Machine learning for rapid discovery of laminar flow channel wall modifications that enhance heat transfer

Cite as: APL Mach. Learn. 2, 016108 (2024); doi: 10.1063/5.0187783
Submitted: 16 November 2023 • Accepted: 10 January 2024 •
Published Online: 25 January 2024

View Online Export Citation CrossMark

Yuri Koide,[1] Arjun J. Kaithakkal,[2] Matthias Schniewind,[1,a)] Bradley P. Ladewig,[3,b)] Alexander Stroh[2,c)] and Pascal Friederich[1,4,c)]

AFFILIATIONS

[1] Institute of Theoretical Informatics, Karlsruhe Institute of Technology, Engler-Bunte-Ring 8, 76131 Karlsruhe, Germany
[2] Institute of Fluid Mechanics, Karlsruhe Institute of Technology, Kaiserstr. 10, 76131 Karlsruhe, Germany
[3] Institute for Micro Process Engineering, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Karlsruhe, Germany
[4] Institute of Nanotechnology, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, 76344 Karlsruhe, Germany

[a)] **Current address:** Scientific Computing Center, Karlsruhe Institute of Technology, D-76128 Karlsruhe, Germany.
[b)] **Current address:** Paul Wurth Chair, Faculty of Science Technology and Medicine, University of Luxembourg, Luxembourg.
[c)] **Authors to whom correspondence should be addressed:** alexander.stroh@kit.edu and pascal.friederich@kit.edu

ABSTRACT

Numerical simulation of fluid flow plays an essential role in modeling many physical phenomena, which enables technological advancements, contributes to sustainable practices, and expands our understanding of various natural and engineered systems. The calculation of heat transfer in fluid flow in simple flat channels is a relatively easy task for various simulation methods. However, once the channel geometry becomes more complex, numerical simulations become a bottleneck in optimizing wall geometries. We present a combination of accurate numerical simulations of arbitrary, flat, and non-flat channels as well as machine learning models trained on simulated data to predict the drag coefficient and Stanton number. We show that convolutional neural networks (CNNs) can accurately predict target properties at a fraction of the computational cost of numerical simulations. We use CNN models in a virtual high-throughput screening approach to explore a large number of possible, randomly generated wall architectures. Data augmentation techniques are incorporated to enforce physical invariances toward shifting and flipping, contributing to precise prediction for fluid flow and heat transfer characteristics. Moreover, we approach the interpretation of the trained model to better understand relevant channel structures and their influence on heat transfer. The general approach is not only applicable to simple flow setups as presented here but can be extended to more complex tasks, such as multiphase or even reactive unit operations in chemical engineering.

I. INTRODUCTION

Heat transfer in fluid flow is an important physical phenomenon, with relevance across many science and engineering disciplines ranging from microfluidic devices in chemical engineering and biomedical implants, all the way to high-temperature physics and cosmology. In this proof-of-concept study, we explore an interesting engineering question, which can be posed as follows: "Is it possible to introduce structural changes to the wall of a channel that increases heat transfer, without having a corresponding increase in the pressure drop?" This fundamental question linked to the ultimate goal of *dissimilar flow control* or *dissimilar heat transfer enhancement* that has been asked for decades by various research groups in different application fields.

Dissimilar heat transfer enhancement is proven to be extremely challenging due to similarity in the mechanisms of momentum and heat transfer.[1] Investigations of various surfaces including specially designed fins,[2,3] dimples,[4] or vortex generators[5] report that an increase in heat transfer (where heat transfer is described by the Stanton number St) is always accompanied by inevitable manifold increase in pressure drop (described by the drag coefficient C_f). This eventually results in a decrease in the Reynolds analogy factor $RA = 2St/C_f$[6] in comparison to a flat channel configuration. It is, however, known that a dissimilar modification of momentum and heat transfer is possible when sophisticated flow control methods are applied. Those control methods, for instance, can be based on the introduction of flow perturbations or optimally distributed blowing/suction profiles from the wall surface.[7–11] These studies confirm that a significant enhancement of the Reynolds analogy factor (tripling RA in comparison to the uncontrolled channel flow) is possible when an appropriate flow manipulation is created. It is found that an introduction of large-scale spanwise rolls significantly promotes heat transfer while the drag coefficient remains less affected. This concept has also been successfully tested in the framework of turbulent channel flows, where $RA > 2$ can be achieved instead of $RA = 1$ in an uncontrolled flow configuration.[12,13] Recent studies in turbulent flows also report a possibility of RA modification using streamwise elongated structures, leading to the formation of turbulence-driven secondary motions.[14] The modification of RA is, however, limited in this case to several percent due to increase in the wetted area and the corresponding increase in C_f.

Most of the above studies were focused on turbulent flow regimes. To simplify the scenario, in the present study, we consider the laminar flow regime in a two-dimensional channel with heat transfer. The immersed boundary method is used for introducing the wall structuring. This approach allows for rapid execution of direct numerical simulations (DNSs), making it possible to investigate a wide range of arbitrarily generated surfaces.

Due to rapid growth in ML method development, there is also a growing interest in applying ML methods to challenges in the field of fluid dynamics. Seminal work on ML in fluid mechanics was focused on image processing based on convolutional neural networks (CNNs). CNNs have been successfully utilized as a straightforward method for learning from flow field data, reducing the computational cost.[15,16] For instance, deep CNNs have been used to construct velocity fields from image pairs of particle image velocimetry.[17] Furthermore, the CNN-based superresolution algorithm has contributed to turbulent flow reconstruction.[18] A recent study established two different models based on CNNs and encoder–decoder (ED) architectures to predict characteristics of the flow and heat transfer around the NACA.[19] The established CNN predicts the aerodynamic coefficients and the Nusselt number. However, there are still open challenges approaching dissimilar heat transfer enhancement and applying ML algorithms to this challenge. Furthermore, interpreting ML models can offer additional insights to complement human understanding and engineering intuition.

In this study, we present a workflow consisting of numerical simulations (Sec. II A), the key parameters for quantifying the performance of a thermo-fluid system (Sec. II B), and dataset generation along with the ML model (i.e., the CNN model) training (Sec. II C). In Sec. III, We show that the ML model can predict fluid flow and heat transfer characteristics with a large speedup compared to numerical simulations and with a high enough accuracy to prescreen a large database of possible channel geometries. Following an analysis of the velocity and temperature fields of some of the relevant structures in the training dataset (Sec. IV A), we interpret the trained ML model using SHapely Additive exPlanations (SHAP), which is a widely used explainable artificial intelligence method (Sec. IV B). Finally, we summarize the conclusions of the current study in Sec. V.

II. METHODS

A. Numerical procedure

For the problem setup, we consider a laminar channel flow with arbitrary wall structuring. The coordinate system of the numerical domain and its geometry ($L_x \times L_y = 10\delta \times 2\delta$ with δ being the half channel height) are illustrated in Fig. 1, where $(x, y) = (x_1, x_2)$ correspond to the streamwise and wall-normal directions, respectively. The velocity components in the two directions are denoted by $(u, v) = (u_1, u_2)$. The analysis is carried out using flow and temperature fields produced by DNS in a channel flow driven at a constant flow rate (CFR). Assuming an incompressible flow, the velocity field is required to satisfy continuity,

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

and the Navier–Stokes equations for a constant property Newtonian fluid:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \frac{1}{\rho}P_x - \frac{1}{\rho}\nabla p + \nu\nabla^2\mathbf{u} + \mathbf{F}_{\mathbf{IBM}}. \tag{2}$$

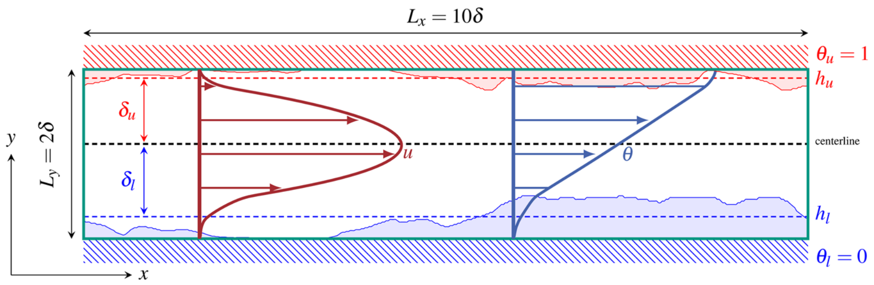


FIG. 1. Laminar channel flow with imposed wall structuring.

Here, p is the fluctuating pressure part, ρ is the density, ν is the kinematic viscosity, and $\mathbf{F_{IBM}}$ represents the external volume force per unit mass required for the immersed boundary method (IBM) with which the wall structuring is introduced into the flow domain.[20] In the present configuration, $\mathbf{F_{IBM}}$ corresponds to frictional drag between the flow and the part of the surface reproduced by the immersed boundary method, i.e., the structured wall surface. P_x is the absolute value of the mean streamwise pressure gradient added to the equation in order to drive the flow through the channel.

Due to the CFR approach, the bulk Reynolds number is fixed to $\mathrm{Re}_b = 2U_b\delta/\nu = 200$ for all the considered simulations, where $U_b = \frac{1}{2\delta}\int_0^{2\delta} u dy$ is the bulk mean velocity. This means that any modification of the flow is translated into an alteration of the resulting mean streamwise pressure gradient P_x required to maintain the chosen flow rate. Periodic boundary conditions are applied in the streamwise directions while the wall-normal extension of the flow domain is bounded by no-slip boundary conditions at the lower and upper domain walls $(y = 0, 2\delta)$. Subscript l and u are used throughout the manuscript to denote quantities on the lower and upper walls, respectively.

Temperature T is treated as a passive scalar and has to satisfy the scalar transport equation:

$$\frac{\partial T}{\partial t} + (\mathbf{u}\cdot\nabla)T = \alpha\nabla^2 T + Q_{IBM}, \tag{3}$$

where α denotes the thermal diffusivity. Periodic boundary conditions are applied for the thermal field in the x-direction, while constant temperature is prescribed on both the lower and upper walls of the flow domain. The non-dimensionalized temperature is defined as $\theta = (T - T_l)/\Delta T_w$ with $\Delta T_w = T_u - T_l$, such that $\theta_l = 0$ and $\theta_u = 1$. The Prandtl number is chosen to be $\mathrm{Pr} = \nu/\alpha = 1$. Q_{IBM} is proportional to the heat transfer rate between the flow and the structured wall and can be considered as the counterpart to the volume force $F_{\mathrm{IBM},x}$ in the momentum equation. This term is adjusted to fulfill the temperature boundary condition on the structured wall. Due to the use of periodic boundary condition for temperature, the absolute value of the heat transfer rate on the two walls should be identical once the solution reaches thermal equilibrium. For the same reason, the mean heat flux in the wall-normal direction is constant in the channel. The present thermal boundary condition is chosen following other studies of heat transfer above structured walls.[21–23]

The solver implementation is based on a spectral solver for incompressible boundary layer flows.[24] The Navier–Stokes equations are numerically integrated using the velocity-vorticity formulation by a spectral method with Fourier decomposition in the horizontal directions and Chebyshev discretization in the wall-normal direction. For temporal advancement, the convection and viscous terms are discretized using the third-order Runge–Kutta and Crank–Nicolson methods, respectively. The flow domain is discretized with $N_x \times N_y = 256 \times 129$ grid nodes, while the immersed boundary method is applied on the dealiased grid (3/2 rule) with $M_x \times M_y = 384 \times 129$ grid nodes.

B. Performance indices

Contrary to the laminar flow in a flat channel, no universal analytical solution can be derived for a channel with arbitrary structuring at both channel walls. Utilizing the meltdown heights of the imposed structure for both walls (h_u, h_l) and splitting the flow into two halves based on the position y_c of the maximal spatially averaged velocity (Fig. 1), the balance between pressure drop P_x and the average effective wall shear stress τ_{eff} is given by

$$\tau_{\mathrm{eff}} = \frac{(\delta_l + \delta_u)}{2} P_x, \tag{4}$$

where δ_u and δ_l define the upper and lower effective channel half heights with respect to y_c. Based on the wall shear stress the mean drag coefficient is given as

$$C_f = \frac{2\tau_{\mathrm{eff}}}{\rho U_b^{\mathrm{eff}\,2}}. \tag{5}$$

Here, the effective bulk mean velocity

$$U_b^{\mathrm{eff}} = \frac{1}{(\delta_u + \delta_l)}\int_0^{2\delta}\langle u\rangle \mathrm{d}y = \frac{2\delta}{(\delta_u + \delta_l)} U_b. \tag{6}$$

The brackets $\langle\rangle$ denote a quantity averaged in x-direction so that a split-up into the mean part $\langle\phi\rangle(y)$ and spatial fluctuation part $\phi'(x,y)$ can be performed for any quantity $\phi(x,y)$:

$$\phi(x,y) = \langle\phi\rangle(y) + \phi'(x,y). \tag{7}$$

Due to asymmetry in the temperature boundary condition, the heat transfer properties have to be separately evaluated for each wall. Hence, the hydraulic diameter is defined for the upper and lower walls as

$$D_{h,u/l} = 4\delta_{u/l}. \tag{8}$$

The Nusselt number for both walls can be estimated with

$$Nu_{u/l} = \frac{4\delta_{u/l} q_{\mathrm{tot}}}{\lambda\Delta\theta_{b,u}}, \tag{9}$$

where q_{tot} denotes the total heat flux and the bulk mean temperature differences are defined as

$$\Delta\theta_{b,l} = \frac{1}{\delta_l U_{b,l}^{\mathrm{eff}}}\int_0^{y_c}\langle u\rangle\langle\theta\rangle \mathrm{d}y, \tag{10}$$

and

$$\Delta\theta_{b,u} = \frac{1}{\delta_u U_{b,u}^{\mathrm{eff}}}\int_{y_c}^{2\delta}\langle u\rangle(1 - \langle\theta\rangle)\mathrm{d}y. \tag{11}$$

The average of Nu_l and Nu_u is computed to determine the resultant Nusselt number of a particular case. The effective bulk mean velocity for each channel half is given by

$$U_{b,l}^{\mathrm{eff}} = \frac{1}{\delta_l}\int_0^{y_c}\langle u\rangle \mathrm{d}y \quad \text{or} \quad U_{b,u}^{\mathrm{eff}} = \frac{1}{\delta_u}\int_{y_c}^{2\delta}\langle u\rangle \mathrm{d}y. \tag{12}$$

The total heat flux q_{tot}, which is a constant as mentioned previously, can be estimated as the sum of

$$q_{\mathrm{tot}} = \lambda\frac{\mathrm{d}\langle\theta\rangle}{\mathrm{d}y} - \rho c_p\langle v'\theta'\rangle + \rho c_p Q_{\mathrm{IBM}}^{y}, \tag{13}$$

where the three terms are, respectively, named the laminar, total fluctuation, and IBM contributions.[14] Here, c_p denotes the specific heat capacity and $Q_{\mathrm{IBM}}^{y} = -\int_y^{\delta} Q_{\mathrm{IBM}}\,\mathrm{d}y$. Finally, the Stanton number is defined based on $Re_{D_h} = 2(\delta_l + \delta_u)U_b^{\mathrm{eff}}/\nu$ and Prandtl number Pr:

$$St = \frac{Nu}{Re_{D_h}\,\mathrm{Pr}}. \tag{14}$$

Reynolds analogy factor RA relates Stanton number to the drag coefficient

$$RA = \frac{2St}{C_f}, \tag{15}$$

and it is used to evaluate the similarity between drag coefficient and heat transfer.[25] An increase in RA highlights a stronger enhancement in heat transfer compared to that in the drag coefficient and hence is desirable in the design of an energy-efficient thermo-fluid system. It has to be noted that under the specified boundary conditions $RA = 0.5333$, with $St = 0.016$ and $C_f = 0.06$, for the flat channel configuration, and can be derived analytically. Hereafter, the values associated with the flat channel configuration will be denoted with the subscript "ref." To find out the uncertainty in C_f and St calculations based on IBM simulations, we performed simulations of 150 flat channels that are located symmetrically or asymmetrically about $y = \delta$ (refer to Fig. 1). Though the mean absolute error (MAE) in C_f was found to be 0.05%, the maximum absolute error (MaxAE) was found to be 2.17%. Similarly, the MAE and MaxAE for St were found to be 0.01% and 5.62%, respectively. So, the expected MaxAE in RA is around 3.45%.

C. Dataset and machine learning model

To generate a diverse dataset of wall structuring, we used a random walk algorithm combined with spline interpolation and discretization on the simulation grid. Each wall structure consists of n supporting points (see Table I) between the start and the end x grid points located at $M_{x_s} = 0$ and $M_{x_e} = 384$. The y grid points M_{y_s} and M_{y_e}, corresponding to M_{x_s} and M_{x_e}, have the same y-position to satisfy periodic boundary conditions. The grid points along the x-axis for the supporting points are sampled from

$$M_{x_i} \sim \mathcal{N}\left(\mu = i\cdot\frac{384}{n},\ \sigma = \sigma_x\cdot\frac{384}{n}\right) \tag{16}$$

where i is the ith supporting point in the interval $[1, n]$. σ_x is varied according to Table I.

TABLE I. Parameters for the structure generation algorithm.

Parameter	Variations
n	$[2, 3, 4, 5, 6, 7, 8, 9, 10]$
σ_x	$[0.1, 0.2, 0.3, 0.4, 0.5, 0.6]$
σ_y	$[0.05, 0.1, 0.15, 0.2]$
r	$[0.0, 0.05, 0.1, 0.15, 0.2]$
a	$[0.0, 0.05, 0.1, 0.15, 0.2]$

The grid point along the y-axis at each respective M_{x_i} is sampled from

$$M_{y_i} \sim \mathcal{N}(\mu = M_{y_{i-1}},\ \sigma = \sigma_y\cdot\Delta), \tag{17}$$

where σ_y is varied from Table I and Δ the available build space in y-direction. A minimum of 50% of the channel height $L_y = 2\delta$ is kept empty for the flow. To allow for larger meanders, the generation algorithm of the first wall surface can use the full 50% of build space, so here Δ remains a constant equal to $M_y/2$ (i.e., 64.5) for all x positions. For the second channel surface, Δ is adjusted according to the first channel surface. It is important to note that the first M_{y_i} grid point, corresponding to M_{x_s}, is drawn from a uniform distribution in the interval $[0, \Delta]$.

The obtained supporting points are then interpolated with cubic Bézier curves. The distance of the control points from the supporting points is determined by the parameter r. For small values of r, the radius of the curves can get very small, resulting in sharp features. The parameter a controls the smoothness of the curve. The values of both r and a are varied in the range $[0, 0.2]$ as mentioned in Table I. For $a = 0$, the angle through one supporting point is determined by the mean of directions to both neighboring points. At higher a, the direction to one neighboring point is weighted higher and hence the curve features stronger edge.[26]

The obtained bottom and top wall curves were then projected onto the dealiased grid. Using 20 random initializations per parameter variation, 108 000 random wall structures were generated, hereafter called the *repository set*.

We calculated C_f and St using the simulation method described in Sec. II A for a randomly sampled subset of 10 800 channel geometries (10% of the repository set). The 10 800 calculations each requiring 20–30 min single core execution on an Intel Xeon Gold 6230 CPU result in a total cost of 3600–5400 CPU h. Out of these, 9185 met specific criteria related to temperature convergence and geometric validity. We set aside 5% (459) of these data to be included in the test set.

We used varying fractions of the remaining 8726 channel geometries (hereafter called the *labeled set*) for hyperparameter optimization of CNNs. Hyperparameters were determined without flat channel geometries. We used a total of 8776 channel geometries as the *training set* and 464 channel geometries as the *test set*. We added 50 flat channels with various channel heights to the training dataset and 5 to the test dataset. The inputs for the CNN are binary images where 1 and 0, respectively, represent solid and fluid domains. Each binary image with 384 × 129 × 1 pixels represents a cross section of the channel geometry, i.e., exactly the same input that is also used

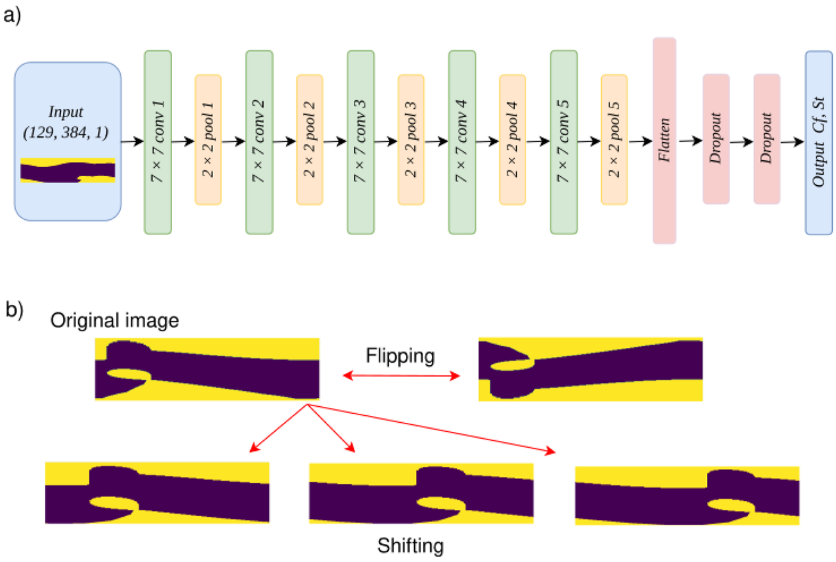


FIG. 2. (a) Overview of CNN architecture and (b) examples of input images when they are shifted and flipped.

TABLE II. The name of the hyperparameters, the range of values considered for their optimization, and the optimum value.

Name	Type	Min	Max	Optimum
Kernel size	Int	2	20	7
Num. of convolution steps	Int	2	5	5
Num. of filters	Int	1	128	128
Num. of neurons	Int	50	6000	6000
Final learning rate	Double	1.0×10^{-11}	1.0×10^{-4}	6.105×10^{-6}
Dropout	Double	0.05	0.5	0.3

in the numerical simulations. The input is passed through a varying *number of convolution steps*, each consisting of a convolution with padding, a varying *kernel size* and a varying *number of filters*, followed by a ReLU-activation and a 2x2x max.-pooling with stride 2. The output of these convolutions is then flattened and passed through one hidden dense layer with ReLU-activation and a varying *number of neurons*. This hidden layer is additionally regularized by a varying *dropout*. C_f and *St* are then predicted with an output layer with two neurons and linear activation. An overview of the optimized CNN model is shown in Fig. 2(a).

The model was implemented using TensorFlow[27] and Keras,[28] and training was done using the Adam optimizer.[29] The model was trained for 100 epochs with a batch size of 256 using the mean squared error (MSE) loss while logarithmically reducing the learning rate from 1×10^{-3} at the tenth epoch to a varying *final learning rate* at the last epoch.

The hyperparameter optimization was done with Bayesian optimization to find the most optimal set of hyperparameters, where optimal was defined as the lowest *validation* set error (see Table II). The reported errors of the model were then calculated on a separate test set. It ensures that overfitting to the training set is minimized and potential overfitting to the validation set during hyperparameter optimization is not included in the reported test set errors. For this, SigOpt was employed[30,31] with 140 experiments on eight asynchronous parallel channels. Each channel has access to a single Tesla A100 GPU. To ensure stable predictions, for each evaluation the

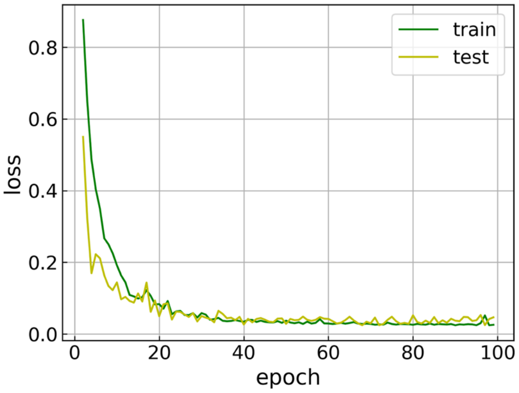


FIG. 3. Training and testing loss curves of the CNN model.

mean MSEs for the predicted C_f and St of a fivefold cross-validation were used as metrics.

Under the present flow and temperature conditions, where no volume forces are considered and temperature is treated as a passive scalar, C_f and St of a channel do not change when the channel geometry is flipped along the channel centerline or shifted along the streamwise direction. We used this information to augment the data during the ML model training. We divided input images into batches. For instance, 35 batches containing 256 images were created. Each half of the 256 images in each batch was flipped and shifted randomly during the training. The channel geometry was flipped about the y-axis and also randomly shifted in the x-direction within the range of M_{x_s} and M_{x_e} [see Fig. 2(b)]. Input images are shuffled every epoch so that the same images are not always flipped and shifted in the same order. Hence, multiple shifted and/or flipped copies of channel geometries from a single geometry were generated without the need for any additional numerical simulation. The effect of shifting and flipping on the test data is discussed in detail in the Appendix. In the end, the computational cost of training the model is negligible (around 20 min on a NVIDIA Tesla V100 GPU).

III. RESULTS

We trained the CNN model and obtained MAEs of 1.90×10^{-3} (3.16% of $C_{f,\mathrm{ref}}$) and 1.29×10^{-4} (0.81% of St_{ref}) for C_f and St predictions on the test dataset. This corresponds to an expected MAE of 1.25×10^{-2} in RA (2.35% of RA_{ref}). The respective r^2-scores for C_f and St are 0.951 and 0.925. Figure 3 shows training and testing loss as a function of epochs. The curves converge from around epoch

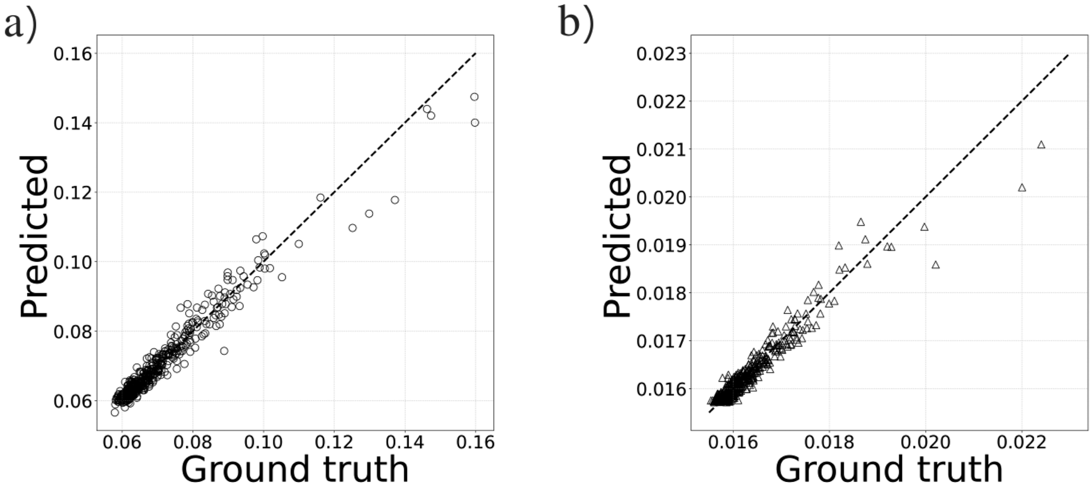


FIG. 4. Predictions of the CNN model of (a) drag coefficient C_f and (b) Stanton number St compared to the ground truth on the test set.

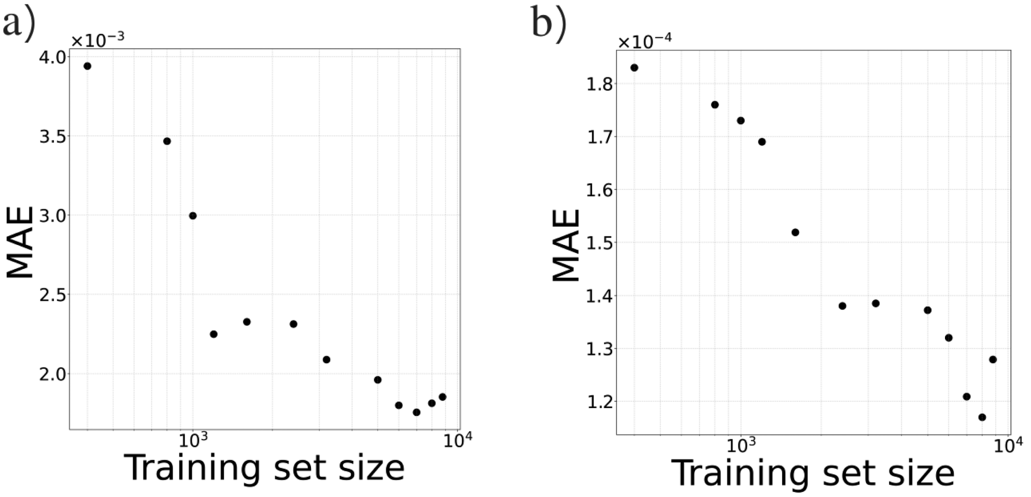
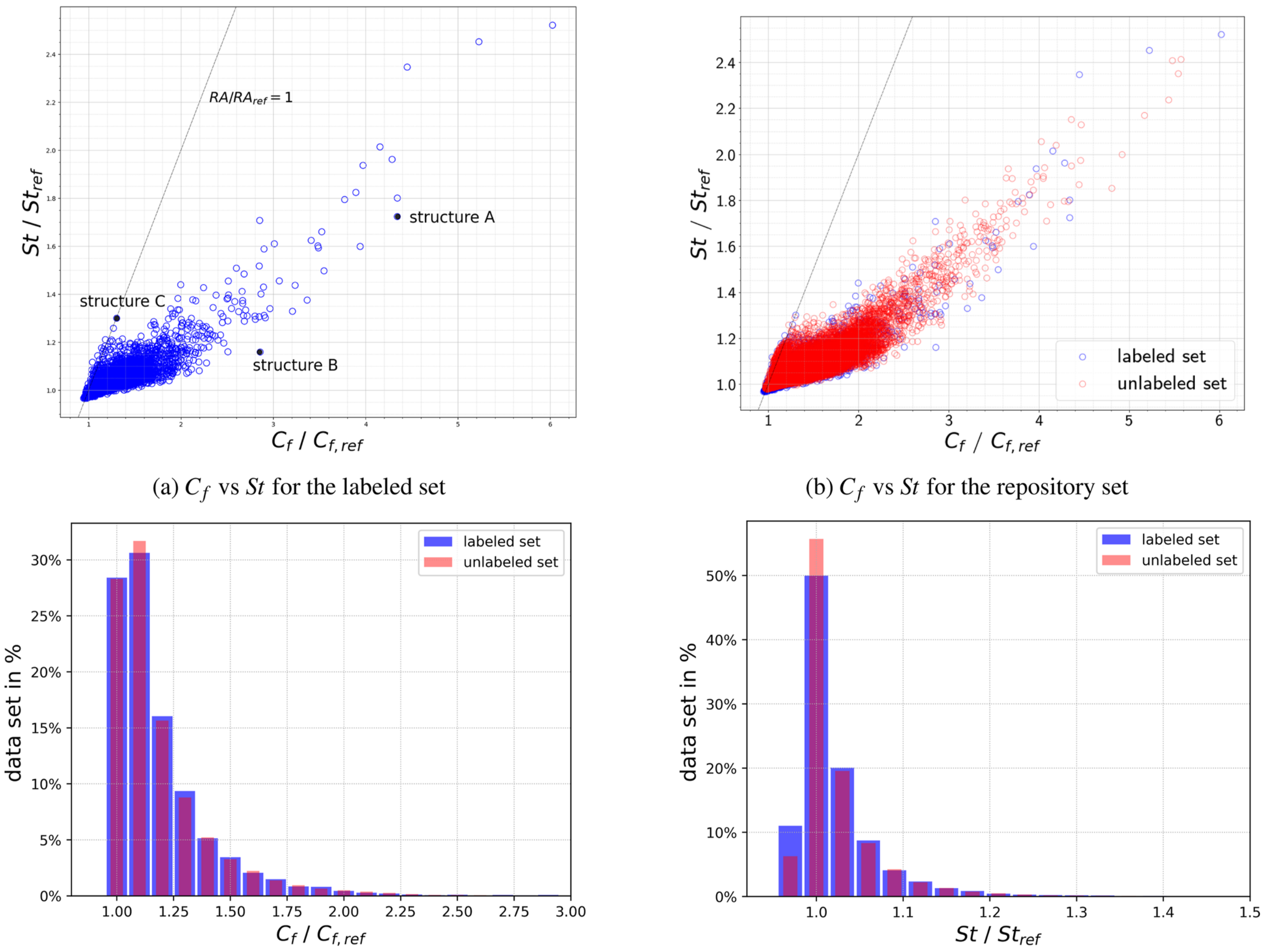


FIG. 5. Learning curve, i.e., mean absolute error as a function of the training data size of CNN models for (a) drag coefficient C_f and (b) Stanton number St.

20 toward 100. A comparison of CNN predictions with simulated ground truth values for the test set is shown in Fig. 4. Most values of C_f and St are well correlated. For a perfect model, the predictions and the ground truth values are the same, so all points would lie on the unity line. Deviations for large values can be seen (C_f = 0.12 ∼, St = 0.020 ∼), but at the same time, this only affects very few channels that are not of practical relevance.

To demonstrate the expected exponential behavior of the learning curve and to estimate necessary training set sizes for given target errors, we generated learning curves (see Fig. 5), where we observe the MAE in C_f and St as a function of the training set size. The hyperparameters were kept constant, and the amount of training data was varied from 5% to 90% of a total of training and test set, 9240 channel geometries. With each diverse dataset, every model is trained independently, and fluctuations are to be expected due to randomly sampled initial weights. We observe an exponential decrease in MAE with increase in the training set size. It can be seen from Fig. 5 that larger datasets increase the model accuracy. In the case of C_f, the MAE score for the test dataset is approximately twice smaller than that of 5%. In the case of St, the factor is slightly smaller. As the plots continue to decrease exponentially, further improvement in the model accuracy can only be expected by substantially increasing the number of data. However, we do not observe strong saturation, indicating that the model is not limited by capacity or incomplete input information.

The scatter plot of 8726 labeled data points from the repository set used for training the CNN is shown in Fig. 6(a). We would like to point out that the training set contains geometries that are well representative of the flow configurations usually encountered in fluid dynamics. Three of such structures that are highlighted in



(a) C_f vs St for the labeled set

(b) C_f vs St for the repository set

(c) Histogram representation of C_f

(d) Histogram representation of St

FIG. 6. Scatter plots of the normalized C_f vs St for (a) the labeled set and (b) the combined labeled and unlabeled sets from the repository. Histogram representation of the labeled and unlabeled sets for (c) C_f and (d) St.

Fig. 6(a) are discussed in detail in Sec. IV. Almost all the points fall below the $RA/RA_{\mathrm{ref}} = 1$ line in Fig. 6(a), indicating that deviations from a flat channel lead to an increase in C_f that outweighs the simultaneous increase in St. We then exploited the speedup of the surrogate machine learning model (<100 ms per channel) compared to the numerical simulation (≈20 – 30 min per channel) to explore the flow and heat transfer characteristics (C_f and St) of a much larger set of unlabeled channel geometries. The term *unlabeled set* is part of the repository set for which the ground truth information is absent, signifying that no simulations have been conducted for these geometries. The calculation of C_f and St for the unlabeled channel geometries took only 2.5 GPU hours. Thus, the benefit of training a machine learning model becomes evident when the model is used for inference on new structured channel geometries. The scatter plot of C_f vs St for the repository set, split up into labeled and unlabeled data points, is shown in Fig. 6(b). In general, it is very difficult to find geometries that can result in dissimilar heat transfer enhancement, i.e., $RA/RA_{\mathrm{ref}} > 1$. However, as can be seen from Figs. 6(a) and 6(b), there are some points very close to (and above) the $RA/RA_{\mathrm{ref}} = 1$ line. However, these points fall within the combined uncertainty level from the simulation results plus the ML prediction. The structures corresponding to these points closely resemble either a flat channel, the trivial solution, or the nontrivial geometry of structure C (see Sec. IV A). To come up with more innovative structures with $RA/RA_{\mathrm{ref}} > 1$, we can combine the speedup of the surrogate model with evolutionary optimization techniques.[32] It

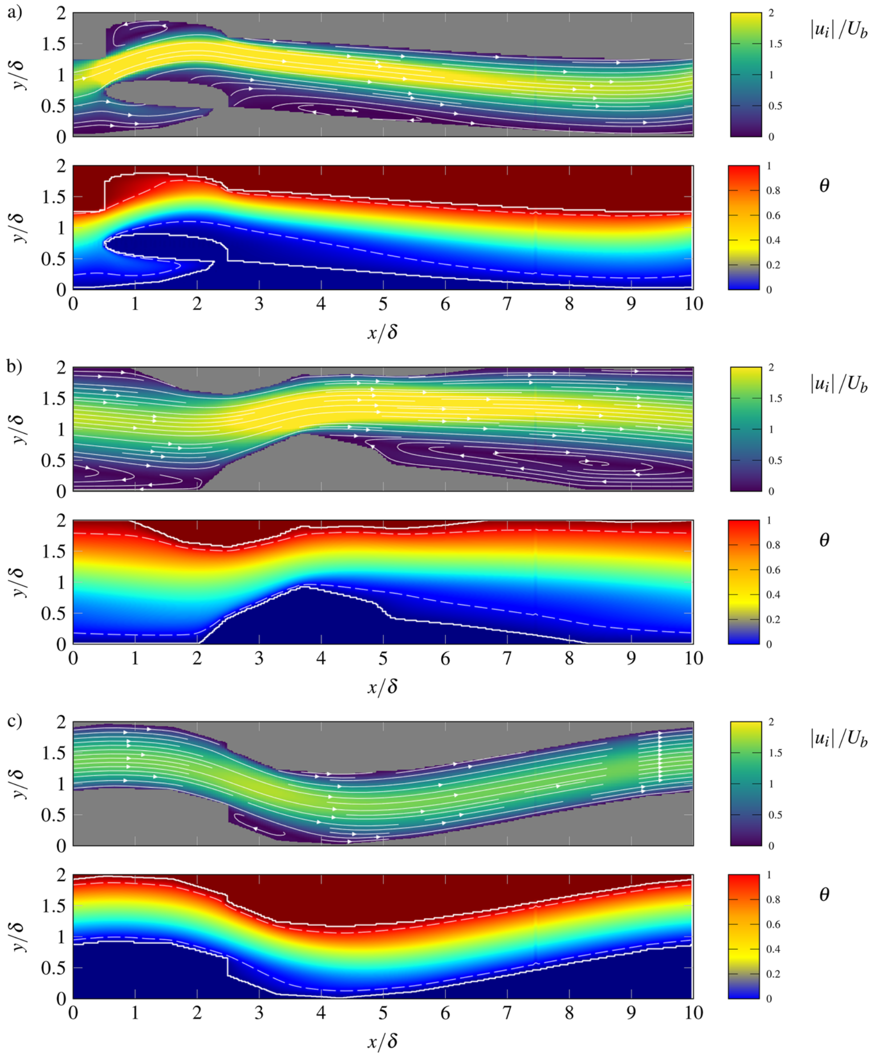


FIG. 7. Mean velocity and temperature fields for structures (a) structure A, (b) structure B, and (c) structure C. Streamlines are added on top of the velocity fields for a better understanding of the flow field, especially the recirculation regions. The white dashed lines in the temperature fields indicate $\theta = 0.10$ and 0.90 and are representative of the thermal boundary layer.

is essential to note that evolutionary optimization algorithms in general require the optimization of their own set of hyperparameters. In this context, having a surrogate ML model that can provide values for the quantities of interest (*RA* for example) in a fraction of a second is highly beneficial. Figures 6(a) and 6(b) also show that the training data adequately represent the repository set. The histogram representation of the repository set shown in Figs. 6(c) and 6(d) confirms the same.

TABLE III. C_f, *St*, and *RA* values for the structures A, B, and C. Also shown are the values normalized with the corresponding values for the flat channel, i.e., $C_{f,ref}$ = 0.06 and St_{ref} = 0.016.

Stru.	C_f	$\frac{C_f}{C_{f,ref}}$	St	$\frac{St}{St_{\text{ref}}}$	RA	$\frac{RA}{RA_{\text{ref}}}$
A	0.2606	4.3429	0.0276	1.7875	0.2113	0.3965
B	0.1715	2.8580	0.0185	1.1593	0.2163	0.4058
C	0.0782	1.3037	0.0208	1.3006	0.5321	0.9976

IV. DISCUSSIONS

A. Thermo-fluid aspect of the selected structures

In this section, we analyze in detail the pressure loss and heat transfer characteristics of the three *labeled* structures highlighted in Fig. 6(a) (namely, structures A, B, and C). The mean flow and temperature fields of these structures are shown in Fig. 7, and their C_f, *St*, and *RA* values are listed in Table III. Interestingly, these three structures resemble three canonical flow configurations encountered in fluid dynamics: vortex generator, converging–diverging nozzle, and backward-facing step. For the three considered structures, the percentage increase in surface area with respect to a flat channel is 25.1%, 5.4%, and 6.3%, respectively.

Among the three structures, the heat transfer achieved is maximum for structure A with the vortex-generating wing-like protrusion. Vortex generators of varying shapes, similar to the wing-like protrusion of structure A, are commonly used in heat exchanger devices to introduce unsteady swirling motions that can increase heat transfer.[33] Though there are two vortices inside the flow field as can be seen in Fig. 7(a), these vortices are part of the recirculation regions. Such regions in fact isolate the wall from the bulk of the fluid and are detrimental to effective heat transfer. Nonetheless, the developing thermal boundary layers generated on both the top and bottom walls [refer to the temperature field in Fig. 7(b)] result in an 80% increase in heat transfer. At the same time, the shape of structure A together with its increased surface area of 25% introduces significant pressure loss leading to a 60% reduction in *RA* when compared to the flat channel.

For structure B, the flow inside gets accelerated inside the converging section followed by fluid deceleration inside the diverging section. Unlike the case of structure A, a strong variation in the thermal boundary layer thickness is present only along the bottom wall. The presence of a recirculation region, covering the entire bottom wall except for the converging–diverging section, further limits heat transfer from the bottom wall. As a result, the increase in heat transfer is only 16%. The considerable pressure loss due to the absence of a well-streamlined converging–diverging section leads to a 60% reduction in *RA* with respect to the flat channel.

Structure C is unique owing to the fact that it contains the backward-facing step together with wall meandering. The structure results in a RA/RA_{ref} value approximately equal to unity but with a 30% increase in heat transfer with respect to the flat channel. This means that the structure results in a proportional increase in pressure loss as well. With the exception of the immediate area downstream of the backward step, the streamlines in the wall-normal direction exhibit behavior akin to those found within a rectilinear channel, maintaining a uniform and evenly spaced distribution. The absence of a developing thermal boundary layer indicates that wall meandering with a 6% increase in the surface area should be the

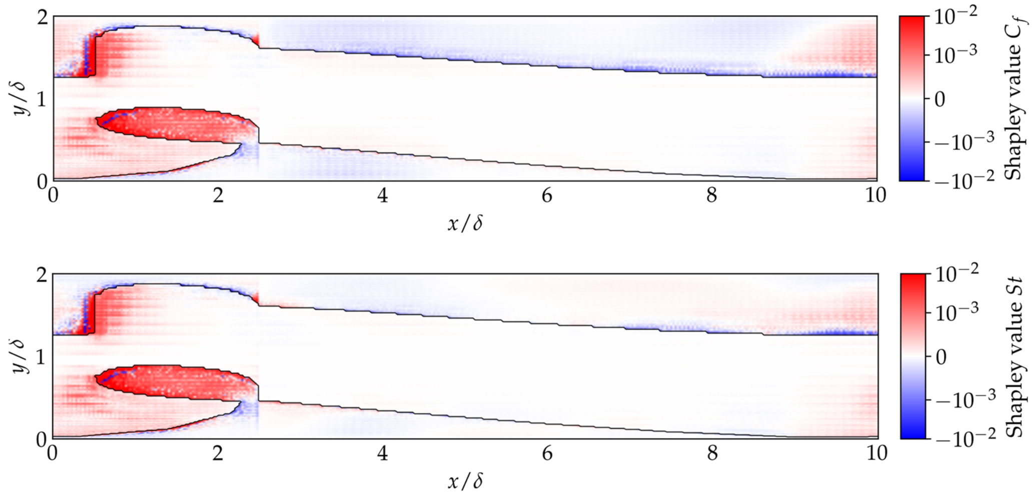


FIG. 8. Shap value for structure A: The top plot corresponds to the local distribution of the shap value for C_f and the bottom one corresponds to that for *St*. A nonlinear color bar is used for better visualization; however, a linear scale is applied between -10^{-3} and 10^{-3} to avoid having the plot go to infinity around zero.

primary reason for increased heat transfer. This suggests the possibility of achieving a value greater than unity for the RA/RA_{ref} value with carefully selected parameters for wall meandering.[34–36] Some of the structures in Figs. 6(a) and 6(b) also indicate this trend. Nevertheless, experimental validation is required to assess the practical efficacy of these geometries, and this constitutes one of our future research objectives.

B. Explainable artificial intelligence

Given a specific structured channel geometry, it is worth knowing which local aspects of this given geometry influence the CNN prediction. In this regard, we used Shapely Additive exPlanations (SHAP) developed by Lundberg and Lee.[37] The resulting Shapley value maps for structure A are shown in Fig. 8. Since our CNN model predicts two outputs, C_f and St, we have two plots that provide local approximations for these two outputs. The outline of structure A is superimposed on top of these plots to obtain a better understanding of the local distribution of the Shapley values.

The Shapley values generated using the DeepExplainer module within the SHAP Python package are computed using the CNN model alongside a distribution of background samples. Here, we are interested in understanding how C_f and St change with respect to that of a flat channel configuration. Hence, we used around 90 flat channels that are located symmetrically or asymmetrically about $y = \delta$ as background samples. It is important to note that the summation of the Shapley value is equal to the difference between the model prediction for the given structure and the mean prediction derived from the background samples.

The Shapley value plots depicted in Fig. 8 suggest that the CNN model predominantly emphasizes the detection of geometric characteristics, such as well-defined edges, corners, and recessed regions when making predictions—especially those geometric characteristics that are responsible for the appearance of flow features, such as boundary layer growth, flow separation, and recirculation regions, that influence the local drag and heat transfer. For structure A, the wing-like protrusion on the bottom wall and the sharp corner on the top wall are the predominant features responsible for the developing boundary layer and also for flow separation leading to the generation of recirculation regions. Since these features contribute to an increase in both C_f and St, the Shapely value plots for them also look similar. Thus, even though the CNN model has been trained on binary representations of structured channel geometries, the ML model has the ability to capture geometrical features of wall structuring related to the flow physics as indicated by the Shapely value distributions. The Shapley value distribution not only helps to understand what the ML model is focusing on, but it can also be used to optimize the shape of the structured channel to achieve a desirable outcome, for example, to increase heat transfer, which we will try to exploit in future work.

V. CONCLUSIONS AND OUTLOOK

We have trained a convolutional neural network (CNN) based on fluid flow and heat transfer simulations in two-dimensional laminar structured channel geometries. The trained CNN model accurately predicts the target properties, i.e., drag coefficient C_f and Stanton number St, for a structure geometry based on its binary representation. As a surrogate model, the CNN can predict the target properties at a fraction of the time (<100 ms per channel) required by numerical simulations (≈20 – 30 min per channel). The significant speedup provided by the surrogate model was then used for predicting C_f and St values of around 80 000 randomly generated geometries. This was done to find out geometries with normalized Reynolds analogy factor RA greater than unity, where $RA = 2St/C_f$ is an indicator of dissimilar heat transfer enhancement. However, our study suggests that to achieve such a goal, we have to exploit the speedup of the surrogate model by combining it with powerful optimization techniques.[32]

One of the highlights of the present CNN model is how we used domain-specific knowledge to augment the original training dataset. Recognizing that C_f and St remain invariant when a structure geometry is shifted along the flow direction or mirrored along the channel centerline, we populated the training dataset accordingly.

We showed that the training data generated using Bézier splines adequately represent structures that are commonly used in thermo-fluid applications. We then used Shapely Additive exPlanations as an explainable artificial intelligence tool to understand the CNN model. When applied to a structure, the Shapley values provide insights into the contributions of specific geometry features to the model's output in comparison to the background reference. In this regard, the Shapley values are not only valuable in understanding the ML models but can also be used to improve their performance.

The current CNN model is trained for a specific set of flow conditions, in terms of the Reynolds number and boundary conditions. As part of future work, we are interested in extending the current ML model to cover a wide range of these conditions. Additionally, in order to further exploit ML models in general and CNNs in particular for the design of chemical engineering unit operations, we plan to implement active learning approaches and generative models to reliably explore the possible design space of structured channel geometries and directly solve the inverse problem, i.e., the suggestion of channel structures given desired target properties. Moreover, trying to predict not just the values of C_f and St but also the velocity and temperature fields inside these arbitrary geometries, for example, using a physics-informed neural network[38] or graph neural network,[39] can be thought of as a natural extension of the present work.

The general approach presented here can be extended to more complex tasks, such as three-dimensional multiphase or even reactive unit operations in chemical engineering. The limitation will be the availability of data or the associated computational cost of the underlying simulations.

ACKNOWLEDGMENTS

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—FR 4072/3—within the Priority Programme "SPP 2331: Machine Learning in Chemical Engineering." The authors acknowledge the support received from the state of Baden-Württemberg through bwHPC. Parts of this work were performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

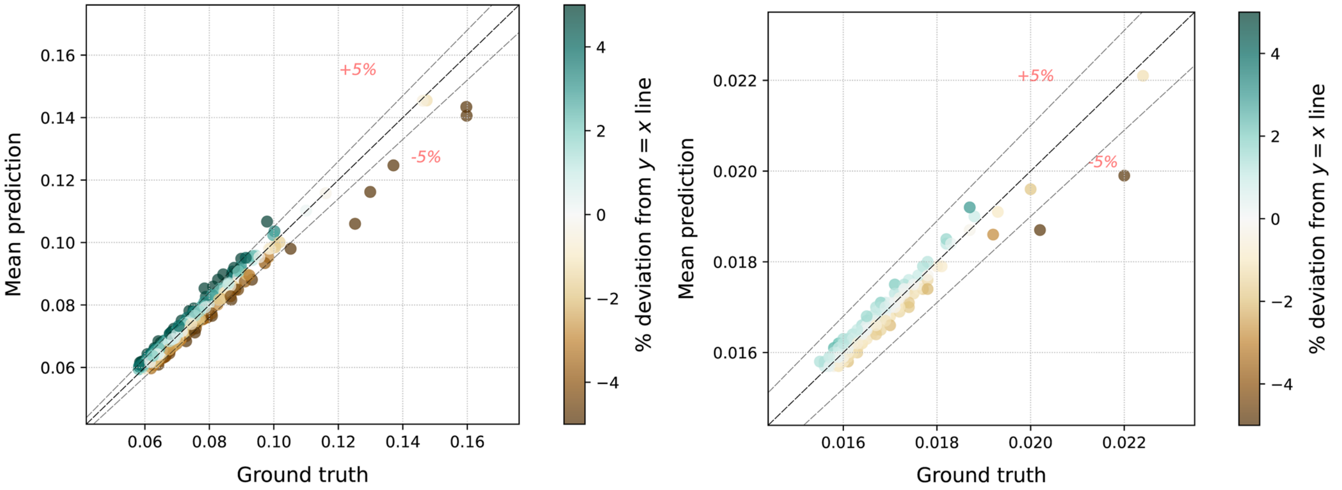


FIG. 9. Effect of shifting and flipping on the CNN prediction of C_f (*left*) and *St* (*right*) on the training data.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Y.K. and A.J.K. contributed equally to this paper.

A.S., P.F., and B.P.L. contributed to conceptualization, funding acquisition, and supervision. Y.K., A.J.K., and M.S. contributed to the investigation, software, and visualization. All authors contributed to the methodology, validation, and writing of the original as well as the final draft. Y.K. and A.J.K. contributed equally.

Yuri Koide: Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Arjun J. Kaithakkal**: Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Matthias Schniewind**: Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Bradley P. Ladewig**: Conceptualization (equal); Funding acquisition (equal); Methodology (equal); Supervision (equal); Validation (equal); Writing – original draft (equal). **Alexander Stroh**: Conceptualization (equal); Funding acquisition (equal); Methodology (equal); Supervision (equal); Validation (equal); Writing – original draft (equal). **Pascal Friederich**: Conceptualization (equal); Funding acquisition (equal); Methodology (equal); Supervision (equal); Validation (equal); Writing – original draft (equal).

DATA AVAILABILITY

The data that support the findings of this study are openly available in GitHub at https://github.com/aimat-lab/ChemEngML.[40]

APPENDIX: MODEL INVARIANCES TOWARD SHIFTING AND FLIPPING

As mentioned in Sec. II B, we investigated the effect of shifting and flipping of structures on the CNN prediction of C_f and *St*. Each of the test data points, excluding the flat channels (i.e., 459 data points), were examined as follows: For each test data point, 32 shifted and flipped copies were generated. Within this set, 16 structures were systematically produced by shifting the original structure along the x-direction in a total of 16 distinct increments within M_{x_s} and M_{x_e}. The remaining 16 structures were obtained by creating flipped images of the previously shifted structures. Subsequently, the (C_f, St) values of these 32 structures were averaged and compared with their respective ground truth values as shown in Fig. 9. The error in mean prediction is within ±5% for the majority of structures in the case of C_f and for nearly all structures in the case of *St*. Similarly, the percentage error in mean prediction for (C_f, St) for structures A, B, and C is within a reasonable range and is estimated to be (−13.87, 4.04), (−15.17, −0.53), and (12.91, 2.85), respectively.

REFERENCES

[1] A. Bejan, *Convection Heat Transfer* (John Wiley & Sons, 2013).

[2] W. Kays and A. London, *Compact Heat Exchangers* (McGraw-Hill, New York, 1984).

[3] R. Manglik and A. Bergles, "Heat transfer and pressure drop correlations for the rectangular offset strip fin compact heat exchanger," Exp. Therm. Fluid Sci. **10**(2), 171–180 (1995).

[4] M. Elyyan, A. Rozati, and D. Tafti, "Investigation of dimpled fins for heat transfer enhancement in compact heat exchangers," Int. J. Heat Mass Transfer **51**(11–12), 2950–2966 (2008).

[5] M. Fiebig, "Vortex generators for compact heat exchangers," J. Enhanced Heat Transfer **2**(1–2), 43 (1995).

[6] O. Reynolds, "On the extent and action of the heating surface of steam boilers," Proc. Lit. Soc. Manchester **14**, 7–12 (1874).

[7] P. Hassanzadeh, G. Chini, and C. Doering, "Wall to wall optimal transport," J. Fluid Mech. **751**, 627–662 (2014).

[8] K. Higashi, H. Mamori, and K. Fukagata, "Simultaneous control of friction drag reduction and heat transfer augmentation by traveling wave-like blowing/suction," Comput. Therm. Sci. **3**(6), 521–530 (2011).

[9] A. J. Kaithakkal, Y. Kametani, and Y. Hasegawa, "Dissimilarity between turbulent heat and momentum transfer induced by a streamwise travelling wave of wall blowing and suction," J. Fluid Mech. **886**, A29 (2020).

[10] A. J. Kaithakkal, Y. Kametani, and Y. Hasegawa, "Dissimilar heat transfer enhancement in a fully developed laminar channel flow subjected to a traveling wave-like wall blowing and suction," Int. J. Heat Mass Transfer **164**, 120485 (2021).

[11] S. Motoki, G. Kawahara, and M. Shimizu, "Optimal heat transfer enhancement in plane Couette flow," J. Fluid Mech. **835**, 1157–1198 (2017).

[12] Y. Hasegawa and N. Kasagi, "Dissimilar control of momentum and heat transfer in a fully developed turbulent channel flow," J. Fluid Mech. **683**, 57–93 (2011).

[13] A. Yamamoto, Y. Hasegawa, and N. Kasagi, "Optimal control of dissimilar heat and momentum transfer in a fully developed turbulent channel flow," J. Fluid Mech. **733**, 189 (2013).

[14] A. Stroh, K. Schäfer, P. Forooghi, and B. Frohnapfel, "Secondary flow and heat transfer in turbulent flow over streamwise ridges," Int. J. Heat Fluid Flow **81**, 108518 (2020).

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE **86**(11), 2278–2324 (1998).

[16] J. Viquerat and E. Hachem, "A supervised neural network for drag prediction of arbitrary 2D shapes in laminar flows at low Reynolds number," Comput. Fluids **210**, 104645 (2020).

[17] Y. Lee, H. Yang, and Z. Yin, "PIV-DCNN: Cascaded deep convolutional neural networks for particle image velocimetry," Exp. Fluids **58**, 171 (2017).

[18] K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," J. Fluid Mech. **870**, 106–120 (2019).

[19] J. Seo, H. Yoon, and M. Kim, "Establishment of CNN and encoder–decoder models for the prediction of characteristics of flow and heat transfer around NACA sections," Energies **15**(23), 9204 (2022).

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

[21] S. Leonardi, P. Orlandi, L. Djenidi, and R. A. Antonia, "Heat transfer in a turbulent channel flow with square bars or circular rods on one wall," J. Fluid Mech. **776**, 512–530 (2015).

[22] Y. Miyake, K. Tsujimoto, and M. Nakaji, "Direct numerical simulation of rough-wall heat transfer in a turbulent channel flow," Int. J. Heat Fluid Flow **22**(3), 237–244 (2001).

[23] Y. Nagano, H. Hattori, and T. Houra, "DNS of velocity and thermal fields in turbulent channel flow with transverse-rib roughness," Int. J. Heat Fluid Flow **25**(3), 393–403 (2004).

[24] M. Chevalier, P. Schlatter, A. Lundbladh, and D. Henningson, "Simson: A pseudo-spectral solver for incompressible boundary layer flows," TRITA-MEK, KTH Mechanics, Stockholm, Sweden, 2007.

[25] J. Bons, "A critical assessment of Reynolds analogy for turbine flows," J. Heat Transfer **127**(5), 472–485 (2005).

[26] The code was developed based on the StackOverflow answer provided at https://stackoverflow.com/a/50751932 for more details please refer to our GitHub code page 22.

[27] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, http://tensorflow.org/, 2015.

[28] F. Chollet *et al.*, Keras, https://keras.io, 2015.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv.1412.6980v9 (2017).

[30] P. Hayes, D. Anderson, B. Cheng, T. J. Spriggs, A. Johnson, and M. McCourt, "SigOpt documentation," Technical Report SO-12/14—Revision 1.07, SigOpt, Inc., 2019.

[31] R. Martinez-Cantin, K. Tee, and M. McCourt, "Practical Bayesian optimization in the presence of outliers," in *International Conference on Artificial Intelligence and Statistics* (PMLR, 2018), pp. 1722–1731.

[32] A. J. Kaithakkal, Y. Koide, M. Schniewind, P. Friederich, and A. Stroh, "Heat transfer enhancement in laminar channel flow by machine learning-guided shape optimization of wall geometry," in 17th International Heat Transfer Conference, 2023.

[33] M. Fiebig, "Embedded vortices in internal flow: Heat transfer and pressure loss enhancement," Int. J. Heat Fluid Flow **16**, 376–388 (1995).

[34] J. M. Floryan, "Flow in a meandering channel," J. Fluid Mech. **770**, 52–84 (2015).

[35] H. M. Metwally and R. M. Manglik, "Enhanced heat transfer due to curvature-induced lateral vortices in laminar flows in sinusoidal corrugated-plate channels," Int. J. Heat Mass Transfer **47**(10–11), 2283–2292 (2004).

[36] Z. G. Mills, A. Warey, and A. Alexeev, "Heat transfer enhancement and thermal–hydraulic performance in laminar flows through asymmetric wavy walled channels," Int. J. Heat Mass Transfer **97**, 450–460 (2016).

[37] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Curran Associates, Inc., 2017), Vol. 30.

[38] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686–707 (2019).

[39] J. Chen, E. Hachem, and J. Viquerat, "Graph neural networks for laminar flow prediction around random two-dimensional shapes," Phys. Fluids **33**(12), 123607 (2021).

[40] Y. Koide, A. J. Kaithakkal, M. Schniewind, A. Stroh, and P. Friederich. "aimat-lab/ChemEngML," GitHub. https://github.com/aimat-lab/ChemEngML