J. Fluid Mech. (2022), *vol.* 933, A18, doi:10.1017/jfm.2021.1041

Predicting drag on rough surfaces by transfer learning of empirical correlations

Sangseung Lee[1],†, Jiasheng Yang[2], Pourya Forooghi[3], Alexander Stroh[2] and Shervin Bagheri[1],†

[1]Department of Engineering Mechanics, FLOW Centre, KTH, Stockholm SE-100 44, Sweden

[2]Institute of Fluid Mechanics, Karlsruhe Institute of Technology, Karlsruhe 76131, Germany

[3]Department of Mechanical and Production Engineering, Aarhus University, 8000 Aarhus C, Denmark

(Received 6 July 2021; revised 30 October 2021; accepted 18 November 2021)

Recent developments in neural networks have shown the potential of estimating drag on irregular rough surfaces. Nevertheless, the difficulty of obtaining a large high-fidelity dataset to train neural networks is deterring their use in practical applications. In this study, we propose a transfer learning framework to model the drag on irregular rough surfaces even with a limited amount of direct numerical simulations. We show that transfer learning of empirical correlations, reported in the literature, can significantly improve the performance of neural networks for drag prediction. This is because empirical correlations include 'approximate knowledge' of the drag dependency in high-fidelity physics. The 'approximate knowledge' allows neural networks to learn the surface statistics known to affect drag more efficiently. The developed framework can be applied to applications where acquiring a large dataset is difficult but empirical correlations have been reported.

Key words: machine learning

1. Introduction

Nearly all surfaces in fluid-related industries are rough at their operating Reynolds numbers. These rough surfaces easily alter the aero- or hydrodynamic properties of a fluid system and induce unfavourable consequences; for instance, reduced stability of aircraft, increased fuel consumption of cargo ships and reduced energy harvesting capacity of wind turbines (Gent, Dart & Cansdale 2000; Dalili, Edrisy & Carriveau 2009; Schultz *et al.* 2011). Therefore, it is crucial to restore or replace surfaces that have degraded past

† Email addresses for correspondence: sanlee@mech.kth.se, shervin@mech.kth.se

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

a critical level. This requires monitoring roughness on industrial surfaces and efficiently predicting their effect on the flow.

One of the most important effects of surface roughness on flow is the increased drag. This additional drag is generally expressed by the Hama roughness function (Hama 1954) or the equivalent sand-grain height. The Hama roughness function is the downward shift of the mean velocity profile induced by roughness. The equivalent sand-grain height is the roughness height from the Moody diagram (Moody 1944) that would cause the same drag as the rough surface of interest. It is worth noting that to calculate the equivalent sand-grain height, the Hama roughness function in the fully-rough regime – where the skin-friction coefficient is independent of Reynolds number – needs to be determined. To accurately determine the Hama roughness function of irregular surfaces, numerical simulations with fully resolved rough structures or experiments are needed. However, neither simulations nor experiments are cost-efficient and hence not feasible for practical purposes. To alleviate the high costs, researchers have found empirical correlations between rough surface statistics and their contribution to drag (Flack & Schultz 2010; Chan *et al.* 2015; Forooghi *et al.* 2017; Flack, Schultz & Barros 2020). However, no universal model has emerged that is able to accurately predict the drag from a sufficiently wide range of roughness. This is mainly attributed to the high-dimensional feature space of irregular rough structures (Chung *et al.* 2021).

Neural networks are known to be capable of extracting patterns in high-dimensional spaces and therefore have been successfully used in various studies using fluid flow data (Lee & You 2019; Kim & Lee 2020; Fukami, Fukagata & Taira 2021). Jouybari *et al.* (2021) developed a multilayer perceptron (MLP) type neural network to find a mapping of 17 different rough surface statistics to equivalent sand-grain height. They reported a state-of-the-art performance in predicting equivalent sand-grain heights from rough surface statistics. A total of 45 data samples, obtained from direct numerical simulations (DNSs) and experiments, were used for training and validating their neural network. Having a large number of data samples to train a neural network is invariably advantageous; nonetheless, constructing a fluid database that is sufficiently large to train a network from scratch imposes significant computational or experimental cost and is often considered impractical. Therefore, for the practical usage of neural networks in fluid-related applications, a framework that allows neural networks to learn a generalized mapping from a limited number of data samples is needed.

In this study, we propose a transfer learning framework to improve the generalization ability of neural networks for predicting the drag on rough surfaces when only a limited number of high-fidelity data samples are available. Transfer learning is a method that improves learning performance by adapting knowledge learned from one domain to another (Pan & Yang 2009; Chakraborty 2021). The method has been used, for example, for self-driving cars, which can be pre-trained with data from virtual car simulators (Martinez *et al.* 2017; Pan *et al.* 2017; Akhauri, Zheng & Lin 2020). Obviously, the driving environment in virtual car simulators is inherently different from the real world, however, pre-training a neural network with 'approximate knowledge' can provide better initial weights for training real self-driving cars.

Similarly, for flow over rough surfaces, empirical correlations between drag and surface roughness provide an approximate knowledge of real flow physics (Colebrook & White 1937; Moody 1944; Flack & Schultz 2010; Chan *et al.* 2015; Forooghi *et al.* 2017; Flack *et al.* 2020). These correlations were developed by a fitting procedure of large experimental and numerical datasets. However, it is not straightforward to make direct use of these datasets, in particular because the information of surface topographies and flow statistics is not always accessible. Indeed, in many cases we have advanced our understanding of

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

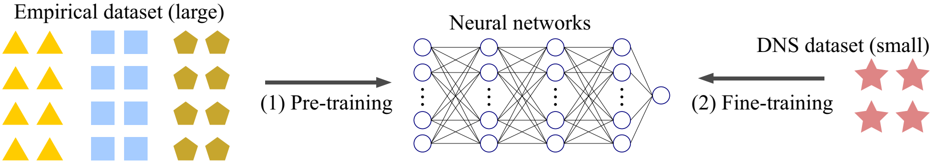

Figure 1. Overview of the transfer learning framework.

physics from the valuable information embedded in empirical correlations, such as those developed by Colebrook & White (1937) and Moody (1944).

The aim of this study is to show that transfer learning of empirical correlations can significantly improve the performance of neural networks for modelling drag on rough surfaces. Our aim is also to analyse a simple neural network with and without transfer learning to gain insight of the learning behaviour, which is strongly connected to the physics of the problem. The objective of the developed neural network model is to predict drag of a particular class of roughness contained in one high-fidelity dataset, which is composed of irregular homogeneous rough surfaces. However, we foresee that transfer learning will play a central role in combining empirical relations with larger datasets from a range of sources to develop predictive models of a significantly larger class of rough walls than those considered here. The paper is organized as follows: the details of the developed transfer learning framework including the neural networks and datasets are explained in § 2. The results of learning a mapping of surface statistics to the Hama roughness function using the transfer learning framework are shown in § 3. The effects of transfer learning are analysed and discussed in § 4, followed by the concluding remarks in § 5.

2. Transfer learning framework

The developed transfer learning framework is composed of two parts: (1) pre-training step and (2) fine-tuning step (figure 1). In the pre-training step, neural networks were trained to learn an 'approximate' knowledge of the mapping of surface statistics to the Hama roughness function. The training data are created by evaluating empirical correlation functions given the surface statistics of different surface topographies. In the fine-tuning step, neural networks were tuned to learn high-fidelity physics from a small DNS dataset by adapting their pre-trained 'approximate' knowledge to the domain of real physics. In § 3, we show that transfer learning of empirical correlations improves the generalization ability of neural networks when learning the drag of rough surfaces from a small amount of data. First, however, we explain the details of the neural networks and pre-training and fine-tuning steps.

2.1. *Neural network*

We used a variation of the MLP type neural network architecture developed by Jouybari *et al.* (2021). The input of the employed network is a set of 17 different rough surface statistics calculated from the surface topography of interest. The 17 input parameters are composed of eight primary surface statistics and nine products of the eight primary statistics. These parameters are selected based on their perceived importance on affecting drag on rough surfaces (Jouybari *et al.* 2021).

Let x, y and z be the streamwise, wall-normal and spanwise directions, $k(x, z)$ the roughness height distribution and A_t the total roughness plan area. Then, one may define

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

$I_1 = k_{rms}/Ra$	$I_2 = Sk = \dfrac{1}{A_t}\int_{x,z}(k-k_{avg})^3\,\mathrm{d}A/k_{rms}^3$
$I_3 = kur = \dfrac{1}{A_t}\int_{x,z}(k-k_{avg})^4\,\mathrm{d}A/k_{rms}^4$	$I_4 = ES_x = \dfrac{1}{A_t}\int_{x,z}\left\lvert\dfrac{\partial k}{\partial x}\right\rvert\mathrm{d}A$
$I_5 = ES_z = \dfrac{1}{A_t}\int_{x,z}\left\lvert\dfrac{\partial k}{\partial z}\right\rvert\mathrm{d}A$	$I_6 = por = \dfrac{1}{A_t k_c}\int_0^{k_c} A_f\,\mathrm{d}y$
$I_7 = inc_x = \tan^{-1}\left\{\dfrac{1}{2}Sk\left(\dfrac{\partial k}{\partial x}\right)\right\}$	$I_8 = \lvert inc_z\rvert = \left\lvert\tan^{-1}\left\{\dfrac{1}{2}Sk\left(\dfrac{\partial k}{\partial z}\right)\right\}\right\rvert$
$I_9 = I_4 I_5$	$I_{10} = I_4 I_2$
$I_{11} = I_4 I_3$	$I_{12} = I_5 I_2$
$I_{13} = I_5 I_3$	$I_{14} = I_2 I_3$
$I_{15} = I_4 I_4$	$I_{16} = I_5 I_5$
$I_{17} = I_2 I_2$	

Table 1. Statistical input parameters of the employed neural network.

the following measures of an irregular rough surface:

$$\left.\begin{aligned} &\text{average roughness height:} && k_{avg} &&= \int_{x,z} k\,\mathrm{d}A/A_t;\\ &\text{average height deviation:} && Ra &&= \int_{x,z} (\lvert k-k_{avg}\rvert)\,\mathrm{d}A/A_t;\\ &\text{root-mean-squared roughness height:} && k_{rms} &&= \left(\int_{x,z}(k-k_{avg})^2\,\mathrm{d}A/A_t\right)^{1/2}. \end{aligned}\right\} \tag{2.1}$$

In addition, we define the crest height k_c and finally the fluid area at each wall-normal plane $A_f(y)$. From these quantities, the 17 statistical input parameters ($\{I_1, I_2, \ldots, I_{17}\}$) can be calculated as listed in table 1. Here, Sk, kur, por, ES_x, ES_z, inc_x and inc_z indicate skewness, kurtosis, porosity, and effective slopes and average inclinations in x and z directions, respectively. Before training, the parameters I_1 to I_{17} were normalized with the mean and standard deviation values calculated from the dataset of empirical correlations (see § 2.2 for the details of the empirical dataset).

Information of input parameters travels through three hidden layers with 18, 7 and 7 neurons, respectively (see figure 2). Leaky rectified linear unit (ReLU) activations ($\max(0.01x, x)$) were applied after the hidden layers. The output of the employed neural network was a scalar value of the Hama roughness function ΔU^+. The superscript + indicates the normalization by the viscous scale $\delta_\nu = \nu/u_\tau$, where ν and u_τ are the kinematic viscosity and the friction velocity, respectively. In this study, the friction Reynolds number $Re_\tau = u_\tau H/\nu = 500$ was used, where H is the channel half-height. At the output layer, a Sigmoid activation function, $a/(1+\mathrm{e}^x)$, was applied to bound the prediction value ($a = 20$ in this study).

2.2. *Pre-training step*

In the pre-training step, neural networks were trained with a large dataset obtained from empirical correlation functions between rough surface statistics and the Hama roughness function ΔU^+. We used three correlations:

$$\Delta U^+ = 2.5\log Ra^+ + 1.12\log ES_x + 1.47; \tag{EMP1}$$

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

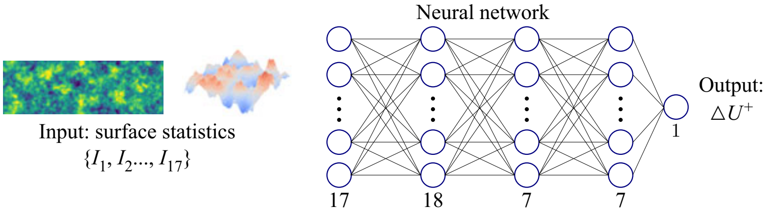

Figure 2. Architecture of the used neural network.

$$\Delta U^+ = 2.5 \log\left[k_c^+ \left(0.67Sk^2 + 0.93Sk + 1.3\right)\left(1.07\left(1 - \mathrm{e}^{-3.5ES_x}\right)\right)\right] - 3.5; \quad \text{(EMP2)}$$

and

$$\Delta U^+ = \begin{cases} 2.5 \log\left[2.48k_{rms}^+(1+Sk)^{2.24}\right] - 3.5, & Sk > 0 \\ 2.5 \log\left[2.11k_{rms}^+\right] - 3.5, & Sk \approx 0, \\ 2.5 \log\left[2.73k_{rms}^+(2+Sk)^{-0.45}\right] - 3.5, & Sk < 0. \end{cases} \quad \text{(EMP3)}$$

Here, (EMP1) was proposed by Chan *et al.* (2015), (EMP2) by Forooghi *et al.* (2017) and (EMP3) by Flack & Schultz (2010); Flack *et al.* (2020).

As mentioned earlier, we will develop neural networks to predict the drag of rough surfaces contained in a small high-fidelity dataset (see § 2.3 for the details of the DNS dataset). The range of the roughness function in this dataset was $5.3 < \Delta U^+ < 8.3$. The lower bound lay at the boundary between transitionally and fully rough regimes. Equation (EMP1) was derived using surfaces in the transitionally and fully rough regimes with $0.4 < \Delta U^+ < 11.4$. Equation (EMP2) was modelled from surfaces with roughness function in the range of $5.4 < \Delta U^+ < 10.1$. Finally, (EMP3) was developed with rough surfaces in the fully rough regime, where most of the surfaces became fully rough when ΔU^+ was over 5.5–6.0. Therefore, all three empirical models contained 'approximate' knowledge of the drag of rough surfaces that could help the neural networks to learn the DNS dataset in the fine-tuning step.

A large dataset was constructed by randomly generating 10 000 irregular rough surfaces and calculating the associated Hama roughness function using (EMP1), (EMP2) and (EMP3). The surfaces were constructed using the Fourier-filtering algorithm suggested by Jacobs, Junge & Pastewka (2017). The algorithm generated self-affine surfaces with a pre-defined power spectrum C,

$$C(q) \propto q^{-2-2h}, \tag{2.2}$$

where q is the wavenumber and h the Hurst exponent. The Hurst exponent $h = 0.8$ was used as in Jacobs *et al.* (2017). This power-law dependence is a well-known attribute of self-affine realistic surfaces (Mandelbrot 1982; Persson *et al.* 2004). Figure 3(*a*) shows a few examples of the generated random surfaces. The randomness was imposed by choosing random amplitudes of the power spectrum and random phase shifts of Fourier modes. Accordingly, the surface statistics, such as the skewness and the effective slope, depended on the particular combination of the random amplitudes and phases of Fourier modes.

For training and validating the neural network, 7000 and 3000 data samples were used, respectively. Figure 3(*b*) shows the scatter plots of the Hama roughness functions, calculated by (EMP1), (EMP2) and (EMP3), against k_{rms}^+, Sk and ES_x from the 3000 validation data samples. The surface generation method based on the spectral density (EMP1), generated surfaces with skewness bounded between −1 to +1 and effective

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

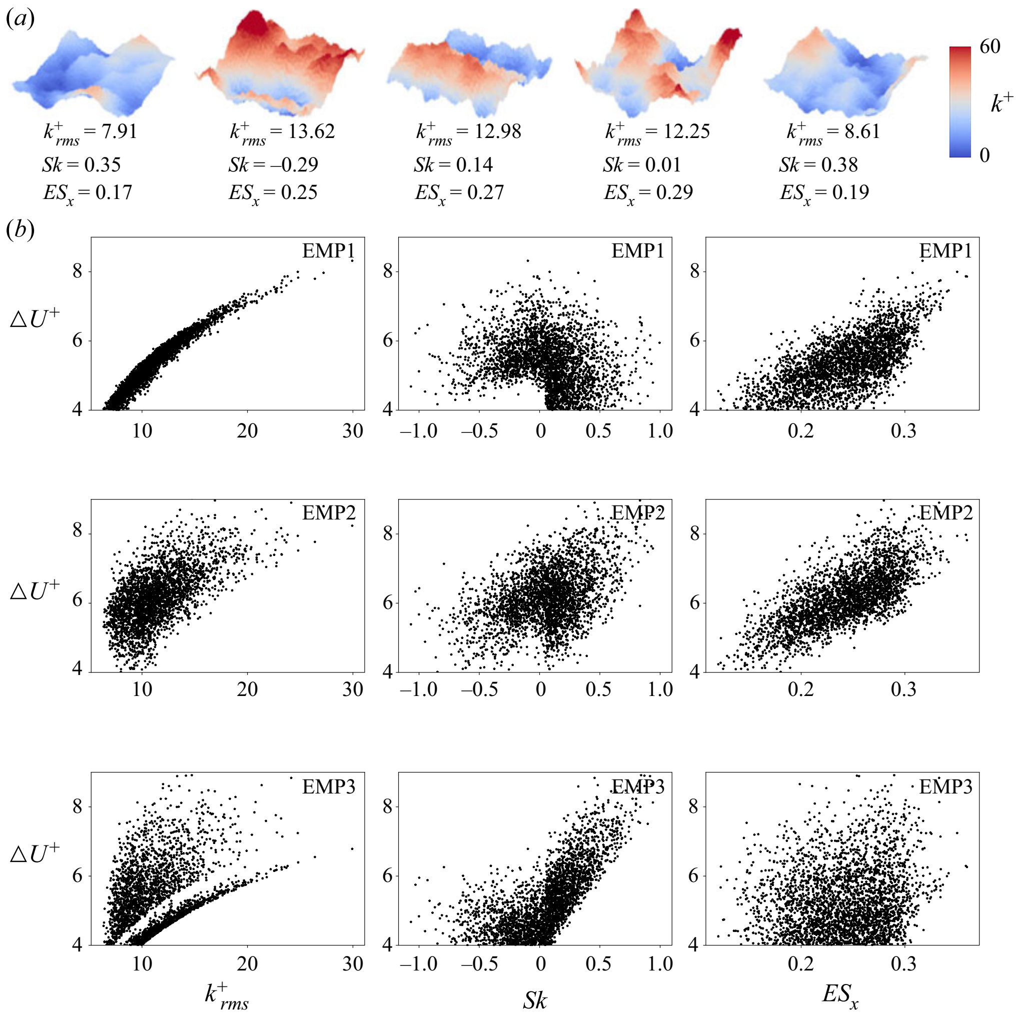

Figure 3. (*a*) Examples of patches of the generated surfaces in the empirical dataset. The size of the patches is 500 × 500 (viscous unit) in the streamwise and spanwise directions. The visualized heights of the surfaces are exaggerated five times to show the topography explicitly. (*b*) Scatter plots of the Hama roughness functions (ΔU^+), calculated by (EMP1), (EMP2) and (EMP3), against k^+_{rms}, *Sk* and ES_x from the 3000 validation data samples.

slopes limited to $\lesssim$0.35. The bounded skewness and limited effective slopes arose from the wavy surfaces generated by Fourier modes and the self-affine power spectrum defined for realistic surfaces, as in Jacobs *et al.* (2017). The random surfaces were thus considered to be in the low-slope regime, where the drag tends to increase with increasing effective slope (Flack & Schultz 2014). From figure 3(*b*), we observed different distributions, or knowledge, of the Hama roughness function from the different empirical models. The aim is to seed this diverse knowledge of empirical correlations in neural networks during the pre-training step.

Three neural networks with the same architecture and the same initial weights and biases were trained simultaneously to learn the different empirical models, (EMP1) to (EMP3).

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

Let NN^i be the neural network trained with the ith empirical model, $W^i_{j,k}$ the weight matrix connecting the jth and kth layers of NN^i, N_w the total number of weights in NN^i, b^i_j the bias vector added in the jth layer, ΔU^+_i the Hama roughness function predicted by the ith empirical model and $\Delta \tilde{U}^+_i$ the Hama roughness function predicted by NN^i. The neural networks were trained to minimize a loss for pre-training:

$$L_p = \frac{1}{3}\sum_{i=1}^{3}\left[\left(\Delta U^+_i - \Delta \tilde{U}^+_i\right)^2 + \sum_{j,k} \mathrm{tr}\left\{(W^i_{j,k})^{\mathrm{T}}(W^i_{j,k})\right\}\right]. \tag{2.3}$$

The first term in the right-hand side is the mean-squared-error loss and the second is the weight regularization loss, also used in Jouybari *et al.* (2021). The sizes of the weight matrix $W^i_{j,k}$ and bias vector b^i_k are $n_j \times n_k$ and n_k, respectively, where n_j and n_k are the number of neurons on the jth and kth layers.

The weights and biases of the neural networks were updated in the direction of minimizing the pre-training loss L_p on the training data samples through the Adam optimizer (Kingma & Ba 2014). The pre-training loss L_p on the validation data samples was also evaluated after each iteration of updates. Batch sizes of 16 and 32 were used for training and validating, respectively. To provide a wide distribution of data samples in the batches, i.e. preventing neural networks from learning biased data focused on average characteristics, we imposed batches to contain 50 % of the data samples with ΔU^+ in the range of 6 to 8, 25 % of the data samples with ΔU^+ in the range of 4 to 6 and the remaining 25 % of the data samples with ΔU^+ in the range of 8 to 10. After a sufficient number of iterations for training and validating, the set of networks ($\{NN^1, NN^2, NN^3\}$) that produced the lowest pre-training loss on the validation data samples was chosen to be merged into a single pre-trained network NN^{pre}. The learned sets of weights and biases in the three neural networks, $\{W^1_{j,k}, W^2_{j,k}, W^3_{j,k}\}$ and $\{b^1_k, b^2_k, b^3_k\}$, were averaged into the weights and biases of $W^{pre}_{j,k}$ and b^{pre}_k as $W^{pre}_{j,k} = \sum_{i=1}^{3} W^i_{j,k}/3$ and $b^{pre}_k = \sum_{i=1}^{3} b^i_k/3$. These pre-trained weights and biases were employed as the initial weights for the fine-tuning step (§ 2.3) to transfer the knowledge of empirical correlations to the domain of high-fidelity physics.

A key factor for the performance of the fine-tuned neural network is the contents of the database used in the pre-training step. If the bound of the ΔU^+ in one's high fidelity dataset is known *a priori*, then using a narrow range of ΔU^+ for pre-training increases the transfer learning performance significantly. We provide an example of this by training the neural networks with a bound of ΔU^+ in Appendix A. In the main text, we assume that the range of ΔU^+ of the high-fidelity dataset is *a priori* unknown.

2.3. *Fine-tuning step*

A total of 35 DNSs of channel flow over rough surfaces at $Re_\tau \approx 500$ were conducted to construct a high-fidelity dataset. The 35 rough surfaces were generated following the method proposed by Pérez-Ràfols & Almqvist (2019). Figure 4(*a*) shows a few examples of the generated surfaces. Note that different surface generation methods were used to construct the DNS and empirical datasets (§ 2.2). The method used for DNS imposed an additional constraint on the roughness height probability distribution. This constraint enabled the generation of rough surfaces with distinctive characteristics.

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

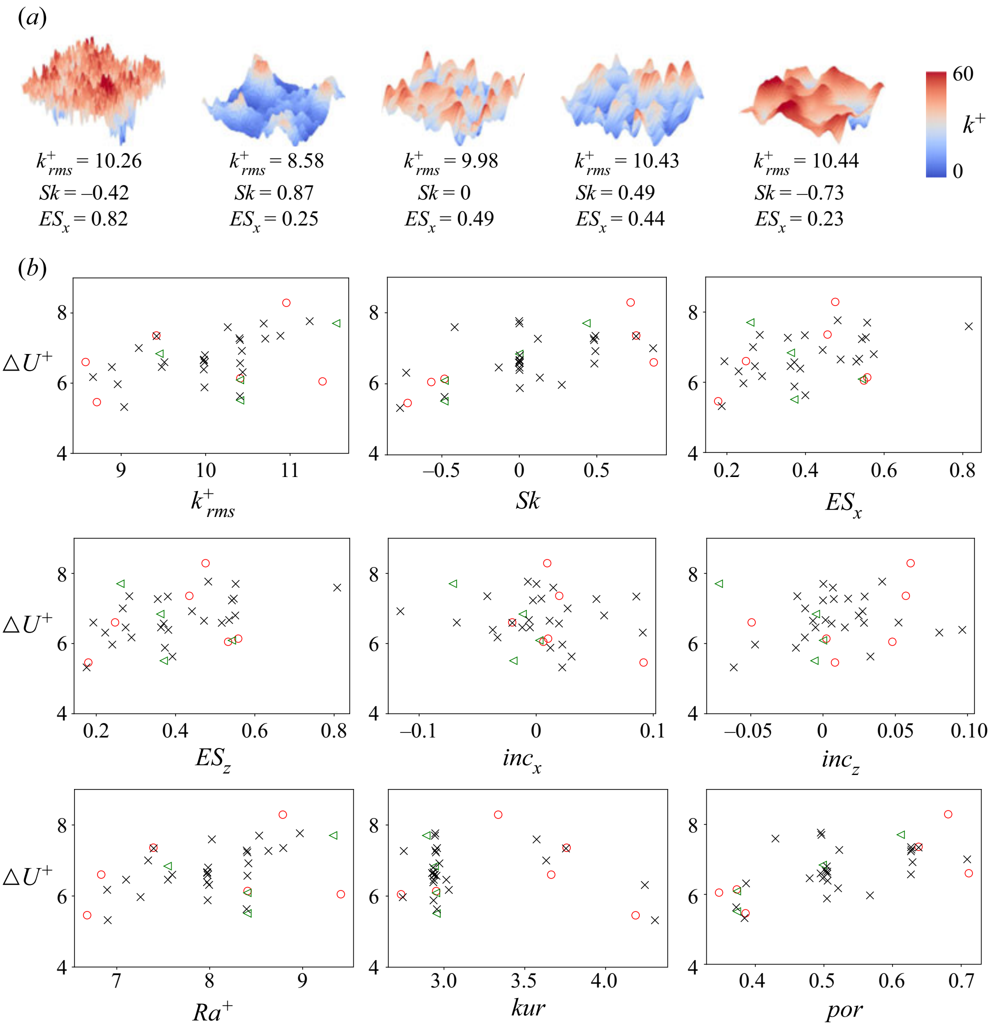

Figure 4. (*a*) Examples of patches of the generated surfaces in the DNS dataset. The size of the patches is 500 × 500 (viscous unit) in the streamwise and spanwise directions. The visualized heights of the surfaces are exaggerated five times to show the topography explicitly. (*b*) Scatter plots of the calculated Hama roughness function (ΔU^+) against various surface statistics of the 35 DNS data samples. The markers ○ (red), ◁ (green) and × (black) indicate the training, validation and test data samples, respectively. The plots are drawn from an example of a combination of training and validation data samples.

For the probability distribution, we used Weibull distributions of the form:

$$f(x; \lambda, s) = \frac{s}{\lambda}\left(\frac{x}{\lambda}\right)^{s-1} \mathrm{e}^{-(x/\lambda)^s}. \tag{2.4}$$

By using different shape (s) and scale (λ) parameters, rough surfaces with non-zero skewness were generated. Gaussian distributions were used to generate rough surfaces with zero skewness. Rough surfaces generated with (EMP3) imitate the roughness caused by wear (Pérez-Ràfols & Almqvist 2019). By using different surface generation methods in the pre-training and fine-tuning steps, we can verify that pre-training on one class of rough surfaces can help learn the drag on a different class of rough surfaces. Such verification is

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

important, as in practice, artificially generated rough surfaces for pre-training would not perfectly imitate real-world rough surfaces.

Details of the methodology and validation of the conducted DNS are reported in the study of Yang *et al.* (2021). Here, we provide a brief overview. A pseudo-spectral incompressible Navier–Stokes solver, SIMSON (Chevalier, Lundbladh & Henningson 2007) with an immersed boundary method (Goldstein, Handler & Sirovich 1993), was employed for the channel simulations. Periodic boundary conditions were used in the streamwise and spanwise directions. No-slip conditions were applied at the top and bottom planes of the channel. Roughness effects were added to both the top and bottom planes. A minimal channel approach was adopted to find a small computational domain size that could accurately reproduce the Hama roughness function calculated from a full channel simulation (Chung *et al.* 2015). Further details on the domain size and grid resolution are provided in Appendix B.

The scatter plots of the calculated Hama roughness function on the 35 data samples against the surface statistics involved in calculating the input of neural networks are shown in figure 4(*b*). The generated surfaces in the fine-tuning and pre-training steps showed moderately different characteristics. For instance, the surfaces generated in the fine-tuning step showed higher limits of the effective slopes (up to ∼0.8) compared with those in the pre-training step (up to ∼0.35). However, note that despite the larger effective slope, the surfaces were not in the regime where drag tends to decrease with increasing effective slope (Flack & Schultz 2014). The 35 data samples were split into 6, 4 and 25 data samples for training, validation and test, respectively. More than 70 % of the data samples were used for testing, i.e., not used during the fine-tuning step, to fairly evaluate the generalization ability of the developed neural networks. Note that a total of 10 data samples were used in the fine-tuning step and these fine-tuning data samples were completely separated from the test data samples. The data used for fine-tuning and testing the neural networks are provided in table 2. The surface statistics and the Hama roughness function in the test dataset are distributed widely in the total dataset as shown by the × markers in figure 4(*b*).

To reduce the uncertainty arising from splitting a very small dataset into training and validation sets, we adopted an approach based on averaging many neural networks. More specifically, we employed 210 neural networks, $\{NN^1, NN^2, \ldots, NN^{210}\}$, to learn from the 210 different data combinations derived by selecting six training and four validation data samples from the ten fine-tuning data samples ($\binom{10}{6} = \binom{10}{4} = 10!/(6!\,4!) = 210$). The weights and biases of these 210 neural networks ($\{W^1_{j,k}, W^2_{j,k}, \ldots, W^{210}_{j,k}\}$ and $\{b^1_k, b^2_k, \ldots, b^{210}_k\}$) were initialized by the pre-trained weights and biases, $W^{pre}_{j,k}$ and b^{pre}_k. Similar to the pre-training step, the weights and biases of each neural network NN^i were updated by the Adam optimizer that minimizes a loss for fine tuning L^i_f,

$$L^i_f = \left(\Delta U^+ - \Delta \tilde{U}^+_i\right)^2 + \frac{1}{N_w}\sum_{j,k} \mathrm{tr}\left\{(W^i_{j,k})^{\mathrm{T}}(W^i_{j,k})\right\}. \tag{2.5}$$

Here, ΔU^+ is the ground truth and $\Delta \tilde{U}^+_i$ is the roughness function predicted by NN^i. After a sufficient amount of updates, the networks with the smallest fine-tuning loss on the validation data samples were chosen to predict the roughness function ΔU^+ on test data (§ 3).

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

Fine-tuning data	ΔU^+	k^+_{rms}	Sk	ES_x	ES_z	inc_x	inc_z	Ra^+	kur	por
F1	5.46	8.712	−0.721	0.178	0.180	0.092	0.008	6.681	4.186	0.386
F2	5.51	10.408	−0.484	0.371	0.370	−0.019	−0.006	8.406	2.954	0.373
F3	6.05	11.385	−0.568	0.549	0.533	0.006	0.048	9.411	2.734	0.347
F4	6.09	10.407	−0.483	0.543	0.542	0.003	0.000	8.405	2.951	0.373
F5	6.14	10.411	−0.483	0.557	0.558	0.010	0.002	8.407	2.952	0.373
F6	6.60	8.578	0.868	0.248	0.247	−0.020	−0.049	6.832	3.664	0.711
F7	6.84	9.453	−0.001	0.363	0.361	−0.012	−0.005	7.547	2.943	0.498
F8	7.36	9.417	0.755	0.456	0.435	0.020	0.057	7.394	3.759	0.638
F9	7.70	11.546	0.433	0.260	0.261	−0.071	−0.072	9.328	2.890	0.612
F10	8.29	10.956	0.718	0.476	0.476	0.009	0.061	8.786	3.336	0.681

Test data	ΔU^+	k^+_{rms}	Sk	ES_x	ES_z	inc_x	inc_z	Ra^+	kur	por
T1	5.32	9.035	−0.770	0.187	0.176	0.022	−0.062	6.902	4.306	0.384
T2	5.63	10.406	−0.483	0.399	0.392	0.030	0.033	8.397	2.956	0.372
T3	5.88	9.986	0.003	0.372	0.373	0.012	−0.019	7.978	2.936	0.504
T4	5.97	8.958	0.274	0.242	0.240	0.022	−0.047	7.256	2.745	0.567
T5	6.17	8.667	0.132	0.289	0.289	−0.033	−0.013	6.897	3.030	0.521
T6	6.31	10.439	−0.730	0.230	0.223	0.091	0.080	7.993	4.243	0.387
T7	6.39	9.978	0.001	0.387	0.382	−0.037	0.096	7.972	2.934	0.505
T8	6.46	8.889	0.001	0.271	0.274	−0.012	−0.005	7.100	2.937	0.498
T9	6.46	9.481	−0.133	0.362	0.363	−0.005	0.015	7.550	3.015	0.479
T10	6.57	10.408	0.484	0.371	0.370	0.019	0.006	8.406	2.954	0.627
T11	6.59	9.986	0.003	0.529	0.517	−0.020	0.028	7.977	2.937	0.505
T12	6.60	9.512	−0.004	0.194	0.193	−0.068	0.052	7.597	2.943	0.495
T13	6.65	9.976	0.001	0.489	0.472	0.011	−0.007	7.972	2.926	0.505
T14	6.67	9.980	0.002	0.537	0.535	−0.006	0.002	7.973	2.936	0.502
T15	6.80	9.992	0.003	0.574	0.551	0.058	0.025	7.983	2.933	0.504
T16	6.92	10.429	0.491	0.444	0.441	−0.116	0.027	8.416	2.972	0.629
T17	7.00	9.208	0.864	0.266	0.267	0.027	−0.012	7.337	3.633	0.709
T18	7.23	10.407	0.483	0.543	0.542	−0.003	0.000	8.405	2.951	0.627
T19	7.27	10.706	0.120	0.355	0.355	0.051	0.007	8.632	2.751	0.522
T20	7.28	10.400	0.480	0.554	0.547	0.004	0.017	8.399	2.946	0.627
T21	7.34	9.419	0.754	0.398	0.380	0.086	0.029	7.397	3.755	0.638
T22	7.35	10.886	0.491	0.284	0.283	−0.042	−0.018	8.792	2.957	0.627
T23	7.59	10.262	−0.418	0.815	0.808	0.015	0.007	8.023	3.573	0.429
T24	7.70	10.688	0.001	0.557	0.552	0.000	0.000	8.532	2.944	0.496
T25	7.77	11.233	−0.001	0.482	0.483	−0.007	0.041	8.970	2.946	0.495

Table 2. The data of roughness function (ΔU^+) and surface statistics used for fine-tuning and testing the neural networks.

3. Prediction of roughness functions

We compare the predicted Hama roughness functions on the 25 test data samples that were not included in the fine-tuning step. The roughness functions were calculated by the following prediction methods: (1) neural networks trained with transfer learning (NNTF); (2) neural networks trained without transfer learning (NN) and (3) empirical correlations (EMP). Neural networks in NN were trained without the pre-training step, i.e. the weights and biases of the networks were randomly initialized instead of being initialized by $W^{pre}_{j,k}$ and b^{pre}_k. Our focus here is on the comparison between NNTF and NN to demonstrate the effects of transfer learning. However, we also include EMP predictions in the comparison because these methods are widespread in the literature.

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

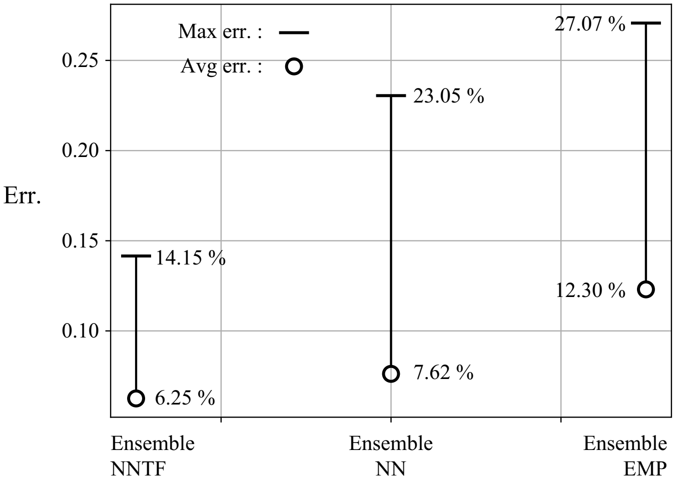

Figure 5. Average and maximum errors from the ensemble predictions performed by NNTF, NN and EMP.

The performance of neural networks can vary greatly depending on the selection of data for training and validation. This is because it is generally hard to form a distribution covering the entire feature space from a limited number of data samples. By ensemble averaging the predictions from neural networks learned from different data combinations, we can partly alleviate this problem. The ensemble predictions from NNTF and NN are thus obtained by averaging the prediction values of the 210 neural networks as

$$\Delta \tilde{U}^{+} = \sum_{i=1}^{210} \Delta \tilde{U}_i^{+}/210. \tag{3.1}$$

Similarly, we use ensemble predictions of the empirical models {(EMP1), (EMP2), (EMP3)} in EMP as

$$\Delta \tilde{U}_{emp}^{+} = \sum_{i=1}^{3} \Delta \tilde{U}_{i,emp}^{+}/3, \tag{3.2}$$

where $\Delta \tilde{U}_{i,emp}^{+}$ is the Hama roughness function predicted by the ith empirical model.

Figure 5 shows the maximum and average errors calculated from the ensemble predictions by NNTF, NN, and EMP. The predictions from NN exhibited 7.62 % and 23.05 % of the average and maximum errors, respectively, while the corresponding error associated with the predictions from NNTF were 6.25 % and 14.15 %. Note that both NN and NNTF are performing better than EMP here. The most significant advantage of using transfer learning is in the reduction of the maximum error; a nearly ten percent decrease is achieved by using NNTF instead of NN. As we will discuss more quantitatively in the next section, this error reduction demonstrates the capability of NNTF in learning a better generalized mapping from a small amount of DNS data.

To further investigate transfer learning, the best and worst performing neural networks out of the 210 networks in NNTF and NN were extracted and compared in figure 6. The errors from the best performing networks in NNTF and NN (figure 6*a*) showed a similar trend as the ensemble predictions (figure 5); both average and maximum errors were clearly reduced by using transfer learning. We also note that the ensemble predictions from

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

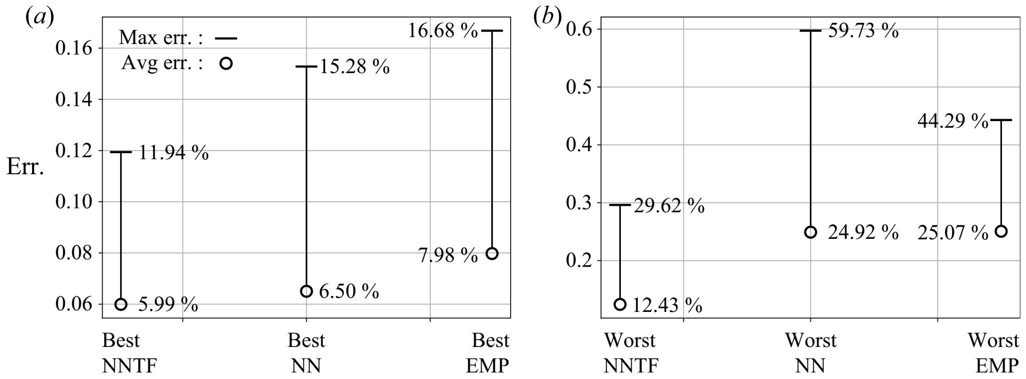

Figure 6. Average and maximum errors from the (*a*) best and (*b*) worst performing neural networks and empirical correlations.

NNTF (figure 5) performed marginally better compared with both the best performing neural network in NN and empirical correlation in EMP (figure 6*a*).

The advantage of using transfer learning was more clearly demonstrated in the worst performing case (figure 6*b*). The maximum error of the worst performing neural network in NN was nearly 60 %, which showed that this network failed to provide a reliable prediction. However, the network in NNTF exhibited significantly smaller errors, which indicated that transfer learning was helping the network to provide a safer prediction even in the worst scenario. Therefore, these results showed that transfer learning enables the networks to learn a more generalized mapping from limited DNS data by adapting an 'approximate' knowledge from empirical correlations. It is also important to emphasize that ensemble predictions should be used in practice to reduce the uncertainty arising from a limited pool of data as can be seen from the wide range of errors that occurred with the best and worst performing neural networks in figure 6.

4. Analysis of transfer learning and approximate knowledge

In the previous section, we demonstrated that transfer learning of empirical correlations improves the learning performances of neural networks. In this section, we analyse the effects of transfer learning and why 'approximate knowledge' in the empirical correlations can help neural networks.

4.1. *Effects of transfer learning on weight generalization*

In this section, we investigate how transfer learning improves the generalization ability by characterizing the learning weights of the networks. Let w_l^i, where $l = 1, 2, \ldots, N_w$ is the vectorized elements of the weight matrices $W_{j,k}^i$ in NN^i ($N_w = 488$ in this study). Then, we define the deviation of weights w_l^i from different NN^i as

$$\sigma_l = \frac{\sqrt{\sum_{i=1}^{210} \left(w_l^i - \mu_l\right)^2 \Big/ 210}}{\sum_{i=1}^{210} \left|w_l^i\right| \Big/ 210}, \tag{4.1}$$

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

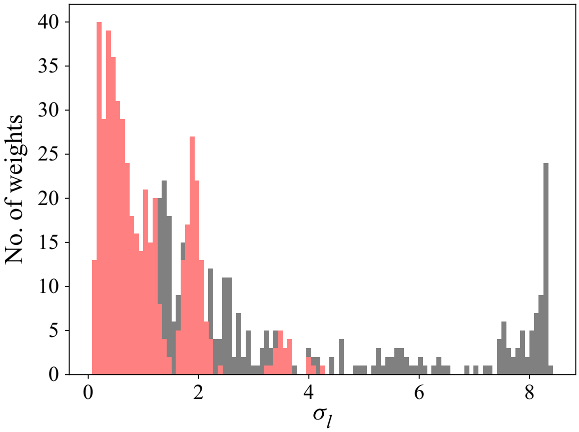

Figure 7. Distributions of the deviations of weights σ_l in NNTF (red) and NN (black).

where $\mu_l = \sum_{i=1}^{210} w_l^i/210$. Note that the networks in NNTF were initialized with the pre-trained weights, and the networks in NN were initialized with the same random weights. Therefore, the deviation σ_l indicates how a weight in a neural network is updated differently for different combinations of training and validation data samples (i.e. different NN^i). In the ideal case with an infinite amount of data samples, the deviation will approach zero as the network converges to specific weights representing a unique generalized mapping of the real physics.

The distributions of the deviation σ_l calculated from NNTF and NN are compared in figure 7. A number of weights with large σ_l were observed in NN, which indicated that the updates of weights largely deviated depending on the data. This implied that the current data were insufficient for NN to learn a generalized mapping. However, the deviations were significantly reduced in NNTF, which implied that the networks in NNTF were learning a better-generalized mapping compared with NN. This was because the networks in NNTF were initialized with the pre-trained weights containing the 'approximate' knowledge. These weights provided a better initial state for the weight optimization problem in the fine-tuning step.

4.2. *Effects of transfer learning on sensitivities of input surface statistics*

This section investigates how transfer learning affects the neural networks in learning the sensitivities of the input surface statistics. The sensitivities of input surface statistics in predicting drag were obtained by calculating the derivatives of drag with respect to each input (surface statistical measure). A similar analysis was employed by Kim & Lee (2020) for evaluating the influences of flow variables in predicting heat transfer characteristics. We define the sensitivity of the input I_i in predicting drag $\Delta\tilde{U}^+$ (defined in (3.1)) as

$$S_i = \left\langle \left| \frac{\partial \Delta \tilde{U}^+}{\partial I_i} \right| \right\rangle . \tag{4.2}$$

Here, $\langle\cdot\rangle$ is the averaging operation along with the calculated derivatives from the 25 test DNS dataset. The derivative $\partial\Delta\tilde{U}^+/\partial I_i$ can be analytically calculated as neural networks are composed of differentiable function compositions and matrix multiplications. However, more efficiently, it can be done by using an automatic differentiation algorithm.

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

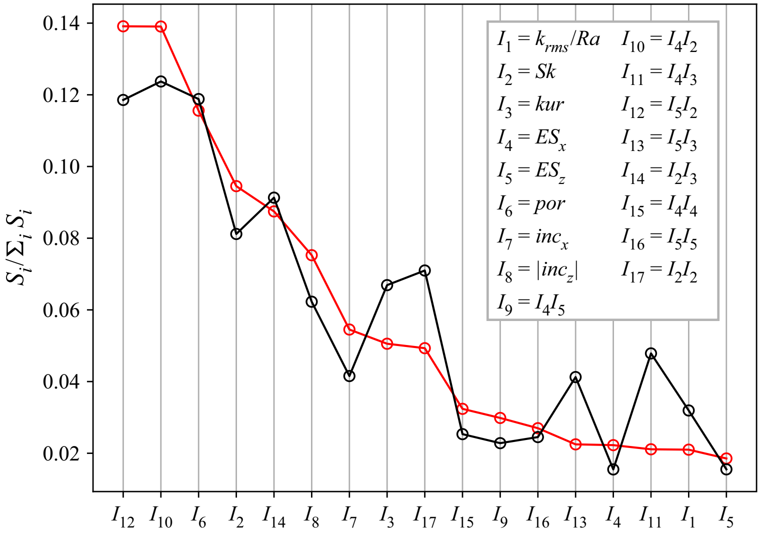

Figure 8. Sensitivities of the input surface statistics in NNTF (red) and NN (black). The sensitivities are normalized with their total sum ($S_i/\sum_i S_i$). The input surface statistics are sorted in descending order from the largest to smallest sensitive surface statistics in NNTF.

In this study, we used the automatic differentiation algorithm implemented in PyTorch (Paszke *et al.* 2017).

Figure 8 shows the sensitivities of the input surface statistics when trained with NNTF and NN. It was observed that for both neural networks, $I_{12} = ES_z \times Sk$, $I_{10} = ES_x \times Sk$ and $I_6 = por$ had the highest sensitivities. In other words, these were the most important surface measures when it came to predicting the friction drag for the particular class of rough surfaces contained in the high-fidelity dataset. These statistics are not explicitly included in the empirical models (EMP1), (EMP2) and (EMP3). Therefore, this indicates that the neural networks were learning the mapping of surface statistics to drag beyond the empirical models. However, the importance of I_{12}, I_{10} and I_{16} was partially implied by previous studies. The importance of the product of effective slope and skewness was partly implied in the study by Forooghi *et al.* (2017), as their empirical model (EMP2) was nonlinear with respect to skewness and effective slope. The importance of porosity in predicting drag was recently emphasized by Jouybari *et al.* (2021). Although the result shown here was for a particular class of rough surfaces, it demonstrated the necessity of expanding the feature space of surface statistics for drag prediction to more complex nonlinear combinations of the most basic statistical moments.

To understand what influence transfer learning has on sensitivities, we can identify from figure 8 the input statistics that are ranked differently by NN and NNTF in terms of importance. The surface statistics related to effective slopes and skewness, e.g. $I_{12} = ES_z \times Sk$, $I_{10} = ES_x \times Sk$, $I_2 = Sk$, $I_4 = ES_x$ and $I_5 = ES_z$, were found to be more sensitive in NNTF. These higher sensitivities in NNTF mainly arose from the inclusion of the approximate knowledge about the drag dependencies on effective slope and skewness in the empirical correlations. Conversely, the sensitivity of the input defined as the square of skewness ($I_{17} = Sk \times Sk$) was smaller in NNTF compared with NN. This indicated that NNTF learns that the square of skewness compromises the information about the sign of skewness, which is known to be important for drag prediction (Jelly & Busse 2018). In addition, NNTF reduces the sensitivities of statistics related to kurtosis ($I_3 = kur$,

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

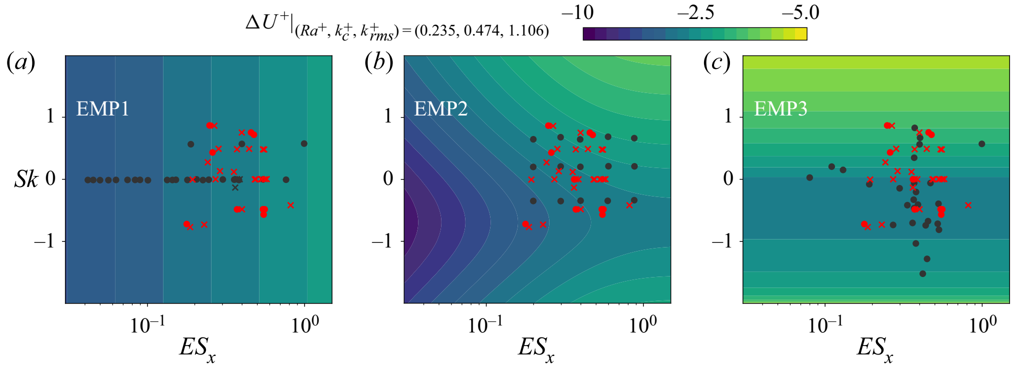

Figure 9. Contours of the roughness function calculated from (EMP1), (EMP2) and (EMP3). The roughness heights $(Ra^+, k_c^+, k_{rms}^+) = (0.234, 0.474, 1.106)$ are used for calculating the empirical correlations. The × (red) and ○ (red) markers indicate the test and fine-tuning DNS data samples, respectively. The ○ (black) markers indicate the fitting data samples used for deriving the empirical datasets (data extracted from Chung *et al.* 2021).

$I_{13} = ES_x \times kur$ and $I_{11} = ES_z \times kur$). As kurtosis measures the outliers (DeCarlo 1997), the reduction indicates that the effects of the outlier roughness heights are relaxed in NNTF.

4.3. *Approximate knowledge in the empirical correlations*

As briefly introduced in § 2.2, the empirical correlations contain different types of approximate knowledge. This is because the empirical correlations were fitted from different types of surfaces (Chung *et al.* 2021); (EMP1) was derived from data of a pipe with sinusoidal irregular roughness; (EMP2) was developed from data of rough surfaces with different arrangements and size distributions of roughness elements; and (EMP3) was constructed from a wide range of realistic rough surfaces, such as gravel and commercial steel pipes. As a consequence, each of the relations (EMP1), (EMP2) and (EMP3) depends on a different measure of roughness height (Ra^+, k_c^+ or k_{rms}^+) and different combinations of surface statistics (*Sk* and/or ES_x). Figure 9 shows contour levels of the drag predicted by (EMP1), (EMP2) and (EMP3) in a map spanned by skewness *Sk* and effective slope ES_x. Note that the roughness heights for (EMP1)– (EMP3) (Ra^+, k_c^+ and k_{rms}^+) were chosen by their respective values that produced $\Delta U^+ = -3.5$ at a common location ($ES_x = 0.3$ and $Sk = 0$) as in Chung *et al.* (2021).

The correlation (EMP1) contains the approximate knowledge that drag tends to increase with increasing effective slope, while it does not contain any dependency of drag with respect to skewness. This is because (EMP1) was derived from fitting data with a narrow regime of skewness, as shown by the black ○ markers in EMP1 of figure 9. Oppositely, the correlation (EMP3) contains the knowledge that drag tends to increase with increasing skewness, without a dependency on the effective slope. This is because the majority of the fitting data for (EMP3) (black ○ markers in EMP3 of figure 9) lies in an insensitive regime of effective slope – between the sparse regime ($ES < 0.3$–0.6) and dense regime ($ES > 0.4$–3.0) – with relatively small effects on drag (Jiménez 2004). Equation (EMP2) shows a nonlinear dependence of drag on the skewness and the effective slope. As a result, this model includes approximate knowledge about the drag dependency on the product of skewness and effective slope as also discussed in § 4.2.

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

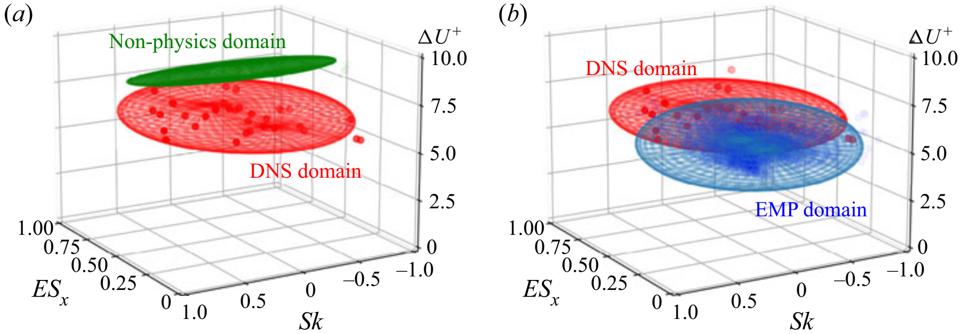

Figure 10. Visualization of the knowledge domains of the DNS domain (red), EMP domain (blue) and non-physics domain (green). Comparisons between the (*a*) DNS and non-physics domains and the (*b*) DNS and EMP domains.

As the empirical correlations are derived by fitting data from a particular set of surfaces, a naive extrapolation of them for predicting drag of other sets of rough surfaces may lead to large errors. For example, if the correlations are directly applied to predict drag on the current test surfaces of the DNS dataset (red × markers in figure 9), the average and maximum errors of (EMP1), (EMP2) and (EMP3) are (17 %, 29 %), (8 %, 17 %) and (25 %, 44 %), respectively. Note that the errors from (EMP2) are the smallest owing to the similar surface spaces between the current DNS data and the fitting data. Interestingly, large errors of the empirical models do not necessarily lead to inaccurate fine-tuned neural networks. To demonstrate this, we fine-tuned neural networks pre-trained only with one empirical correlation. We found that the average and maximum errors of the ensemble networks fine-tuned from (EMP1), (EMP2) and (EMP3) were (6 %, 18 %), (6 %, 17 %) and (7 %, 14 %), respectively. Despite the high errors of the empirical correlations of (EMP1) and (EMP3), their resulting fine-tuned networks were as effective as the fine-tuned network from (EMP2). This indicated that an empirical model for pre-training does not necessarily need to be accurate, it merely needs to provide an approximate knowledge of the dependencies on surface statistics. Moreover, NNTF, which is pre-trained with all of the approximate knowledge provided by the three empirical models, achieves a more favourable overall performance (figure 5), compared with the fine-tuned networks pre-trained with any single EMP. This is because the knowledge contained in each empirical model contributes with information of drag dependencies. This can be qualitatively shown by visualizing knowledge domains.

Figure 10 compares the knowledge domains of high-fidelity physics (DNS domain), approximate knowledge in the empirical models (EMP domain) and non-physical knowledge from the randomly initialized neural networks (non-physics domain). To visualize the knowledge domains, we extracted principal axes of the data samples composed of (ES_x, Sk, ΔU^+) in each domain. Note that ΔU^+ in the DNS, EMP and non-physics domains are obtained from simulations, empirical models and randomly initialized neural networks, respectively. The DNS domain is composed of rough surfaces in the DNS dataset, while the EMP and non-physics domains are composed of rough surfaces in the empirical dataset. After computing the principal axes in each domain, ellipsoids that approximately bound the DNS, EMP and non-physics domains along their principal axes are visualized. As shown in figure 10(*a*), no particular correlation (or rather alignment) can be found between the non-physics domain and the DNS domain. However, an alignment between the EMP domain and the DNS domain is observed in figure 10(*b*).

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

Therefore, it is expected that neural networks with approximate knowledge from the EMP domain can more easily adapt to the DNS domain compared with those without any physical knowledge.

We also quantified the alignment between the knowledge domains by calculating the angles between the principal axes. The angles between the principal axes in the EMP and DNS domains were computed as (16.5°, 18.1°, 8.9°), while the angles between the principal axes in the non-physics and DNS domains were computed as (120.9°, 119.9°, 12.3°). Accordingly, the aligned 'approximate knowledge' from EMPs assisted the networks to adapt to the DNS domain; thus, a better performance of neural networks was achieved by transfer learning of empirical correlations.

5. Conclusions

We have developed a transfer learning framework to learn the Hama roughness function from a limited number of DNS data samples. The framework is composed of two steps: (1) pre-training step and (2) fine-tuning step. In the pre-training step, neural networks learn an 'approximate' knowledge from empirical correlations. In the fine-tuning step, the networks are fine-tuned using a small DNS dataset. Neural networks trained with transfer learning show a significant improvement in predicting the roughness function on test data samples not included in the fine-tuning step. This is because the 'approximate' knowledge in empirical correlations is aligned to high-fidelity physics, which assists neural networks to learn better-generalized weights. In addition, a sensitivity analysis shows that the neural networks with transfer-learning were clearly emphasizing the importance of certain input surface statistics (effective slopes, skewness and porosity) in predicting drag. These extracted statistics are in general good agreement with what has been reported by other investigations, but also highlight that nonlinear functions of several surfaces statistics could provide high correlation with drag. We have shown that the prediction performance is enhanced when the 'approximate' knowledge of the empirical dataset is well aligned to the high-fidelity dataset. Therefore, it is advantageous to employ empirical correlations developed for classes of rough surfaces that are similar to one's high-fidelity dataset. The current NNTF, trained with irregular homogeneous rough surfaces, would not be effective when predicting drag on regular rough surfaces (e.g., cuboids, bars, etc.) or inhomogeneous rough surfaces. Similarly, predicting values of roughness function beyond training is also expected to not be effective owing to the general limitations of data-driven methods in extrapolation.

To further increase the generalization ability of neural networks, ultimately, the dataset has to be expanded. A large database would also help to study classes of rough surfaces with undiscovered empirical correlations, where pre-training is currently not possible. Accordingly, a collective community effort is needed to construct and train Big Data of flows over many classes of rough surfaces in different flow conditions. One step in this direction is the online database at http://roughnessdatabase.org (Engineering and Physical Sciences Research Council & University of Southampton 2020), currently under construction. Moreover, new primary surface statistics that strongly affect drag may be discovered in the future by extending the sensitivity analysis in § 4.2 with a larger surface parameter space and Big Data. Also, expansion of the network structure would enable one to fully leverage information inside large databases. The structure of the neural network in this study is very simple, as our focus has been to show that transfer learning can improve the performance of a given network. Thus, this network structure should not be considered as the optimal model for predicting drag. In addition to expanding the database, further

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

optimization of the neural networks will also improve the performance for drag prediction. The current transfer learning code will be made available online (e.g. see https://www.bagherigroup.com/research/open-source-codes/).

Finally, while we have considered transport of momentum (i.e. drag coefficient), empirical relations are widespread also for characterizing transport of energy (Nusselt number) and mass (Sherwood number). Therefore, the proposed framework can also be applied to other engineering applications where only a limited amount of high-fidelity data is available, but a significant amount of knowledge has been accumulated.

Funding. This work was supported by the Swedish Energy Agency under grant number 51554-1, Swedish Foundation for Strategic Research (FFL15-001), and Friedrich and Elisabeth Boysen Foundation (BOY-151). S.L. and S.B. would like to thank Dr O. Beijbom for the helpful discussions regarding the neural networks.

Declaration of interests. The authors report no conflict of interest.

Author ORCIDs.

Sangseung Lee https://orcid.org/0000-0001-7341-8289;

Jiasheng Yang https://orcid.org/0000-0003-0091-6855;

Pourya Forooghi https://orcid.org/0000-0001-9212-514X;

Alexander Stroh https://orcid.org/0000-0003-0850-9883;

Shervin Bagheri https://orcid.org/0000-0002-8209-1449.

Appendix A. Effects of a bounded roughness function in the datasets

Here, we provide an example of training the neural networks with bounded ΔU^+. If the bound of ΔU^+ in one's high-fidelity dataset is known, the transfer learning performance can be improved by accordingly bounding the ΔU^+ for pre-training. In this example, we used the bound of $\Delta U^+ > 6$, as the surfaces in the current DNS dataset asymptotically reach the fully rough regime when $\Delta U^+ \approx 6$ (Yang *et al.* 2021). We excluded the data samples with $\Delta U^+ < 6$ from the 35 DNS data samples (figure 4). As a result, a total of 29 DNS data samples were used in this section. The 29 data samples were split into 5, 3 and 21 data samples for training, validation and testing. Note that more than 70 % of the data samples were used for testing, as in § 3. The ΔU^+ of the empirical dataset was bounded to $\Delta U^+ > 6$ as the DNS dataset. After pre-training neural networks with the bounded ΔU^+, we fine-tuned a total of 56 neural networks, which learnt from the 56 different combinations of 5 training and 3 validation data samples out of the total 8 fine-tuning data samples ($\binom{8}{5} = \binom{8}{3} = 8!/(5!\,3!) = 56$).

The average and maximum errors on the 21 test data samples predicted by the ensembles of (1) NNTF with bounded ΔU^+ (NNTF-BU), (2) NN with bounded ΔU^+ (NN-BU) and (3) EMP with bounded ΔU^+ (EMP-BU) were (5.49 % and 14.60 %), (19.61 % and 45.61 %) and (13.22 % and 27.07 %), respectively (figure 11). The errors from NN-BU were notably increased compared with the errors from NN (7.62 % and 23.05 %, figure 5). This was because the number of data samples used for training NN-BU had decreased by ∼20 % compared with the number used for training NN. However, despite the decrease of the number of fine-tuning data samples, NNTF-BU showed similar errors compared with NNTF (6.25 % and 14.15 %, figure 5). Accordingly, NNTF-BU achieved nearly 15 % and 30 % decrease in error percentages compared with NN-BU, which was more significant than those achieved from NNTF (figure 5). Therefore, the transfer learning performance can be improved when the range of ΔU^+ of surfaces in one's high-fidelity dataset is *a priori* known. It is also worth mentioning that the errors from EMP-BU are similar to those

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

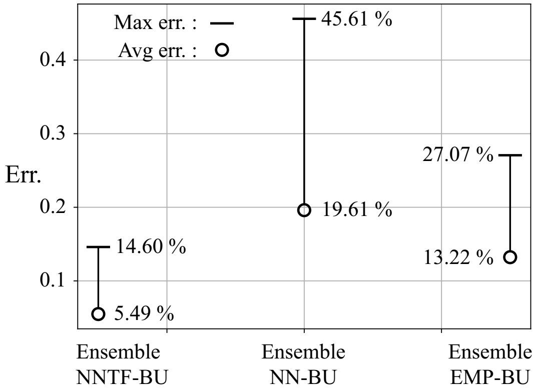

Figure 11. Average and maximum errors from the ensemble predictions performed by NNTF-BU, NN-BU and EMP-BU.

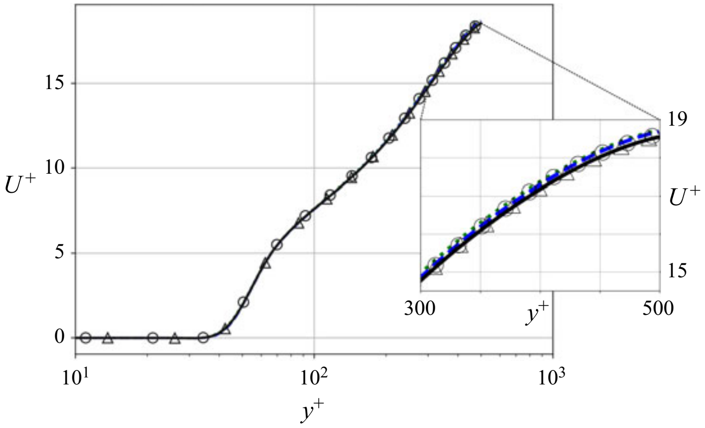

Figure 12. Profiles of ΔU^+ calculated from case 1 (green dotted line), case 2 (blue dashed line), case 3 (black solid line), case 4 (○) and case 5 (△).

from EMP (12.30 % and 27.07 %, figure 5), which indicates that the employed empirical models are reasonably valid even for ΔU^+ slightly under 6.

Appendix B. Grid resolution and domain size

To determine the appropriate mesh resolution in our simulations, a grid independence test was conducted. We conducted the test for the rough surface with the smallest Taylor microscale (λ_T) along with the 35 DNS data samples. The definition of the Taylor microscale proposed by Yuan & Piomelli (2014) was adopted in the current study. We studied the grid independence using five different resolutions in a minimal channel domain ($2.0H \times 2.0H \times 0.4H$): case 1, ($192 \times 401 \times 36$); case 2, ($256 \times 361 \times 48$); case 3, ($256 \times 401 \times 48$); case 4, ($256 \times 451 \times 48$); case 5, ($300 \times 401 \times 60$). The calculated mean velocity (U^+) profiles are shown in figure 12. The profiles from the five different cases were found to be nearly identical in both the inner and outer layers. Thus, in the current study, we used the grid resolution of case 3. This resolution corresponded to $\Delta^+ < 5$ in both streamwise and spanwise directions. In addition, with this resolution, the grid sizes were at least four times smaller than the Taylor microscales of all surfaces in

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

both streamwise and spanwise directions ($\lambda_T/\Delta > 4.5$), which satisfied the grid resolution constraint introduced in Jouybari *et al.* (2021).

The sensitivity of the domain size was also studied. We performed additional 12 full channel DNS to inspect the feasibility of using minimal channel DNS in the current study. For a given channel half-height H, the domain and grid sizes for the full channel simulations were respectively ($8H \times 2H \times 4H$) and ($900 \times 401 \times 480$) in the streamwise, wall-normal and spanwise directions. The smallest domain and grid sizes for the minimal channel simulations were ($2.0H \times 2.0H \times 0.4H$) and ($256 \times 401 \times 48$), respectively. The surface statistics of the rough surfaces were reasonably well converged in both domains. For instance, the maximum differences of k^+_{rms}, ES and Sk between the full and minimal channel domains for the rough surfaces were 0.038, 0.025 and 0.006, respectively. The resulting average and maximum errors between ΔU^+ calculated from the full and minimal channel domains were 1.5 % and 4.6 %, respectively. As these errors were notably smaller than those from the predictions of neural networks (§ 3 and Appendix A), we found the current minimal channel approach to be sufficient for our purposes. The full details of the surfaces and simulations are available in Yang *et al.* (2021).

References

Akhauri, S., Zheng, L.Y. & Lin, M.C. 2020 Enhanced transfer learning for autonomous driving with systematic accident simulation. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5986–5993. IEEE.

Chakraborty, S. 2021 Transfer learning based multi-fidelity physics informed deep neural network. *J. Comput. Phys.* **426**, 109942.

Chan, L., MacDonald, M., Chung, D., Hutchins, N. & Ooi, A. 2015 A systematic investigation of roughness height and wavelength in turbulent pipe flow in the transitionally rough regime. *J. Fluid Mech.* **771**, 743–777.

Chevalier, M., Lundbladh, A. & Henningson, D.S. 2007 Simson – a pseudo-spectral solver for incompressible boundary layer flow. *Tech. Rep.* TRITA-MEK 2007:07. KTH Mechanics.

Chung, D., Chan, L., MacDonald, M., Hutchins, N. & Ooi, A. 2015 A fast direct numerical simulation method for characterising hydraulic roughness. *J. Fluid Mech.* **773**, 418–431.

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

Colebrook, C.F. & White, C.M. 1937 Experiments with fluid friction in roughened pipes. *Proc. R. Soc. Lond.* A **161** (906), 367–381.

Dalili, N., Edrisy, A. & Carriveau, R. 2009 A review of surface engineering issues critical to wind turbine performance. *Renew. Sust. Energ. Rev.* **13** (2), 428–438.

DeCarlo, L.T. 1997 On the meaning and use of kurtosis. *Psychol. Methods* **2** (3), 292.

Engineering and Physical Sciences Research Council & University of Southampton 2020 Roughness database. Available at: http://roughnessdatabase.org/.

Flack, K.A. & Schultz, M.P. 2010 Review of hydraulic roughness scales in the fully rough regime. *Trans. ASME J. Fluids Engng* **132** (4), 041203.

Flack, K.A. & Schultz, M.P. 2014 Roughness effects on wall-bounded turbulent flows. *Phys. Fluids* **26** (10), 101305.

Flack, K.A., Schultz, M.P. & Barros, J.M. 2020 Skin friction measurements of systematically-varied roughness: probing the role of roughness amplitude and skewness. *Flow Turbul. Combust.* **104** (2), 317–329.

Forooghi, P., Stroh, A., Magagnato, F., Jakirlić, S. & Frohnapfel, B. 2017 Toward a universal roughness correlation. *Trans. ASME J. Fluids Engng* **139** (12), 121201.

Fukami, K., Fukagata, K. & Taira, K. 2021 Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *J. Fluid Mech.* **909**, A9.

Gent, R.W., Dart, N.P. & Cansdale, J.T. 2000 Aircraft icing. *Phil. Trans. R. Soc. Lond.* A **358** (1776), 2873–2911.

Goldstein, D., Handler, R. & Sirovich, L. 1993 Modeling a no-slip flow boundary with an external force field. *J. Comput. Phys.* **105** (2), 354–366.

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press

HAMA, F.R. 1954 Boundary layer characteristics for smooth and rough surfaces. *Trans. Soc. Nav. Archit. Mar. Engrs* **62**, 333–358.

JACOBS, T.D.B., JUNGE, T. & PASTEWKA, L. 2017 Quantitative characterization of surface topography using spectral analysis. *Surf. Topogr.* **5** (1), 013001.

JELLY, T.O. & BUSSE, A. 2018 Reynolds and dispersive shear stress contributions above highly skewed roughness. *J. Fluid Mech.* **852**, 710–724.

JIMÉNEZ, J. 2004 Turbulent flows over rough walls. *Annu. Rev. Fluid Mech.* **36**, 173–196.

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. & LEE, C. 2020 Prediction of turbulent heat transfer using convolutional neural networks. *J. Fluid Mech.* **882**, A18.

KINGMA, D.P. & BA, J. 2014 Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015*. ICLR.

LEE, S. & YOU, D. 2019 Data-driven prediction of unsteady flow over a circular cylinder using deep learning. *J. Fluid Mech.* **879**, 217–254.

MANDELBROT, B.B. 1982 *The Fractal Geometry of Nature*. WH Freeman.

MARTINEZ, M., SITAWARIN, C., FINCH, K., MEINCKE, L., YABLONSKI, A. & KORNHAUSER, A. 2017 Beyond grand theft auto V for training, testing and enhancing deep learning in self driving cars. PhD thesis, Princeton University.

MOODY, L.F. 1944 Friction factors for pipe flow. *Trans. Am. Soc. Mech. Engrs* **66**, 671–684.

PAN, S.J. & YANG, Q. 2009 A survey on transfer learning. *IEEE Trans. Knowl. Data Engng* **22** (10), 1345–1359.

PAN, X., YOU, Y., WANG, Z. & LU, C. 2017 Virtual to real reinforcement learning for autonomous driving. In *British Machine Vision Conference 2017, London, UK, September 4–7, 2017*. BMVA Press.

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 2017 Workshop on Autodiff, Long Beach, CA*.

PÉREZ-RÀFOLS, F. & ALMQVIST, A. 2019 Generating randomly rough surfaces with given height probability distribution and power spectrum. *Tribol. Intl* **131**, 591–604.

PERSSON, B.N., ALBOHR, O., TARTAGLINO, U., VOLOKITIN, A.I. & TOSATTI, E. 2004 On the nature of surface roughness with application to contact mechanics, sealing, rubber friction and adhesion. *J. Phys.: Condens. Matter* **17** (1), R1.

SCHULTZ, M.P., BENDICK, J.A., HOLM, E.R. & HERTEL, W.M. 2011 Economic impact of biofouling on a naval surface ship. *Biofouling* **27** (1), 87–98.

YANG, J., STROH, A., CHUNG, D. & FOROOGHI, P. 2021 DNS-based characterization of pseudo-random roughness in minimal channels. *J. Fluid Mech.* (submitted) arXiv:2106.08160.

YUAN, J & PIOMELLI, U 2014 Estimation and prediction of the roughness function on realistic surfaces. *J. Turbul.* **15** (6), 350–365.

https://doi.org/10.1017/jfm.2021.1041 Published online by Cambridge University Press