TOWARDS CONVOLUTIONAL NEURAL NETWORKS FOR HEAT EXCHANGERS IN ELECTRIFIED AIRCRAFT

G. Bokil, T.-F. Geyer, S. Wolff
DLR Institute of Electrified Aero Engines
Department of Control of Propulsion System, Cottbus, Germany

Abstract

Heat exchangers (HEX) are one of the most crucial components in the thermal management system of future electrified aircraft. To precisely model the convective heat transfer, high-fidelity Computational Fluid Dynamics (CFD) simulations are commonly carried out. However, due to their complexity, employing them in a design optimization loop is computationally expensive. This might lead to sub-optimal designs. One possible solution to solve this problem is to develop surrogate models to replace simulations with predictions. In recent trends, Convolutional Neural Networks (CNN) have shown large potential in modeling external aerodynamic flows. By employing this approach for heat exchangers, a geometry-adaptive U-net is developed to predict the velocity, pressure and temperature distribution of the air flow over various HEX fin configurations directly from geometry and boundary conditions. The model is trained on the steady state results obtained from solving unsteady Navier-Stokes equations using the open-source simulation toolkit Phiflow. The trained model is able to predict the flow fields for unseen fin configurations with an accuracy of 95 %. Moreover, it estimates the scalar pressure drop and temperature difference with an error of only 4 %. Due to the notably reduced computational cost compared to CFD simulations, the surrogate model can prove useful in performing rapid heat exchanger optimization to minimize pressure drop and maximize heat transfer.

Keywords

convolutional neural network; surrogate modeling; computational fluid dynamics; convective heat transfer

NOMENCLATURE

Symbols

α	angle of attack	rad
D	thermal diffusion coefficient	m^2/s
$d()$	distance function	
$\partial\Omega$	boundary	
$\cdot$	dot product	
δ	fin pitch	m
Δ	Laplacian operator	
l_f	length of the fins	m
$\min()$	minimum function	
∇	gradient operator	
n_f	number of fins	
ν	kinematic viscosity	m^2/s
Nx	number of grid points in X-direction	
Ny	number of grid points in Y-direction	
Ω	computational domain	
p	pressure	Pa
Pr	Prandtl number	
Re	Reynolds number	
ρ	density	kg/m^3
T	temperature	K
$\bar{u}$	velocity vector	m/s
w_f	width of the fins cross-section	m
X	length of the domain	
Y	width of the domain	

Abbreviations

BC	Boundary Condition
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
HEX	Heat Exchanger
RANS	Reynolds-Averaged Navier Stokes
SDF	Signed Distance Function
SLS	Semi-Lagrangian Scheme

doi: 10.25967/610419

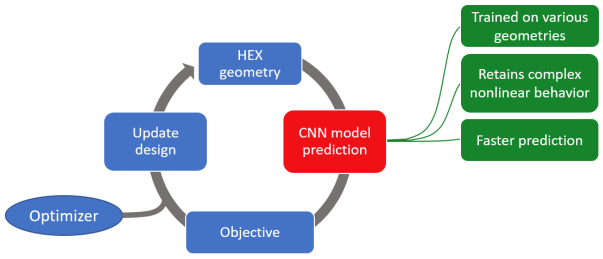


FIG 1. Surrogate model based optimization workflow

1. INTRODUCTION

Electric systems in every application are susceptible to high temperatures and require adequate cooling. Introducing electric drives as a safe propulsion system in an aircraft will thus require a powerful and efficient thermal management system. Predominantly, the heat generated by hydrogen fuel cells and batteries has to be effectively dissipated. Hence, the design and analysis of heat exchangers needs to be precise. Typical system architectures for electric propulsion involve the HEX to be placed in the nacelle downstream of the propellers (see, for example, [1]). A duct directs the incoming cold air towards the HEX and thus cools down the working fluid. Various HEX topologies such as the plate-fin, tube-fin, offset strip-fin and other types are available for this purpose. To simulate the convective heat flow, high-fidelity CFD simulations are commonly performed to obtain accurate field information. Additionally, during the design phase, 1D analytical models are used to compute the heat transfer, pressure drop, and overall performance of HEXs. After careful analysis and optimization, the final topology and dimensions are selected based on the system requirements. Although the final design can be indeed effective, it may not be optimal. That is because 1D models are far from reality. The effect of turbulence and thermal boundary layers on the performance cannot be easily approximated via 1D models. Thus, one needs an accurate and fast 2D/3D surrogate model in the optimization process. The surrogate models can then be included in the optimization process as shown in Fig 1.

Alongside established surrogate and reduced order modeling methods, CNNs have recently proved their capabilities in learning the latent geometry representations and spatial features. Ribeiro et al. [2] used an Autoencoder-type CNN architecture for image-to-image translation. They trained the network to predict the velocity and pressure distribution in a 2D fluid flow over arbitrary objects using the SDF representation of the computational domain and a boundary condition (BC) mask as inputs. Their model was trained on results from OpenFOAM [3] simulations using the SIMPLE algorithm [4]. Similarly, Thuerey et al. [5] trained a U-net [6] model to predict the velocity and pressure fields for flow over various airfoil shapes [7] using free stream BCs and a binary mask. Their network was trained on results obtained from RANS simulations with the Spalart-Allmaras [8] turbulence closure model. A different approach was used by Guo et al. [9], who generated simulation data of flow over 100,000 random shapes using the 2D and 3D Lattice-Boltzmann-Method (LBM). A common-encoder separate-decoder model was trained on this data to predict the velocity components from the respective SDF representation of the geometry. They also compared the effect of using SDF representation and a binary representation of the geometry as an input on the model performance. According to their conclusions, the CNN model with SDF as an input was more generalizable than a simple binary mask as it provided more geometric and boundary information. Portal-Porras et al. [10] used the CNN approach to predict the flow field quantities and the scalar aerodynamic coefficients for different flow control devices on an airfoil
Most of the literature on deep learning methods as surrogate models for flow prediction deal with external aerodynamics. In addition to the above mentioned literature, the capabilities of different CNN architectures to predict the flow over arbitrary geometries, vehicle shapes and airfoils from geometric data have also been recently studied [11–14]. Compared to reduced order modeling methods like the Proper Orthogonal Decomposition (POD), the CNN models have been observed to perform as good as POD with 70 basic modes [15]. Recently, these approaches are being adapted for convective flow applications. Seo et al. [16] trained a CNN encoder-decoder model to predict the velocity, pressure and temperature flow fields for flow over various NACA airfoils. Wang et al. [17] performed a comparison study between the conventional Radial Basis Function (RBF) regression, Gaussian Processes Regression (GPR), and dense Artificial Neural Networks (ANN) with CNN architectures for predicting the convective heat transfer in a U-bend channel. Their observations revealed that the novel CNN-based architectures perform better. As opposed to flow field prediction, Keramati and Hamdullahpur [18] developed a CNN surrogate to predict the scalar heat transfer from geometry images for 2D flow over arbitrary shapes. However, in the case of HEX design and optimization, the use of CNN surrogate models is yet to be extensively explored. Consequently, this research presents a preliminary study of the capability of a geometry-adaptive CNN surrogate model in learning the convective flow over heat exchangers. It should be noted that accurate physical modeling was not the primary focus of this study.

2. PROBLEM DEFINITION

For investigating the CNN-based approach, an air-cooled plate-fin heat exchanger is selected in this study. A 2D plane between the fins is considered for simulation. Fig 2 displays the 2D computational domain and the HEX geometry parameters.
The domain length was set to $L = 2$ m and height $W = 1$ m. To simulate the air flow over fins in the channel between the plates, the governing equations to be solved in the domain (Ω) are the unsteady incompressible Navier-Stokes equations in Eq. (1) with the energy equation in Eq. (2) and the continuity equation in Eq. (3) as

$$\frac{\partial \bar{u}}{\partial t} + (\bar{u} \cdot \nabla)\bar{u} = -\frac{1}{\rho}\nabla p + \nu \Delta \bar{u} \quad \text{in } \Omega, \tag{1}$$

$$\frac{\partial T}{\partial t} + (\bar{u} \cdot \nabla)T = D\Delta T \quad \text{in } \Omega, \tag{2}$$

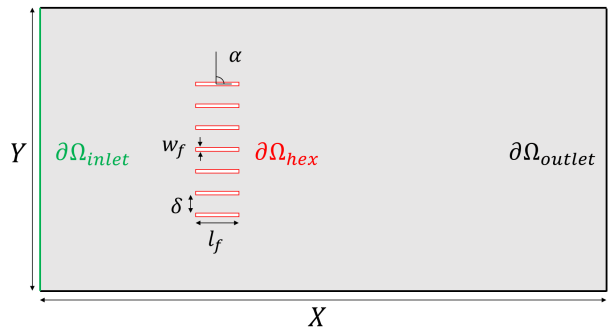


FIG 2. Problem definition

Parameter	Lower bound	Upper bound
n_f	2	8
l_f	$\frac{10}{Ny}$	$\frac{20}{Ny}$
δ	$\frac{4}{Ny}$	$\frac{14}{Ny}$
α	$-\frac{\pi}{4}$	$\frac{\pi}{4}$

TAB 1. Range of geometry parameters

FIG 3. Layers in the U-net

$$\nabla \cdot \bar{u} = \bar{0} \quad \text{in } \Omega. \tag{3}$$

In these equations, p is the pressure, $\bar{u}$ is the velocity vector, ρ is the density of the fluid and D is the thermal diffusivity. The constants in the simulation are $\nu = 1e-4$, $Pr = 1$, $Re \approx 550$, and

$$D = \frac{1}{Re\ Pr}, \tag{4}$$

where ν is the kinematic viscosity, Pr is the Prandtl number and Re is the Reynolds number. The Reynolds number changes as the length of the fin is changed for generating different geometries. The governing equations were solved using the open-source differentiable solver Phiflow [19]. The following boundary conditions were applied:

$$\bar{u} = (0.5, 0.0) \quad \text{on } \partial\Omega_{inlet}, \tag{5}$$

$$\bar{u} = (0.0, 0.0) \quad \text{on } \partial\Omega_{hex}, \tag{6}$$

$$T = 220K \quad \text{on } \partial\Omega_{inlet}, \tag{7}$$

$$T = 420K \quad \text{on } \partial\Omega_{hex} \text{ and} \tag{8}$$

$$\frac{\partial P}{\partial x} = \frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = \frac{\partial T}{\partial x} = 0 \quad \text{on } \partial\Omega_{outlet}. \tag{9}$$

The computational domain was discretized in a uniform structured grid of resolution (Nx, Ny), where $Nx = 256$ and $Ny = 128$. The advection terms in Eq. (1) and Eq. (2) were treated using the Semi-Lagrangian Scheme (SLS) [20]. The diffusion terms in Eq. (1) and Eq. (2) were treated using an explicit scheme with four sub-steps. Finally, the incompressibility constraint in Eq. (3) was applied using the Chorin-Temam projection [21] to compute pressure.

3. DATA GENERATION

To train a CNN model, a training dataset needs to be generated. In the current study, the fin geometry was parameterized into number of fins (n_f), length of the fins (l_f), fin pitch (δ), and angle of attack (α) as shown in Fig 2. The width of the fins (w_f) was kept constant. To generate various geometries, these variable design parameters were randomly sampled from chosen respective ranges as shown in Tab. 1.

By randomly sampling the geometry parameters, 850 different HEX configurations were created. Transient simulations were carried out for all the geometries. The velocity and temperature boundary condition fields were saved at the simulation start. Along with the BC mask, the SDF field was also generated and saved. The SDF is defined as the shortest distance of each point x in the domain from the obstacle boundary according to

$$f_{sdf} = \min(d(x, \partial\Omega)) \quad \text{for } x \in \Omega. \tag{10}$$

Simulations were assumed to have reached steady-state when the time-derivatives were below the tolerance of 1e-5. Then, the velocity components (U_x, U_y), pressure (p) and temperature (T) fields were saved. For better training performance, the SDF field was normalized between 0 and 1. Since the pressure is computed up to a constant for incompressible flows, the actual value of the pressure has little importance. The pressure drop is more important. Thus, the pressure values are also normalized for training and given as P.

4. SURROGATE MODELING

To develop a CNN model, the U-net [6] architecture was chosen. U-nets are popularly used in image segmentation and are known to be able to extract complex features. Moreover, the "skip connections" allow the flow of high-level and low-level information in the encoding and decoding parts of the model to be connected and carried forward. The U-net architecture was constructed using a [3,3] filter size for convolution/de-convolution layers and Leaky-ReLU activation layer. By stacking these layers in an alternate fashion with max pooling and upsampling layers of [2,2] filter size at the last, encoding and decoding blocks were created respectively. The compression and expansion consisted of [16,32,64,128,64,32,16] number of filters each.

The encoding blocks compress the inputs to a latent space and decoding blocks expand these latent features into flow fields. The skip connections basically concatenate the outputs of the decoding block and a corresponding encoding block and send them to the next decoding block. A single U-net model was created using the above blocks, as shown

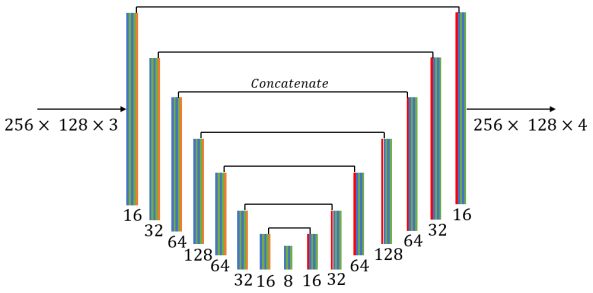


FIG 4. U-net architecture

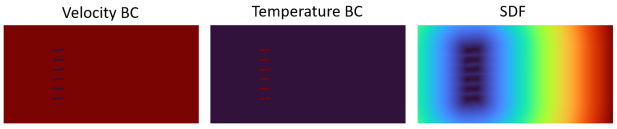


FIG 5. U-net model inputs

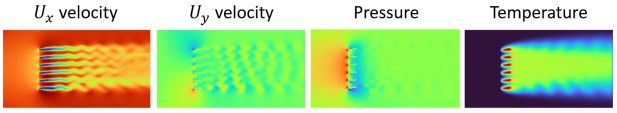


FIG 6. U-net model outputs

FIG 7. Training history for the U-net model

in Fig 4. The U-net received three inputs namely velocity BC, temperature BC and the SDF representation at 256 ×128 resolution, as shown in Fig 5. The network generates four outputs namely x- and y-components of the velocity as well as the pressure and temperature at the same resolution shown in Fig 6. The model was built using the TensorFlow 2.11.0 library [22] in Python 3.7.4. Out of a total of 850 samples, 50 samples were separated for testing. The model contained approximately 1.8 million trainable parameters and was trained for 200 epochs using the Adam [23] optimizer. The $L1$-norm was chosen as the loss function as it is more robust against outliers. A validation split of 0.1, an initial learning rate of 1e-3 and a batch size of 10 was used during training. The learning rate was reduced as a function of training loss using the ReduceLROnPlateau scheduling.

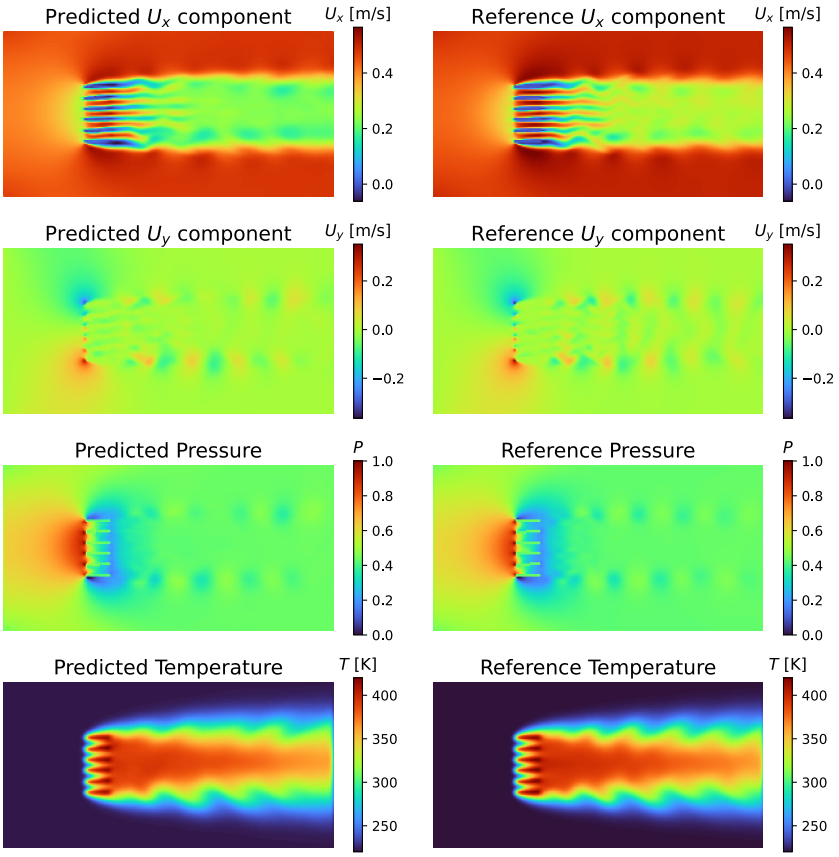
5. RESULTS

At the end of training, the model achieved a total Mean Squared Error (MSE) of 4e-6 on the training data and 4e-4 on the validation data as shown in Fig 7. On the unseen 50 test samples, individually, the MSE was 2e-3 for x- and y-components of the velocity, 1e-3 for pressure and 2e-3 for temperature. To qualitatively measure the accuracy of the predictions, the Symmetric Mean Absolute Percentage Error (SMAPE) was selected as an appropriate criterion as it does not suffer from exploding error for zero magnitudes like the relative percentage error. It is defined as

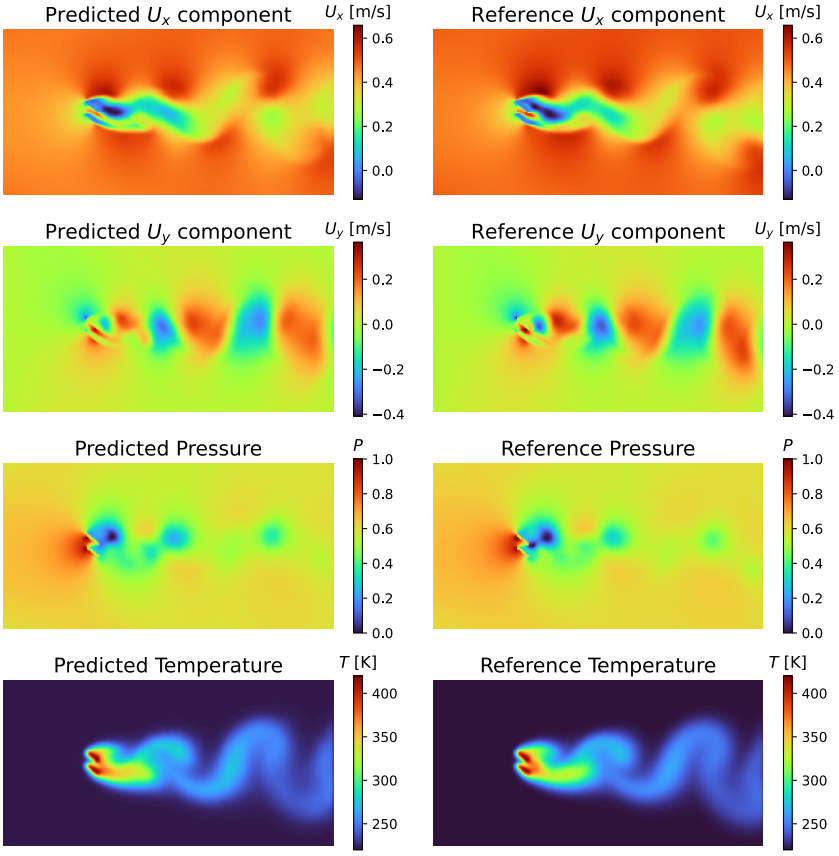
$$SMAPE = 100 \frac{\sum_{i=1}^{n} |x_i - y_i|}{\sum_{i=1}^{n} x_i + y_i} \%. \tag{11}$$

The overall mean SMAPE was observed to be 2 % on the training set and 4 % on the test set. Individually, the SMAPE was 1 % for the x-component of the velocity, 1 % for the y-component of the velocity, 2 % for pressure and 4 % for temperature on the training set. Similarly, the SMAPE was 2 % for the x-component of the velocity, 3 % for the y-component of the velocity, 5 % for pressure and 8 % for temperature on the test set. The proposed U-net architecture was able to predict the flow fields with only 1.1 million parameters and 800 training samples. To achieve the same accuracy on the same grid resolution in a similar application, the U-net model by Matthias et al. [12] required 53 million parameters, the U-net by Thuerey et al. [5] required 2-30 million parameters and the common-encoder separate decoder model by Guo et al. [9] required more than 10 million parameters. From this preliminary study it can thus be inferred that the proposed U-net architecture would also require less parameters for higher resolutions and more complex flows.

Fig 8 shows the comparison between the model prediction and the reference CFD result for four different test geometries. At the first glance, the CNN prediction matches well with the CFD results and is visibly indistinguishable. It can be observed in Fig 8a that the model predicts the average U_x velocity flow correctly along with the oscillations occurring in the wake of the fins. Similarly, minor pressure fluctuations and periodic U_y oscillations are predicted with negligible errors. Due to less fluctuations in temperature, the predictions were even more accurate. For the configuration in Fig 8b, two fins with certain angle of attack are placed. As a result, the fins act like a bluff body and lead to the von Kármán vortex shedding phenomenon. In this case, the model was able to correctly predict the periodic nature of the flow quantities and flow separation at the fin boundaries. Considering more fins with different orientations shown in Fig 8c and Fig 8d, the predicted pressure and temperature fields are in agreement with the CFD results. In Fig 8d, the large eddies occurring due to the pressure differences in the wake were predicted with a slight artificial averaging or diffusion. Considering the model was trained on a very few geometries, it performs quite well on unseen geometries. However, it is susceptible to non-physical predictions and minor averaging phenomena in case of vortices. More examples of the comparison between the prediction and simulation can be found in the Appendix. When using

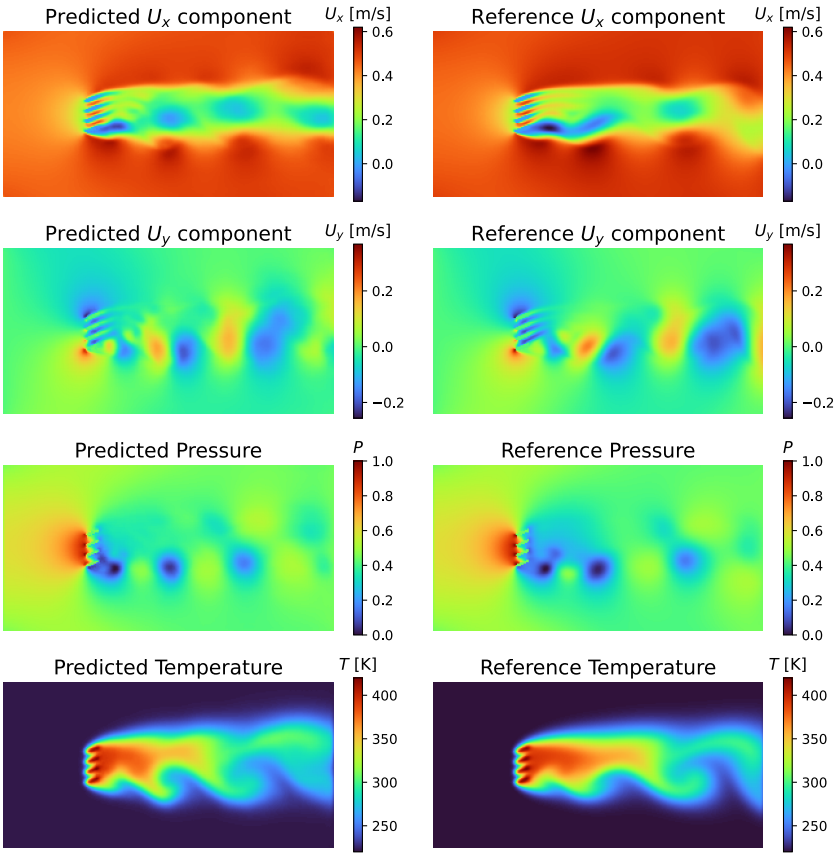


(a) Parameters: $n_f = 6, l_f = 0.123, \delta = 0.045, \alpha = 5\,\mathrm{deg}$

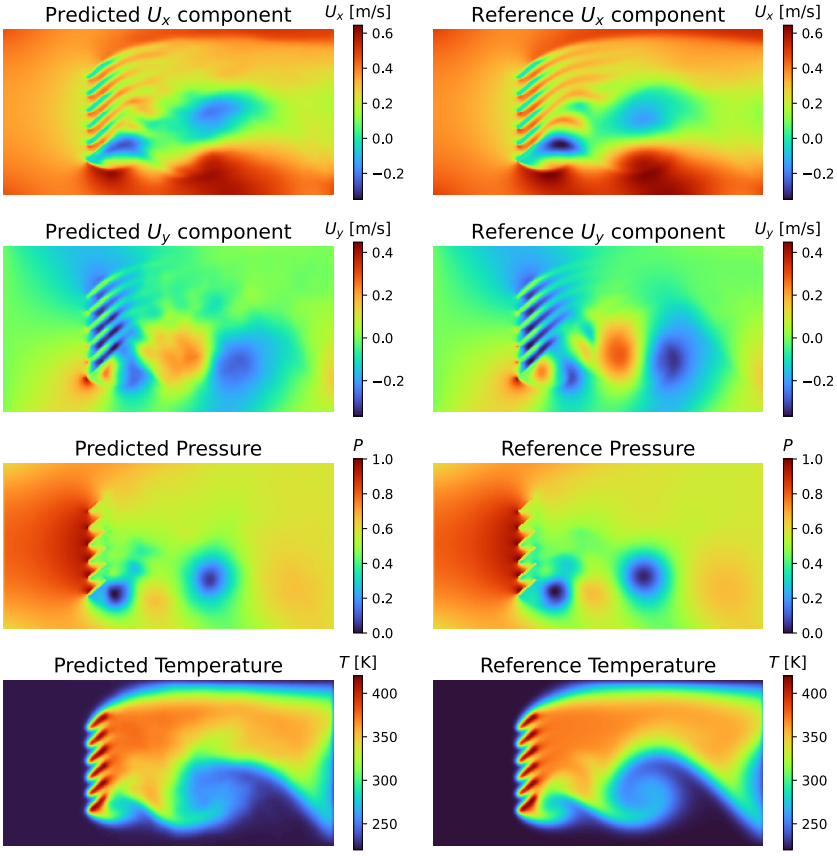


(b) Parameters: $n_f = 2, l_f = 0.1, \delta = 0.05, \alpha = -35\,\mathrm{deg}$

FIG 8. Comparison between CNN prediction and CFD results (1)



(c) Parameters: $n_f = 4, l_f = 0.99, \delta = 0.047, \alpha = 21\,\mathrm{deg}$



(d) Parameters: $n_f = 6, l_f = 0.137, \delta = 0.086, \alpha = 45\,\mathrm{deg}$

FIG 8. Comparison between CNN prediction and CFD results (2)

this surrogate model for optimization, the objective function will depend on the scalar characteristics like the temperature difference and pressure drop. To further inspect the model accuracy, the temperature difference $\Delta T = T_{outlet} - T_{inlet}$ and the pressure drop $\Delta P = P_{hex} - P_{outlet}$ was computed from the predictions and compared with the CFD results in Fig 9.

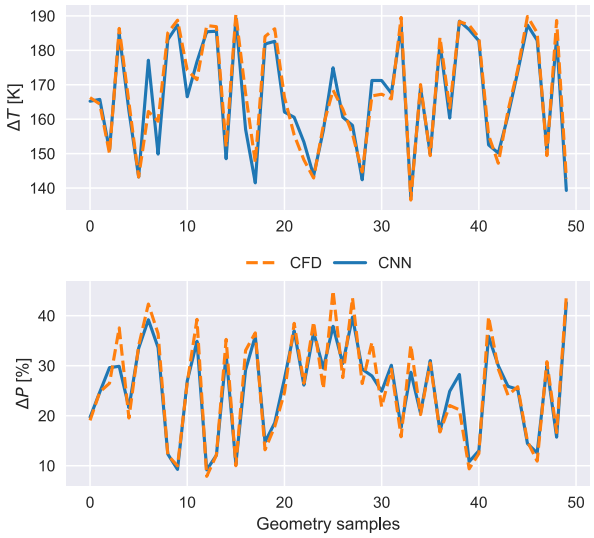


FIG 9. Comparison between CNN and CFD for ΔT and ΔP

On the test geometries, the temperature difference was mostly accurate with a Mean Absolute Percentage Error (MAPE) of 5 %. In case of the pressure drop, the error was approximately 8 %.

6. CONCLUSION AND OUTLOOK

Simulating flow over heat exchangers is challenging due to the combination of turbulence and heat transfer. Expensive CFD simulations are thus not suitable for geometry optimization. In recent trends, replacing CFD with deep learning models is being explored. The goal of this research was to test the capability of Convolutional Neural Networks (CNNs) in approximating the HEX simulation results. To this end, a geometry-adaptive CNN model was developed in the current study, which was able to predict the flow fields, namely the x- and y-components of the velocity, the pressure and temperature, directly from a Signed Distance Function (SDF) representation and boundary conditions (BC) on the domain. The trained model performed with an accuracy of 95 % on unseen test geometries. Additionally, due to the instantaneous prediction, the computational speed-up compared to the CFD simulation was roughly 1000 times. Since the model was trained purely on data, the predictions are not necessarily compliant with the Navier-Stokes equations. Hence, the predictions may appear averaged or diffused at the locations of vortices. In general, the CNN surrogate modeling approach displays high potential towards learning convective flow over heat exchangers.

Efforts are currently being made to circumvent the above mentioned drawback of this model. The immediate goal is to apply physics-informed constraints on the CNN model. Enforcing the governing equations and boundary conditions will ensure that the predictions are physical and not mere approximations. Thus, the residual of the Navier-Stokes equations and incompressibility condition would be added to the loss during training. This is challenging especially for CNN-based approaches since the gradients of the field cannot be as easily computed as in the case of Physics-Informed Neural Networks (PINNs). To capture the multi-scale effects, simulations with finer grid resolution and methods like LES and DNS are sought. Deeper networks for better feature extraction need to be tested. It is expected that by this approach the model will be able to predict the intricate turbulent structures in the flow. Other CNN architectures like the Generative Adversarial Network (GAN), Echo State Network (ESN) and Variational Autoencoders (VAE) will also be investigated. Since the proposed model is able to predict the scalar pressure drop and temperature difference, the future work will also employ this model for geometry optimization to maximize heat transfer, minimize pressure drop and thus improve the overall efficiency of heat exchangers in an electric aircraft.

Acknowledgment

The authors are grateful for the insightful discussions with Sahil Bhapkar and Chetan Kumar Sain (DLR) about heat exchanger design and simulation strategies in the DLR-project *AHEAD*.

Contact address:

gaurav.bokil@dlr.de

References

[1] Chetan K Sain, Jeffrey Hänsel, and Stefan Kazula. Conceptual design of air and thermal management in a nacelle-integrated fuel cell system for an electric regional aircraft. In *AIAA AVIATION 2023 Forum*, 2023. AIAA paper 2023-3875. DOI: 10.2514/6.2023-3875.

[2] Mateus Dias Ribeiro, Abdul Rehman, Sheraz Ahmed, and Andreas Dengel. Deepcfd: Efficient steady-state laminar flow approximation with deep convolutional neural networks. *arXiv preprint arXiv:2004.08826*, 2020. DOI: 10.48550/arXiv.2004.08826.

[3] Christopher Greenshields. *OpenFOAM v10 User Guide*. The OpenFOAM Foundation, London, UK, 2022.

[4] Suhas V. Patankar. *Numerical Heat Transfer and Fluid Flow*. Electro Skills Series. Hemisphere Publishing Corporation, 1980. ISBN: 9780070487406. DOI: 10.1007/978-981-13-1903-7.

[5] Nils Thuerey, Konstantin Weißenow, Lukas Prantl, and Xiangyu Hu. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 58(1):25–36, jan 2020. DOI: 10.2514/1.j058291.

[6] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical im-

age segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. DOI: 10.1007/978-3-319-24574-4$_2$8.

[7] Michael S. Selig. *UIUC airfoil data site*. Urbana, Ill. : Department of Aeronautical and Astronautical Engineering University of Illinois at Urbana-Champaign, 1996-, 1996. Title from title display (viewed on Nov. 5, 1999);Mode of access: World Wide Web.

[8] Philippe Spalart and Steven Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA*, 439, 01 1992. DOI: 10.2514/6.1992-439.

[9] Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 481–490, New York, NY, USA, 2016. Association for Computing Machinery. DOI: 10.1145/2939672.2939738.

[10] Koldo Portal-Porras, Unai Fernandez-Gamiz, Ekaitz Zulueta, Alejandro Ballesteros-Coll, and Asier Zulueta. CNN-based flow control device modelling on aerodynamic airfoils. *Scientific Reports*, 12(1), May 2022. DOI: 10.1038/s41598-022-12157-w.

[11] Saakaar Bhatnagar, Yaser Afshar, Shaowu Pan, Karthik Duraisamy, and Shailendra Kaushik. Prediction of aerodynamic flow fields using convolutional neural networks. *Computational Mechanics*, 64(2):525–545, jun 2019. DOI: 10.1007/s00466-019-01740-0.

[12] Matthias Eichinger, Alexander Heinlein, and Axel Klawonn. Surrogate convolutional neural network models for steady computational fluid dynamics simulations. *ETNA - Electronic Transactions on Numerical Analysis*, 56:235–255, 2022. DOI: 10.1553/etna$_v$$ol56s$235.

[13] Cihat Duru, Hande Alemdar, and Özgür Uğraş Baran. CNNFOIL: Convolutional encoder decoder modeling for pressure fields around airfoils. *Neural Computing and Applications*, 33(12):6835–6849, November 2020. DOI: 10.1007/s00521-020-05461-x.

[14] Satyadhyan Chickerur and P Ashish. A convolutional neural network based approach for computational fluid dynamics. In *2021 Second International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, pages 1–5, 2021. DOI: 10.1109/ICSTCEE54422.2021.9708548.

[15] Ming-Yu Wu, Yan Wu, Xin-Yi Yuan, Zhi-Hua Chen, Wei-Tao Wu, and Nadine Aubry. Fast prediction of flow field around airfoils based on deep convolutional neural network. *Applied Sciences*, 12(23), 2022. DOI: 10.3390/app122312075.

[16] Janghoon Seo, Hyun-Sik Yoon, and Min-Il Kim. Establishment of CNN and encoder-decoder models for the prediction of characteristics of flow and heat transfer around naca sections. *Energies*, 15(23), 2022. DOI: 10.3390/en15239204.

[17] Qi Wang, Weiwei Zhou, Li Yang, and Kang Huang. Comparison between conventional and deep learning-based surrogate models in predicting convective heat transfer performance of u-bend channels. *Energy and AI*, 8:100140, 2022. DOI: https://doi.org/10.1016/j.egyai.2022.100140.

[18] Hadi Keramati and Feridun Hamdullahpur. Deep convolutional surrogates and freedom in thermal design. *Energy and AI*, 13:100248, 2023. DOI: https://doi.org/10.1016/j.egyai.2023.100248.

[19] Philipp Holl, Vladlen Koltun, and Nils Thuerey. Learning to control PDEs with differentiable physics. *CoRR*, abs/2001.07457, 2020.

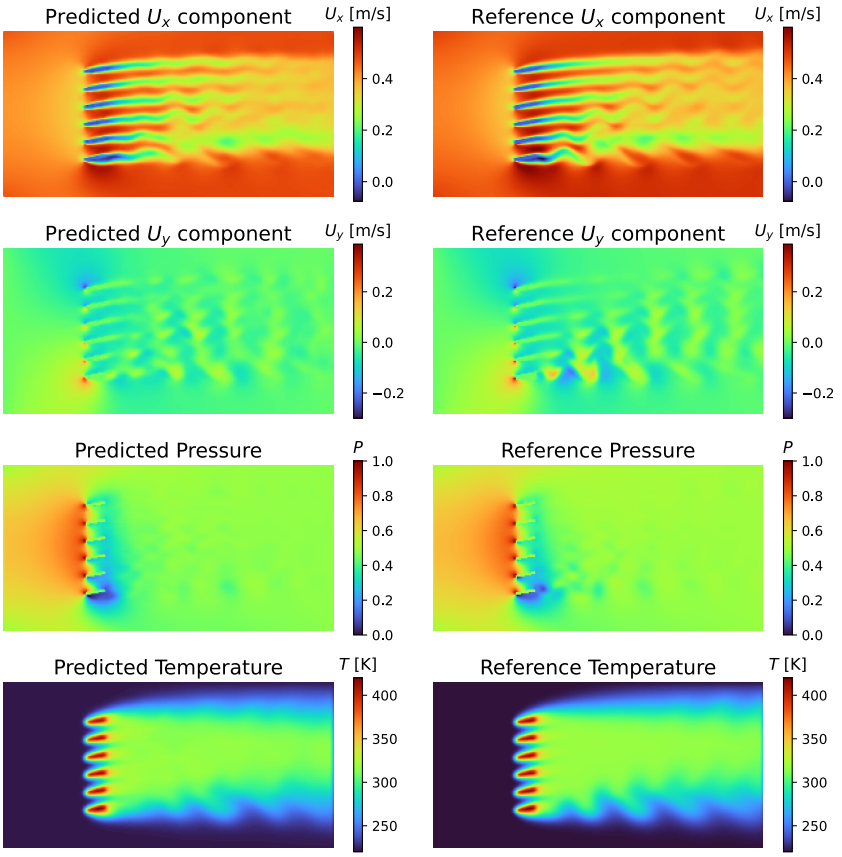
[20] Jos Stam. Stable fluids. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 121–128, USA, 1999. ACM Press/Addison-Wesley Publishing Co. DOI: 10.1145/311535.311548.

[21] Alexandre Joel Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 2(1):12–26, 1967. DOI: https://doi.org/10.1016/0021-9991(67)90037-X.

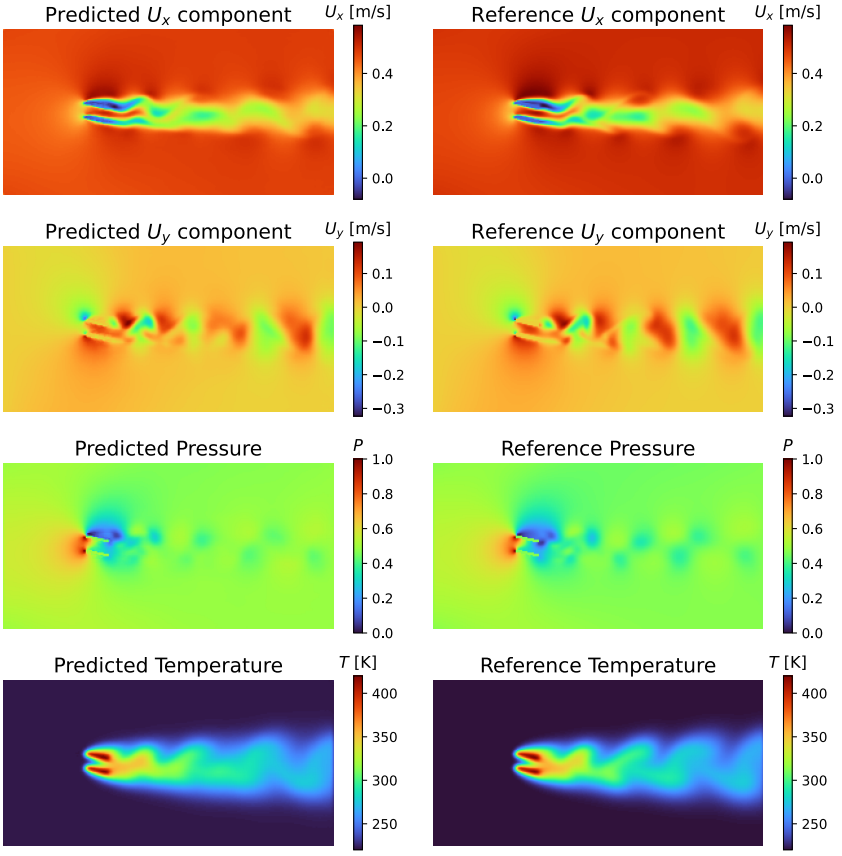
[22] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. DOI: 10.48550/arXiv.1412.6980.

APPENDIX

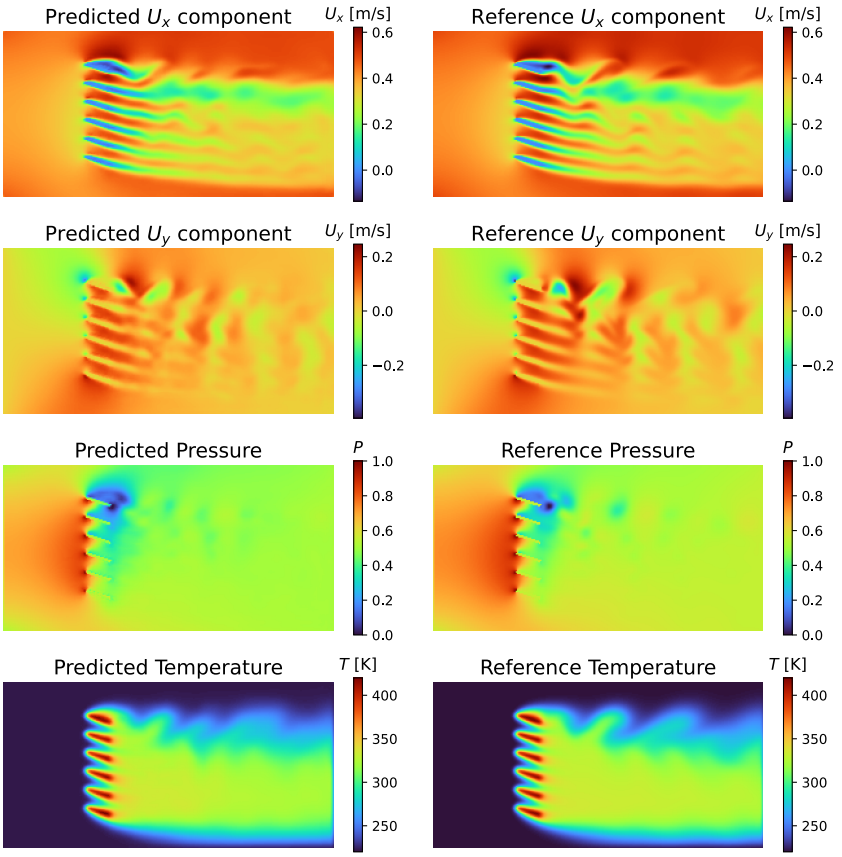


(a) Parameters: $n_f = 6, l_f = 0.3, \delta = 0.088, \alpha = 9\,\mathrm{deg}$

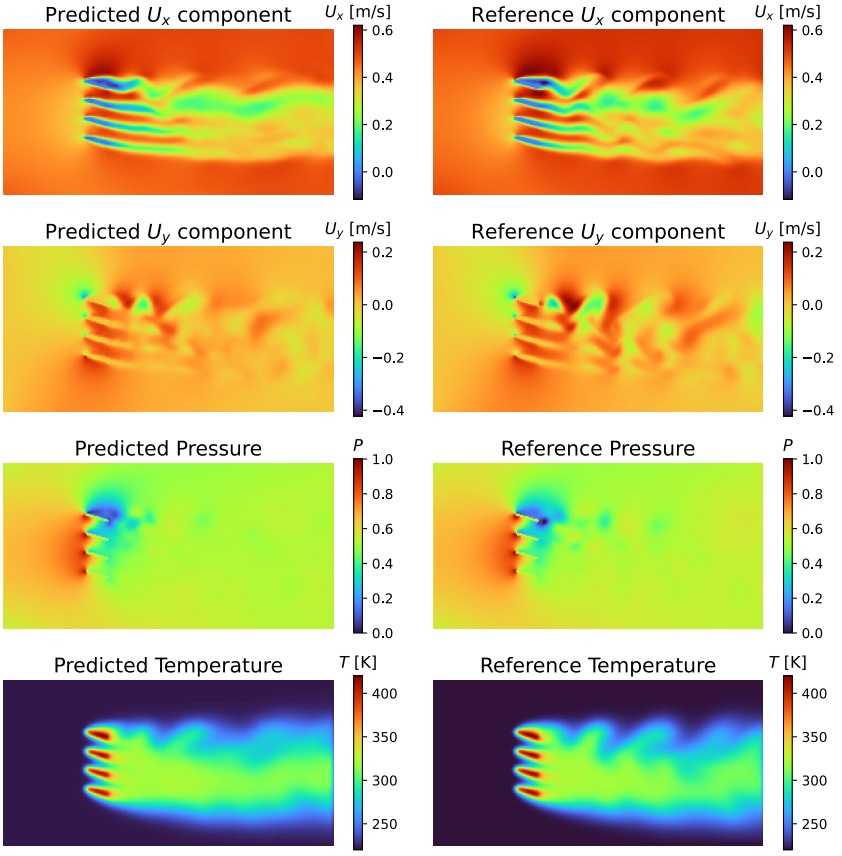


(b) Parameters: $n_f = 2, l_f = 0.11, \delta = 0.074, \alpha = -17\,\mathrm{deg}$

FIG 10. Additional comparison between CNN prediction and CFD results (1)



(c) Parameters: $n_f = 6, l_f = 0.13, \delta = 0.096, \alpha = -10\,\mathrm{deg}$



(d) Parameters: $n_f = 4, l_f = 0.119, \delta = 0.077, \alpha = -18\,\mathrm{deg}$

FIG 10. Additional comparison between CNN prediction and CFD results (2)