, as the target test cases to make accurate predictions. In order to utilize different Reynolds numbers Re_τ or Prandtl numbers Pr, further research is required to establish the best methodology to include these quantities as input to the neural network.

5. Conclusions

This study presented a deep learning architecture capable of predicting detailed maps for the local skin friction factors $C_f(x, z)$ and Nusselt numbers $Nu(x, z)$ of irregular surfaces. The results show that machine learning is able to achieve reliable results while predicting both global force and heat transfer parameters. A sensitivity study with respect to the dataset size also revealed a significant reduction in the errors for the momentum and heat transfer predictions once the dataset size is increased from 8 to 32 DNS cases. The comparisons performed with respect to traditional correlations proved that deep learning is a valid alternative to generate improved predictions for important flow parameters, such as the skin friction factors $\overline{C_f}$ or the Nusselt numbers $\overline{Nu}$. Moreover, the machine learning systems trained were able to obtain reliable predictions while working with rough surfaces containing abrupt changes in their roughness elements. The maximum error for $\overline{C_f}$ using traditional correlations was 24.9%, whereas deep learning only reached errors of 8.1%. Similarly, the maximum errors for $\overline{Nu}$ were reduced from 13.5% using traditional correlations to only 2.9% using deep learning. Regarding the local errors for $C_f(x, z)$ and $Nu(x, z)$, the probability density functions revealed that the median of the errors were lower than 28.42% and 6.37% respectively. The reason for the higher errors observed in C_f is likely related to non-local effects caused by pressure drag, as there is no analogous phenomenon for heat transfer. Therefore, it is recommended to perform further research regarding the creation deep learning models to predict the behavior of turbulent flows past rough surfaces.

Table 5
Comparison of the maximum and the mean relative errors in the predictions for the global skin friction factors $\overline{C_f}$ and Nusselt numbers $\overline{Nu}$ using different models for rough surfaces belonging to Categories I and II of the DNS database.

	$L_1\ \overline{C_f}$	$L_\infty\ \overline{C_f}$	$L_1\ \overline{Nu}$	$L_\infty\ \overline{Nu}$
Machine learning	2.69%	8.07%	0.94%	2.87%
Correlation by Flack and Schultz (2010), Dipprey and Sabersky (1963)	7.89%	24.88%	10.05%	13.49%
Re-calibrated correlations from Flack and Schultz (2010), Dipprey and Sabersky (1963)	2.78%	8.92%	1.02%	3.74%
$n_s - 1$ averaged values	2.95%	11.12%	1.03%	3.86%

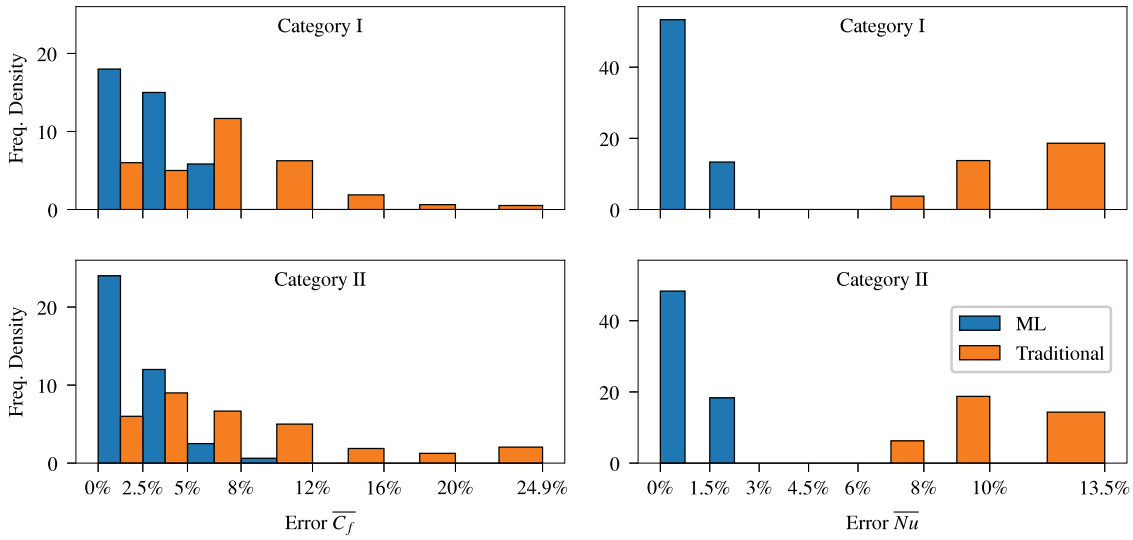


Fig. 15. Histograms comparing the distribution of errors for the skin friction values $\overline{C_f}$ and the Nusselt numbers $\overline{Nu}$ between the deep learning model and the traditional correlations of Flack and Schultz (2010) and Dipprey and Sabersky (1963). The title of each subplot indicates the category of the DNS database to which the test cases found in each histogram belong. For rough surfaces belonging to Category I of the DNS database, the mean relative errors in the machine learning predictions are 2.92% for $\overline{C_f}$ and 0.90% for $\overline{Nu}$. Similarly, for Category II, the mean relative errors are 2.46% and 0.97% for $\overline{C_f}$ and $\overline{Nu}$ respectively.

CRediT authorship contribution statement

Rafael Diez Sanhueza: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing. **Ido Akkerman:** Conceptualization, Methodology, Supervision, Writing – review & editing. **Jurriaan W.R. Peeters:** Conceptualization, Methodology, Supervision, Writing – review & editing, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Time and space complexity optimization

During the current study, one of the main challenges encountered while using traditional convolutional neural networks is that their architecture has been adapted to reduce an input image until a single prediction point is obtained, and to discard the results of all intermediate convolutional layers. This process is highly inefficient, as it can be observed in Fig. A.16. Here, an example is presented regarding the sub-images required to perform predictions for the local skin friction values $C_f(x,z)$ or Nusselt numbers $Nu(x,z)$ associated with two neighboring data points P_1 and P_2 inside a rough surface. The large overlapping area highlighted in Fig. A.16 corresponds to the shared portion of both sub-images which must be scanned redundantly once predictions are required for a second data point P_2. From a mathematical perspective, this implies that in order to obtain predictions for the local skin friction values $C_f(x,z)$ or Nusselt numbers $Nu(x,z)$ associated to a rough surface discretized using images with $N_x \times N_z$ pixels, the total numbers of pixels to be processed by the deep learning architecture is proportional to $\mathcal{O}\left(N_x^2 N_z^2\right)$. In practical terms, a traditional convolutional neural network replicating the behavior of the deep learning systems presented in Tables 3 or 4 would require processing a total of 365,148,000 pixels for each DNS case considered.

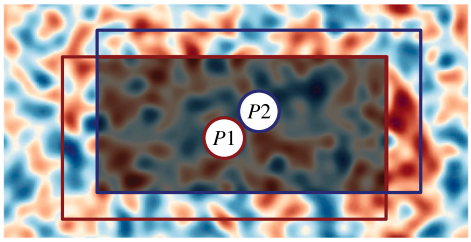


Fig. A.16. Example regarding two prediction points $P1$ and $P2$ obtained as output from a traditional convolutional neural network. The shaded area corresponds to the shared portion of the input images passed to the convolutional neural network, which must be reprocessed again.

However, an efficient alternative can be developed to solve the previous problem, and to reduce the total number of pixels to be processed from quadratic complexity $\mathcal{O}\left(N_x^2 N_z^2\right)$ to linear complexity $\mathcal{O}\left(N_x N_z\right)$ with respect to the total image size $N_x N_z$. The approach is based on the idea of sharing the results of all intermediate convolutional layers by using an advanced dilation system that replaces the reduction strides found in traditional convolutional neural networks by a unified matrix system. A schematic representation of this approach can be found in

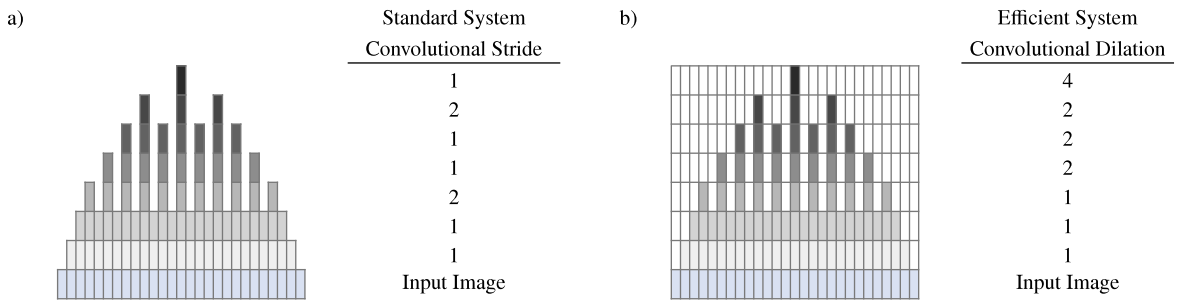


Fig. A.17. Comparison in 1-D between a standard convolutional neural network and the efficient computer vision system developed.

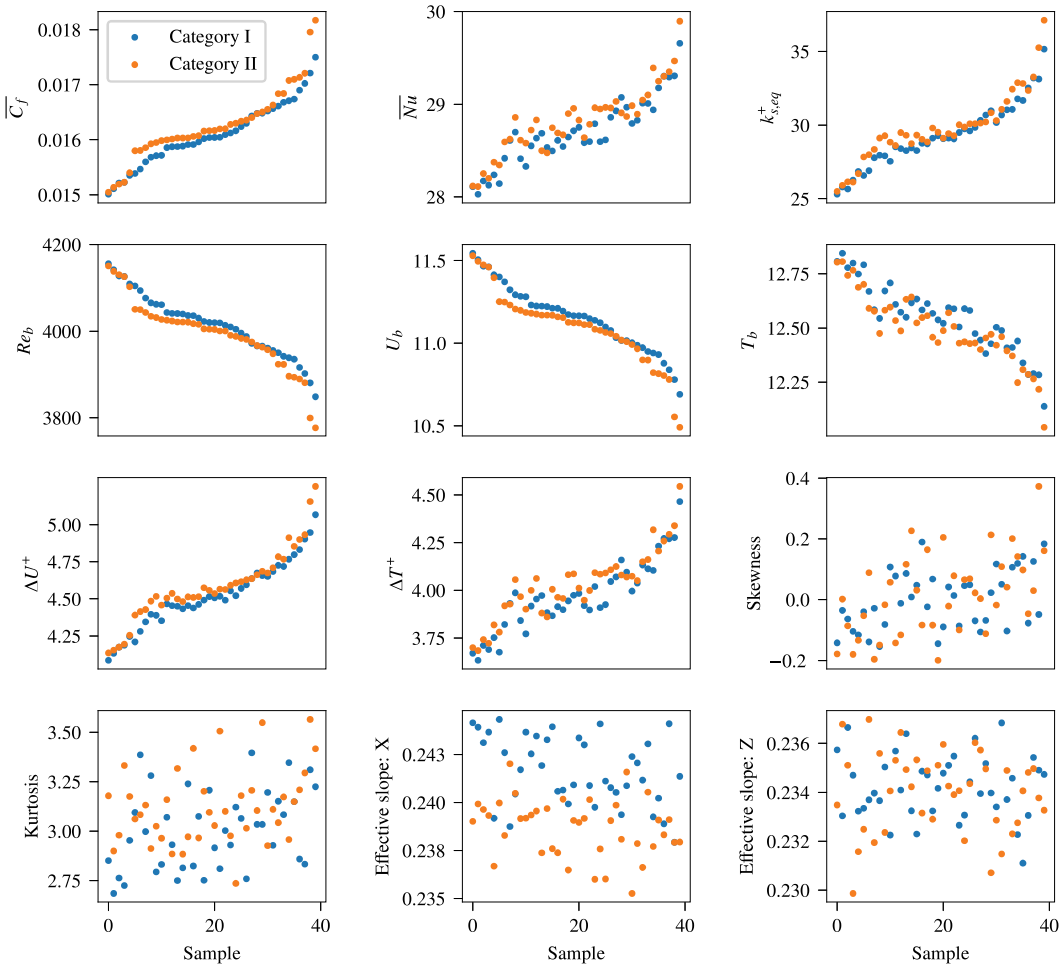


Fig. B.18. Distribution of averaged flow quantities reported in Table 1 for our DNS database, as well as other traditional surface metrics. For improved readability, all values shown for Categories I and II are sorted in ascending order according to their global skin friction factors $\overline{C_f}$. All the rough surfaces shown in the subplots have a root-mean-squared height of 0.0358.

Fig. A.17. Here, it can be noted that, instead of reducing images using convolutional strides, the efficient deep learning system proposed uses increased levels of dilation to replicate the effect of using reduction strides. The main benefit of this approach is that neighboring data points, such as P_1 and P_2 back in Fig. A.16, can share the results for the intermediate convolutional layers, and all the predictions for the local skin friction values $C_f(x, z)$ or Nusselt numbers $Nu(x, z)$ associated to a rough surface can be obtained after only one pass through the neural network. As a result, the total number of pixels to be processed for each DNS case is reduced from 365,148,000 pixels into only 39,200 pixels, which equivalent to a reduction factor of 9315. Moreover, the running times required to train deep learning systems can be reduced from months of GPU-time into only a few hours. The optimized architecture described in Fig. A.17 is highly compatible with GPU's, since most machine learning frameworks contain efficient implementations of the dilation operator for convolutions.

Appendix B. Averaged flow quantities and surface metrics for the DNS database

In this Appendix, subplots are presented regarding the distribution of the flow quantities reported in Table 1, as well as the value of traditional surface metrics for our DNS database. For improved transparency, the results of Fig. B.18 are split for rough surfaces belonging to Categories I and II of the DNS database. As mentioned in Section 2.4,

all the rough surfaces found in our DNS database have a constant root-mean-squared height of 0.0358.

References

Battiti, R., 1989. Accelerated backpropagation learning: Two optimization methods. Complex Syst. 3 (4).

Busse, A., Lützner, M., Sandham, N.D., 2015. Direct numerical simulation of turbulent flow over a rough surface based on a surface scan. Comput. & Fluids 116, 129–147.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition. pp. 1800–1807.

Chung, D., Hutchins, N., Schultz, M.P., Flack, K.A., 2021. Predicting the drag of rough surfaces. Annu. Rev. Fluid Mech. 53 (1), 439–471.

De Marchis, M., Milici, B., Napoli, E., 2019. Large eddy simulations on the effect of the irregular roughness shape on turbulent channel flows. Int. J. Heat Fluid Flow 80, 108494.

Dipprey, D., Sabersky, R., 1963. Heat and momentum transfer in smooth and rough tubes at various prandtl numbers. Int. J. Heat Mass Transfer 6 (5), 329–353.

Fadlun, E., Verzicco, R., Orlandi, P., Mohd-Yusof, J., 2000. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. J. Comput. Phys. 161 (1), 35–60.

Flack, K.A., 2018. Moving beyond moody. J. Fluid Mech. 842, 1–4.

Flack, K.A., Schultz, M.P., 2010. Review of Hydraulic Roughness Scales in the Fully Rough Regime. J. Fluids Eng. 132 (4), 041203.

Forooghi, P., Stripf, M., Frohnapfel, B., 2018. A systematic study of turbulent heat transfer over rough walls. Int. J. Heat Mass Transfer 127, 1157–1168.

Golovin, K.B., Gose, J.W., Perlin, M., Ceccio, S.L., Tuteja, A., 2016. Bioinspired surfaces for turbulent drag reduction. Phil. Trans. R. Soc. A 374 (2073), 20160189.

Gong, W., Shen, J., Dai, W., Li, K., Gong, M., 2021. Research and applications of drag reduction in thermal equipment: A review. Int. J. Heat Mass Transfer 172, 121152.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In: IEEE International Conference on Computer Vision. pp. 1026–1034.

Jouybari, M.A., Yuan, J., Brereton, G.J., Murillo, M.S., 2021. Data-driven prediction of the equivalent sand-grain height in rough-wall turbulent flows. J. Fluid Mech. 912, A8.

Kim, J., Moin, P., Moser, R., 1987. Turbulence statistics in fully developed channel flow at low Reynolds number. J. Fluid Mech. 177, 133–166.

Ling, J., Kurzawski, A., Templeton, J., 2016. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. J. Fluid Mech. 807, 155–166.

Lupo Pasini, M., Yin, J., Li, Y.W., Eisenbach, M., 2021. A scalable algorithm for the optimization of neural network architectures. Parallel Comput. 104–105, 102788.

Moody, L.F., 1944. Friction factors for pipe flow. Trans. Asme 66, 671–684.

Napoli, E., Armenio, V., De Marchis, M., 2008. The effect of the slope of irregularly distributed roughness elements on turbulent wall-bounded flows. J. Fluid Mech. 613, 385–394.

Nikuradse, J., 1933. Laws of flow in rough pipes. VDI Forschungsheft 361.

Nilpueng, K., Wongwises, S., 2015. Experimental study of single-phase heat transfer and pressure drop inside a plate heat exchanger with a rough surface. Exp. Therm Fluid Sci. 68, 268–275.

Orlandi, P., Leonardi, S., 2006. DNS of turbulent channel flows with two- and three-dimensional roughness. J. Turbul. 7, N73.

Parish, E.J., Duraisamy, K., 2016. A paradigm for data-driven predictive modeling using field inversion and machine learning. J. Comput. Phys. 305, 758–774.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A., 2017. Automatic differentiation in pytorch. In: NIPS-W.

Peeters, J., Sandham, N., 2019. Turbulent heat transfer in channels with irregular roughness. Int. J. Heat Mass Transfer 138, 454–467.

Sandberg, R.D., Zhao, Y., 2022. Machine-learning for turbulence and heat-flux model development: A review of challenges associated with distinct physical phenomena and progress to date. Int. J. Heat Fluid Flow 95, 108983.

Sanhueza, R.D., Smit, S., Peeters, J., Pecnik, R., 2022. Machine learning for RANS turbulence modelling of variable property flows. https://www.sciencedirect.com/science/article/pii/S0045793023000609.

Schultz, M.P., Flack, K.A., 2009. Turbulent boundary layers on a systematically varied rough wall. Phys. Fluids 21 (1), 015104.

Thakkar, M., Busse, A., Sandham, N., 2017. Surface correlations of hydrodynamic drag for transitionally rough engineering surfaces. J. Turbul. 18 (2), 138–169.

Thakkar, M., Busse, A., Sandham, N., 2018. Direct numerical simulation of turbulent channel flow over a surrogate for nikuradse-type roughness. J. Fluid Mech. 837.

Ventola, L., Robotti, F., Dialameh, M., Calignano, F., Manfredi, D., Chiavazzo, E., Asinari, P., 2014. Rough surfaces with enhanced heat transfer for electronics cooling by direct metal laser sintering. Int. J. Heat Mass Transfer 75, 58–74.

Weatheritt, J., Sandberg, R., 2016. A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. J. Comput. Phys. 325, 22–37.

Yuan, J., Piomelli, U., 2014. Estimation and prediction of the roughness function on realistic surfaces. J. Turbul. 15 (6), 350–365.