Deep Learning Surrogates for Computational Fluid Dynamics

Liwei Chen, Berkay Cakal, Xiangyu Hu, Nils Thuerey

Technical University of Munich

Abstract

We investigate the accuracy of surrogate deep learning models for the direct inference of Reynolds-Averaged Navier-Stokes solutions and in the context of shape optimization problems. With our best models we arrive at a mean relative pressure and velocity error of less than 3% across a range of previously unseen airfoil shapes. In addition, we show that these learned models can be employed as surrogates to carry out shape optimization problems, i.e. to find a drag minimal profile with a fixed cross-section area subjected to a two-dimensional steady laminar flow. Due to its combination of generality in conjunction with the fast runtime, our deep learning-based optimization framework shows promise for general aerodynamic design problems.

1 Introduction and Overview

Deep learning methods have achieved huge successes in the field of computer vision [9, 4, 5], and there are first success stories for applications in the area of physics simulations [18, 20, 2, 13, 1]. Our goal is to investigate the accuracy and flexibility of trained deep learning models for the inference of Reynolds-averaged Navier-Stokes (RANS) simulations of airfoils in two dimensions. RANS simulations are time-averaged and provide an important building block for practical fluid problems. As such play an important role in many applications and disciplines. We demonstrate that the trained models yield a very high computational performance, and can be used for challenging shape optimization problems. As learning task we focus on the direct inference of RANS solutions from a given choice of boundary conditions, i.e., airfoil shape and freestream velocity. The specification of the boundary conditions as well as the solution of the flow problems will be represented by Eulerian field functions, i.e. Cartesian grids. For the solution we typically consider velocity and pressure distributions. Deep learning as a tool makes sense in this setting, as the functions we are interested in, i.e. velocity and pressure, are smooth and well represented on Cartesian grids. Also, convolutional layers, as a particularly powerful component of current deep learning methods, are especially well suited for such grids. The setup we describe in the following is a very generic approach for PDE boundary value problems, and as such is applicable to a variety of other equations beyond RANS.

We demonstrate this flexibility below by targeting the inference solution, in addition to shape optimization problems for fluid flow. To understand the mechanisms underlying drag reduction and to develop optimization algorithms, a large amount of analytical and computational work has been performed [11, 3, 6, 8]. Pironneau et al. [11] already analysed the minimum drag shape for a given volume in Stokes flow, and later for the Navier-Stokes equations [12]. By using the adjoint variable approach, Kim et al. [7] investigated the minimal drag profile for a fixed cross-section area in the two-dimensional laminar flow. More recently Katamine et al. [6] studied the same problem at different Reynolds numbers. Although the laminar flow regimes are well studied, due to the separation and nonlinear nature of the fluid, it can be challenging for surrogate models to predict the

1st NeurIPS workshop on Interpretable Inductive Biases and Physically Structured Learning (2020), virtual.

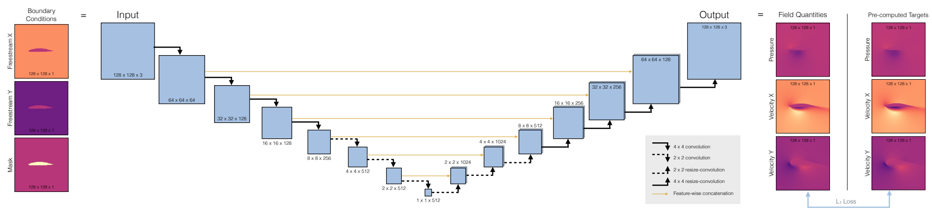


Figure 1: Our U-net architecture receives three constant fields as input, each containing the airfoil shape. The black arrows denote convolutional layers. while orange arrows indicate skip connections. The inferred outputs have exactly the same size as the inputs, and are compared to the targets with an L_1 loss. The target data sets on the right are pre-computed with OpenFOAM.

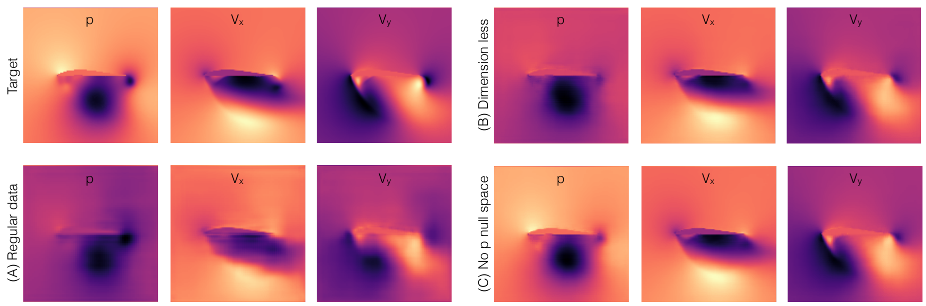


Figure 2: Ground truth target, and three different data pre-processing variants. It is clear that using the data directly (A) leads to noteable artifacts with blurred and jagged solutions. While velocity normalization (B) yields significantly better results, the pressure values still show large deviations from the target. These are reduced by removing the pressure null space from the data (C).

drag-minimal shape as well as aerodynamic forces. To our knowledge, no previous studies exists that investigate this topic and quantitatively assess the results in the context of deep learning surrogates.

Below, we use trained deep neural network for RANS flow inference as a solver in the shape optimization. In comparison to conventional surrogate models [21] and other optimization work involving deep learning [10, 14, 19], we make use of a generic model that infers flow solutions: in our case it produces fluid pressure and velocity as field quantities. I.e., given encoded boundary conditions and shape, the DNN surrogate produces a flowfield solution, from which the aerodynamic forces are calculated. Thus, both the flowfield and aerodynamic forces can be obtained during the optimization.

2 Fluid Flow Regression with Neural Networks

We consider problems of the form $\mathbf{y} = \hat{f}(\mathbf{x}; \Theta)$, i.e., for given an input $\mathbf{x}$ we want to approximate the output $\mathbf{y}$ of the true function $\hat{f}$ as closely as possible with a representation f based on the weights Θ such that $\mathbf{y} \approx f(\mathbf{x}; \Theta)$. Our neural network model for f is based on the U-Net architecture [15], and uses 7 blocks in the encoder section and another 7 in the decoder section of the network, each with two convolutional layers, a batch normalization, and ReLU activation functions. An illustration of the architecture is shown in Fig. 1.

The RANS simulations make use of the widely used Spalart-Allmaras [17] one equation turbulence model. We resample the region around the airfoils with a Cartesian 128^2 grid to obtain the ground truth pressure and velocity data sets. In order to generate ground truth data for training, we compute the velocity and pressure distributions of flows around airfoils with *OpenFOAM*. We consider a space of solutions with a range of Reynolds numbers $Re = [0.5, 5]$ million, incompressible flow, and angles of attack in the range of ± 22.5 degrees.

The models are trained with an L_1 loss for 80000 iterations with an Adam optimizer, a learning rate $\eta = 0.0004$ and batch size $b = 10$. In this context, it is important to normalize the input quantities.

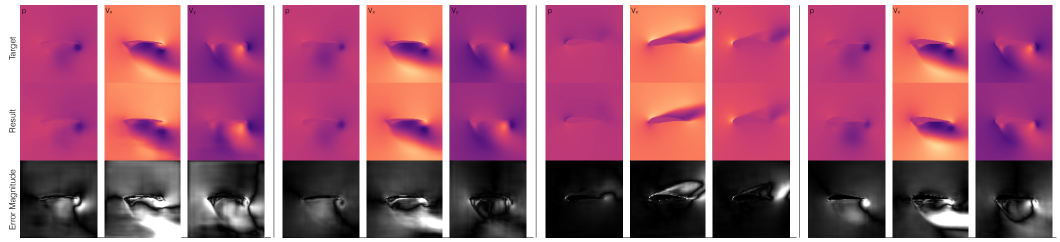


Figure 3: A selection of inference test results with particularly high errors. Each target triple contains, f.l.t.r., $\hat{p}_o$,$\mathbf{v}_{o,x}$,$\mathbf{v}_{o,y}$, the model results are shown below. The bottom row shows error magnitudes, with white indicating larger deviations from the ground truth targets.

Directly performing supervised learning via the OpenFOAM outputs exhibits a very significant average error of 291.34, while a non-dimensionalization reduces the error to only 0.0566, with a further reduction to 0.0136 by removing the pressure null space. An example of how these errors manifest themselves in the inferred solutions is shown in Fig. 2.

With the correct normalization, we can achieve very good flow approximations: A network with 30.6m weights achieves an average relative error of 2.6% across all three output channels. Due to the differences between velocity and pressure functions, this error is not evenly distributed. Rather, the model is trained for reducing L_1 differences across all three output quantities, which yields relative errors of 2.15% for the x velocity channel, 2.6% for y, and 14.76% for pressure values. In Fig. 3 we also show examples of inference case with larger errors. Despite these individual samples, the inference networks learn to represent the complex space of flow solutions with sufficient accuracy, and can produce solutions almost instantly. Next, we leverage this combination of fast runtime with good approximation accuracy in the context of shape optimization.

3 Shape Optimization with Learned Surrogate Models

We consider two-dimensional incompressible steady laminar flows over profiles of given area and look for the minimal drag design. The profile is initialised with a circular cylinder and updated by utilizing steepest gradient descent as optimisation algorithm. The Reynolds number Re_D in the present work is based on the diameter of the initial circular cylinder. It can be also interpreted that the length scale is defined as the equivalent diameter for given area S of an arbitrary shape, i.e. $D = 2\sqrt{S/\pi}$. In the present work, $D \approx 0.394[m]$ is used. The shape of the immersed body is given in terms of a differentiable level-set representation. We use a signed distance function ϕ, with

$$\phi = -d(\Gamma(t)) \text{ for } \boldsymbol{x} \in \Omega\ ; \quad \phi = 0 \text{ for } \boldsymbol{x} \in \partial\Omega \text{ (on } \Gamma)\ ; \quad \phi = d(\Gamma(t)) \text{ for } \boldsymbol{x} \in \mathcal{D} - \Omega \tag{1}$$

where $d(\Gamma(t))$ denotes the Euclidean distance from $\boldsymbol{x}$ to Γ. We compute aerodynamic forces due to pressure distribution and viscous effect acting on the immersed shape on the computational grid, and consider the drag force as the loss in the optimisation, i.e., we minimize

$$\mathcal{L} = \boldsymbol{F}_{pressure} + \boldsymbol{F}_{viscous}. \tag{2}$$

To solve the constrained optimisation problem we proceed with the following steps: Initialise level set function ϕ such that the initial shape (i.e. a circular cylinder) is corresponding to $\phi = 0$. For a given ϕ, calculate drag (i.e. loss $\mathcal{L}$). Terminate if the optimisation converges, e.g. drag history reaches a statistically steady state. Calculate the gradient $\frac{\partial \mathcal{L}}{\partial \phi}$. In practice, we update ϕ using the second-order Runge-Kutta method, and discretise the convection term with a first-order upwind scheme [16,]. To ensure $\|\nabla\phi\| \approx 1$, the fast marching method is used to solve the Eikonal equation. The area of the shape is normalized to a prescribed constant, and lastly, we constrain the barycenter of the shape to be at the origin of the computational domain. These steps are iterated until convergence.

In order to assess the quality of the optimized shapes, we compare two cases, for $Re_D = 1$ and $Re_D = 40$ to optimization with the full OpenFOAM solver and with data from previous work. The resulting comparisons are shown in Fig. 4. The initial shape (shown in green) converges towards a stable final state that matches both reference shapes very well.

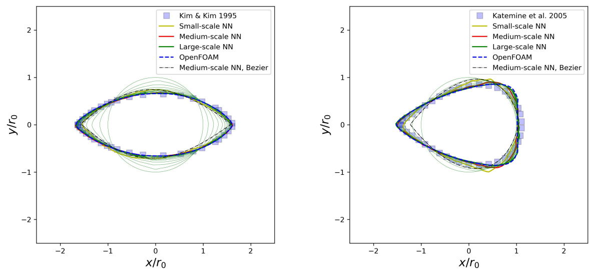
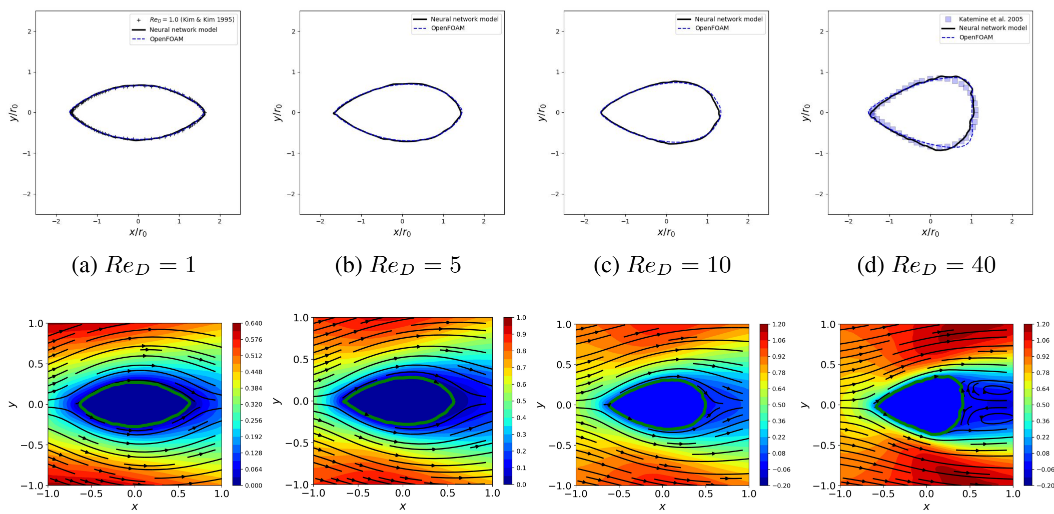


Figure 4: The converged shapes at $Re_D = 1$ (right) and $Re_D = 40$ (left) with intermediate states predicted by large-scale NN model at every 10th iteration.



(a) $Re_D = 1$ (b) $Re_D = 5$ (c) $Re_D = 10$ (d) $Re_D = 40$

(e) $Re = 1$ (f) $Re = 5$ (g) $Re = 10$ (h) $Re = 40$

Figure 5: Shapes after optimization at $Re_D = 1$, 5, 10, and 40. The black solid lines denote the results using neural network model, the blue dashed lines denote the results from *OpenFOAM* and the symbols denote the corresponding reference data. The top row shows the final shapes, while the bottom row visualizes the resulting flow in terms of streamlines.

Next, we turn to a model that was trained for a wider range of Reynolds numbers with $Re_D = [0.5, \cdots, 42]$. Despite the significantly larger, and more complex space of solutions, the network yields robust optimizations that converge towards the ground truth shapes obtained with a full solver. Specific examples are shown in Fig. 5 in terms of converged shapes and resulting flow. We find it is especially encouraging that a model can yield stable shape optimisations across a significant range of Reynolds numbers, as the optimizations with the pre-trained model outperform the OpenFOAM baseline by a speed-up factor of more than 300X.

Additional details for the inference as well as the shape optimisation case will be supplied as supplemental material upon acceptance.

4 Conclusions

We have presented a deep learning based method for the inference of fluid flow simulations. There are numerous avenues for future work in the area of physics-based deep learning, e.g., to employ trained flow models in the context of other inverse problems. The high performance and differentiability of a CNN model yield a very good basis for tough problems, as demonstrated by our shape optimization results. This showcases the possibilities of using deep neural networks as surrogates for a variety of challenging problems in physical sciences.

References

[1] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner. Data-driven discretization: a method for systematic coarse graining of partial differential equations. *arXiv:1808.04930*, 2018.

[2] A. Beck, D. Flad, and C.-D. Munz. Deep neural networks for data-driven turbulence models. *ResearchGate preprint*, 2018.

[3] R. Glowinski and O. Pironneau. On the numerical computation of the minimum-drag profile in laminar flow. *J. Fluid Mech.*, 72:385–389, 1975.

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *Proc. of IEEE Comp. Vision and Pattern Rec.*, 2017.

[5] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv:1710.10196*, 2017.

[6] E. Katamine, H. Azegami, T. Tsubata, and S. Itoh. Solution to shape optimisation problems of viscous fields. *Int. J. Comut. Fluid Dyn.*, 19(1):45–51, 2005.

[7] D. W. Kim and M. Kim. Minimum drag shape in two-dimensional viscous flow. *Int. J. Numer. Methods Fluids*, 21(2):93–111, 2005.

[8] T. Kondoh, T. Matsumori, and A. Kawamoto. Drag minimization and lift maximization in laminar flows via topology optimization employing simple objective function expressions based on body force integration. *Struct. Multidiscipl. Optim.*, 45:693–701, 2012.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105. NIPS, 2012.

[10] J. Li, M. Zhang, J. R. R. A. Martins, and C. Shu. Efficient aerodynamic shape optimization with deep-learning-based geometric filtering. *AIAA J.*, Articles in Advance:1–17, 2020.

[11] O. Pironneau. On optimum profiles in Stokes flow. *J. Fluid Mech.*, 59:117–128, 1973.

[12] O. Pironneau. On optimum design in fluid mechanics. *J. Fluid Mech.*, 64:97–110, 1974.

[13] M. Raissi, A. Yazdani, and G. E. Karniadakis. Hidden fluid mechanics: A navier-stokes informed deep learning framework for assimilating flow visualization data. *arXiv:1808.04327*, 2018.

[14] S. A. Renganathan, R. Maulik, , and J. Ahuja. Enhanced data efficiency using deep neural networks and gaussian processes for aerodynamic design optimization, 2020.

[15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[16] J. A. Sethian and P. Smereka. Level set methods for fluid interface. *Annu. Rev. Fluid Mech.*, 35:341–372, 2003.

[17] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992.

[18] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin. Accelerating eulerian fluid simulation with convolutional networks. In *International Conference on Machine Learning*, volume 70, pages 3424–3433, 2017.

[19] J. Viquerat, J. Rabault, A. Kuhnle, H. Ghraieb, A. Larcher, and E. Hachem. Direct shape optimization through deep reinforcement learning, 2019.

[20] Y. Xie, E. Franz, M. Chu, and N. Thuerey. tempoGAN: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Transactions on Graphics (TOG)*, 37(4):1–15, 2018.

[21] R. Yondo, E. Andrés, and E. Valero. A review on design of experiments and surrogate models in aircraft real-time and many-query aerodynamic analyses. *Prog. Aerosp. Sci.*, 96:23–61, 2018.