Transformer-based Prediction of Vehicle Aerodynamics

John Higgins, Cyril Ngo Ngoc, Nicolas Fougère, Faron Hesse, Victor Oancea, David Sondak

Dassault Systèmes SIMULIA Corp.
175 Wyman Street, Waltham MA (USA)

John.HIGGINS@3ds.com
Cyril.NGONGOC@3ds.com
Nicolas.FOUGERE@3ds.com
Faron.HESSE@3ds.com
Victor.OANCEA@3ds.com
David.SONDAK@3ds.com

Abstract: The vehicle market is evolving rapidly. New players are entering the market, many variants of a vehicle are investigated prior to freezing the design, and more. In this context, vehicle aerodynamics is ever more crucial. It directly impacts the vehicle range and plays a major role in meeting regulation targets. Vehicle manufacturers must also keep in mind the need for a shorter time-to-market, where one must design faster and not permit late-stage redesign. Therefore, faster and earlier assessment of vehicle aerodynamics is imperative. Computational Fluid Dynamics (CFD) has opened the door to virtual aerodynamic testing, allowing manufacturers to test their vehicle shapes before developing a costly and time-intensive prototype that then needs to be experimented on using a wind tunnel. While high-fidelity CFD, such as the PowerFLOW® software from Dassault Systèmes, will remain an integral part of the aerodynamic development process of major OEMs, the growth of machine learning (ML) and continual improvement of its algorithms has opened doors to speed-up computational aerodynamics, allowing automotive manufacturers to get feedback on their vehicle design in a matter of minutes. The current work illustrates how aerodynamic data obtained using the Lattice Boltzmann Method with PowerFLOW® combined with transformer-based ML can enable car manufacturers to obtain clean 3D contour plots of the vehicle's surface X-force (or any other simulated quantity) distribution and the associated integrated vehicle drag force within several minutes on a single GPU (after training of the ML model). This represents a significant reduction in computational cost and time.

Contribution: 2025 FKFS Conference on Vehicle Aerodynamics and Thermal Management
15 – 16 October 2025 | Leinfelden-Echterdingen

1 Introduction

The ground vehicle market continues to shift its focus to more energy-efficient designs. As a result, aerodynamic performance of the vehicle is more critical than ever in achieving range and regulatory goals. In order for a vehicle platform to reach its aerodynamic performance targets, vehicle manufacturers must evaluate more designs virtually. In order to meet time-to-market deadlines, there is a need to evaluate more designs earlier in the design phase (such as the concept design phase). This is sometimes referred to as "left-shifting" of simulation within the design cycle.

Historically, high-fidelity Computational Fluid Dynamics (CFD) tools using the Lattice Boltzmann Method (LBM) such as the PowerFLOW® software from Dassault Systèmes have been used to virtually evaluate vehicle aerodynamics. This will remain an integral tool in both early and late stages of the design process. In early stages of the design process, high-fidelity CFD will be used to generate datasets with which to train machine learning (ML) models. In later stages of the design process where the highest level of simulation accuracy is required, high-fidelity CFD will continue to be used to verify final designs and validate physical tests. With the growth of machine learning surrogate models, it is possible to evaluate the aerodynamics of a vehicle virtually in a matter of minutes [1]. This will accelerate the aforementioned "left-shifting" of simulation within the design cycle by allowing design studios to quickly and easily evaluate the aerodynamic performance. This will ultimately allow vehicle OEMs to evaluate significantly more designs than they would have using traditional CFD methods, leading to more innovative and efficient vehicles.

Vehicle OEMs have been using CFD to predict their vehicle aerodynamics performance at different stages within the design process for decades. As a result, they have acquired a large database of simulation results. These existing datasets can be leveraged as a promising starting point for training ML models. Integrating new high fidelity simulation datasets into these trained models will further enhance their prediction capabilities.

There has been a surge of effort in scientific machine learning for CFD in recent years covering most aspects of CFD with varying degrees of success [2] [3] [4]. Early work focused on the development of novel RANS turbulence model closures with more recent extensions being developed for LES closures [5] [6] [7]. A large amount of effort has centered on physics-informed neural networks (PINNs) [8] [9] [10] as a way of solving conservation equations using neural networks. PINNs have some appealing qualities including straightforward blending of data with conservations laws, being fully differentiable, easy adaptation to complex geometries, and the ability to easily incorporate a variety of boundary conditions. In spite of these apparent advantages, PINNs are not competitive with traditional solvers and remain an active area of research. Machine learning algorithms have also been developed to discover new discretization strategies [11], accelerate traditional solvers [12], improve mesh generation [13] [14], discover equations from data [15], generate super-resolved flow fields from under-resolved data [16] [17] [18] [19], and most relevant to this manuscript, develop surrogate models [20] [21] [22] [23] [24] [25]. The majority of the work on ML has focused on data generated from Navier-Stokes simulations including RANS, LES, and DNS datasets. However, some recent work has emerged on fully-differentiable Lattice Boltzmann solvers [26], learning Lattice Boltzmann collision operators [27], and learning collision operators for the Bolztmann equation [28]. Regardless of the algorithm used, any scientific machine learning approach is strongly influenced by the quality of the training data. The present work uses PowerFLOW® CFD to generate state of the art training datasets for the development of surrogate models.

In this study, we apply a transformer-based machine learning surrogate model [25] [29] to the design of a vehicle external surface. The input to the neural network is a set of PowerFLOW® LBM CFD simulations applied to a sedan vehicle with varying design changes made to the A-surface. It is important to note that this ML methodology uses history-based data as input, meaning that geometric parameters are not required as input to the neural network. The reference vehicle model used in the present study is the DrivAer model from TU Munich [30]. PowerFLOW® CFD was previously validated on this particular vehicle model [31]. The trained neural network was used to predict the X-component of the surface force contours. This result was then used to calculate the overall vehicle drag coefficient (C_D) via integration over the entire vehicle surface. Both the surface X-force contours and C_D were found to be in excellent agreement with the PowerFLOW® results, even for vehicle designs outside of the training set. To the best of our knowledge, this is the first time an ML model has been trained using data generated with the Lattice Boltzmann Method in the context of vehicle aerodynamics.

2 CFD Numerical Approach to Generate the Dataset

The design set used for training contained large changes to the vehicle A-surface, including shape changes to the front bumper, windshield, and rear glass as well as ride height and front wheel deflectors on/off. These shape changes were applied through a mesh morphing technique. Figure 1 shows a subset of the designs that were used to train the neural network. A total of 50 DrivAer designs were included in this study.

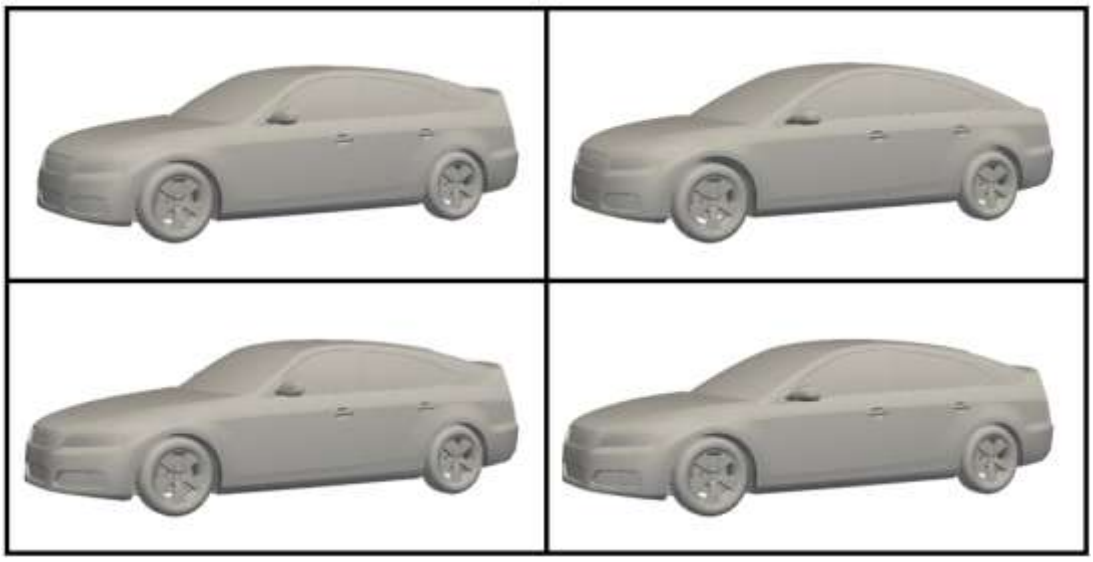

Figure 1: Subset of DrivAer designs used in study

All 50 of the DrivAer designs used in the present study were simulated using the PowerFLOW® LBM CFD solver. A full numerical description along with information regarding the Very Large Eddy Simulation (VLES) turbulence model can be found within the following references [32] [33] [34] [35] [36] [37] [38] [39] .

Such a CFD approach has been extensively validated by comparison to physical test from wind tunnel data [40] [41] [42] [43]. This gives confidence into the accuracy of the database.

The results of these CFD simulations were used as the input dataset to the neural network. The external aerodynamics simulation setup utilized an open-road scenario, consisting of a large domain with a velocity inlet far upstream of the vehicle and a pressure outlet far downstream. The walls and ceiling of the domain were modeled as frictionless walls. The floor was modeled with a moving wall condition to match the freestream velocity. Incompressible flow was assumed, with a freestream of $U_\infty = 140\ kph$ and an ambient temperature of $T = 20°C$. The reference area used for drag coefficient calculation was $A = 2.156\ m^2$. For added realism, the wheels of the vehicle were encased in a local rotating reference frame in order to simulate the effects of rotating wheel geometry with a sliding mesh technique.

The fluid volume was setup using a variable resolution scheme. The finest fluid cell size was 2.5mm, applied on regions containing highly unsteady flow and/or large velocity gradients such as mirrors, a-pillars, and wheels. The fluid and surface grid used for simulation was generated automatically by the PowerFLOW® discretizer. This resulted in 97M elements in the fluid domain and 7M elements on the vehicle surface.

All of the transient CFD simulations were run for 2 seconds of physical time. The end of the initial transient was identified using the approach described in [44]. The time-averaged drag force for all of the simulations converged within 1 count (0.001 C_D) accuracy with a 95% confidence interval. Figure 2 shows an example of the drag time history plot for one of the simulations. This 50-simulation set resulted in a broad range of time-averaged overall C_D values, ranging from 0.247 to 0.308.

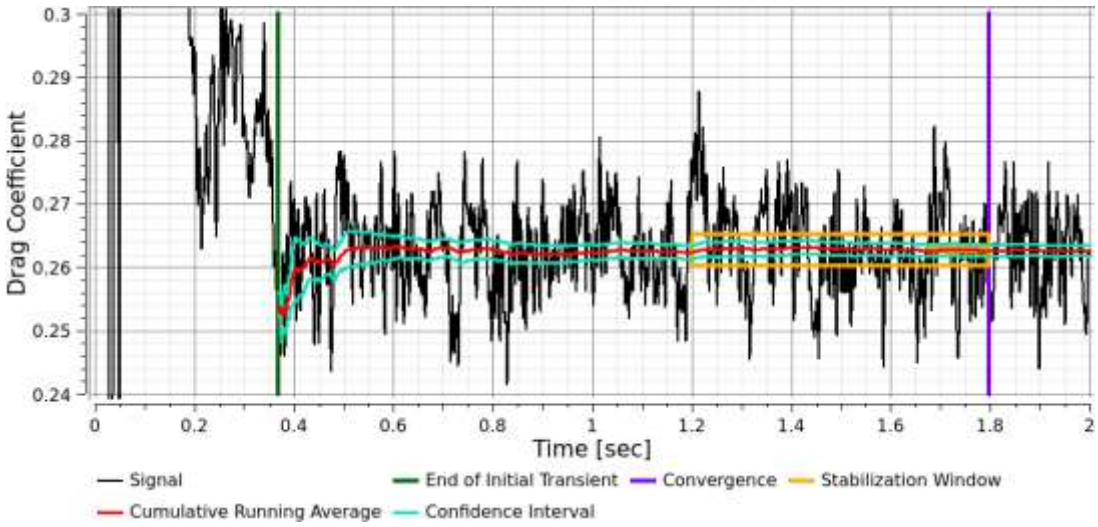

Figure 2: Transient drag history plot for Run 1

Before inputting the CFD dataset into the neural network, some post-processing was performed. The data on the surface of the vehicle directly from CFD contained around 3M points. This was too large for the neural network to run on a single-GPU card due to memory limitations. Therefore, the per-simulation dataset size was reduced from ≈3M points to 130k points via Voronoi kernel spatial interpolation [45]. Doing so introduced a mean error of 1.73% into the interpolated dataset. All of the results forthcoming neglect this error and compare the interpolated data (ground truth) to the ML prediction.

3 Machine Learning Training and Predictions

A transformer-based machine learning technique [25] [29] was used to train on the entire vehicle surface X-force contours from the PowerFLOW® simulations. A 90-10 training-test split of the 50-simulation dataset was used, resulting in 45 training points and 5 blind test points. Figure 3 shows the training and test loss curves where we track the evolution of the mean squared error for the training and test sets as a function of epoch. The training had completed after 700 epochs.

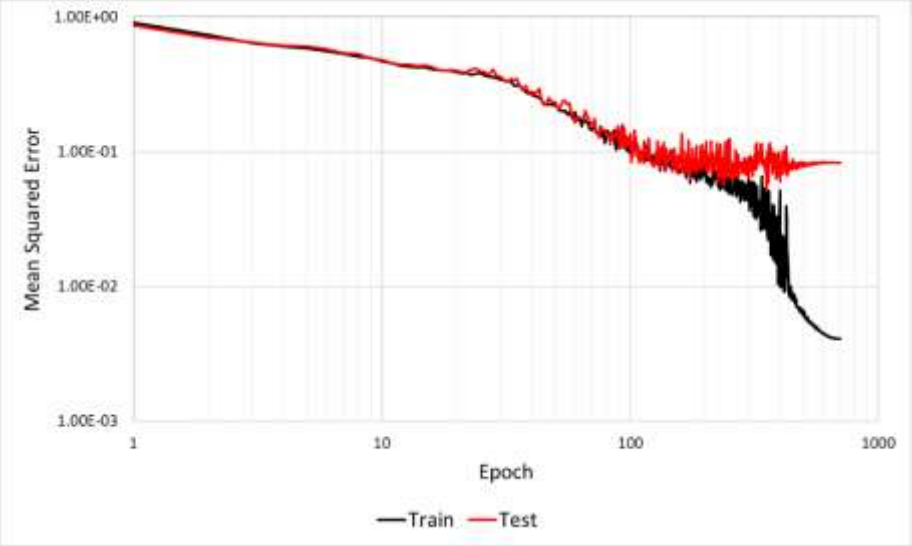

Figure 3: Training and test loss curves

The total vehicle drag force is calculated by integrating over the vehicle surface the X-component of the force per unit area surface predictions from the machine learning model. Analyzing both of these metrics is key to evaluating the accuracy of the model. Figure 4 shows a comparison of the X-component of the surface force contours for the test run that had the best C_D ML prediction. The top image shows the ground truth result, the middle image shows the ML prediction, and the bottom image shows their difference. The C_D error for this design point is 0.18%, indicating very good agreement. Qualitatively, the surface X-Force contour predictions correlate well with the ground truth. There are some minor discrepancies seen between the ground truth and ML prediction. These differences are primarily present at regions of high curvature, such as the front bumper, mirrors, and wheels.

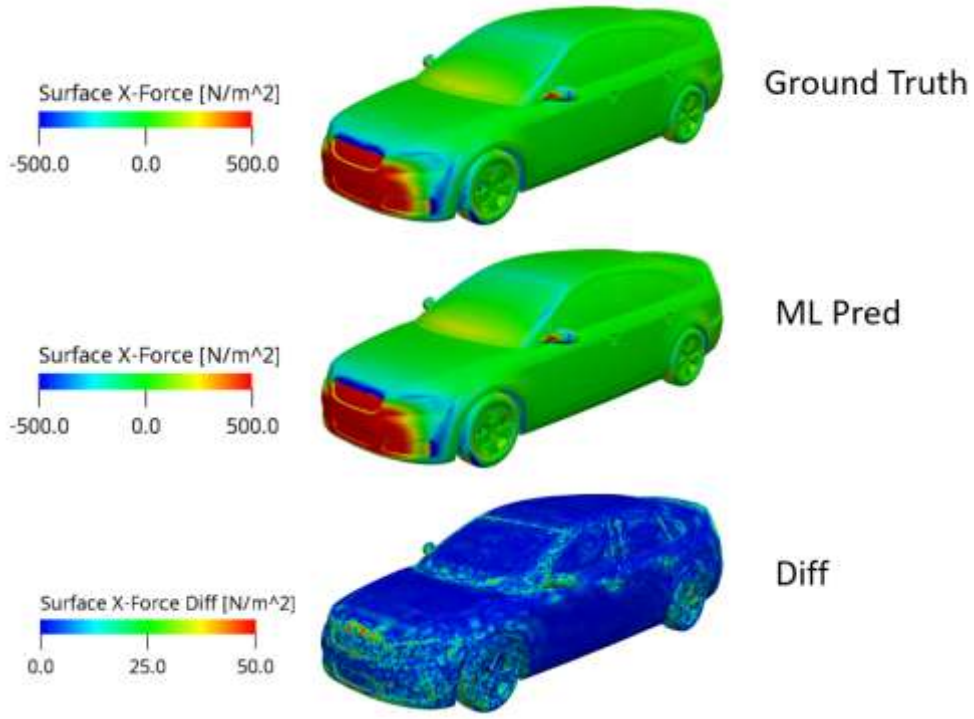

Figure 4: Surface X-Force for Run 18. This is the best prediction among all of the test points. top: Ground truth, middle: ML prediction, bottom: difference

Figure 5 shows a comparison of the surface X-force contours for the test run that had the worst C_D ML prediction. Even here, the ML prediction is in quite good agreement. The C_D error for this design point is 1.00%. The discrepancies between ground truth and ML prediction are in similar regions as the “best” run.

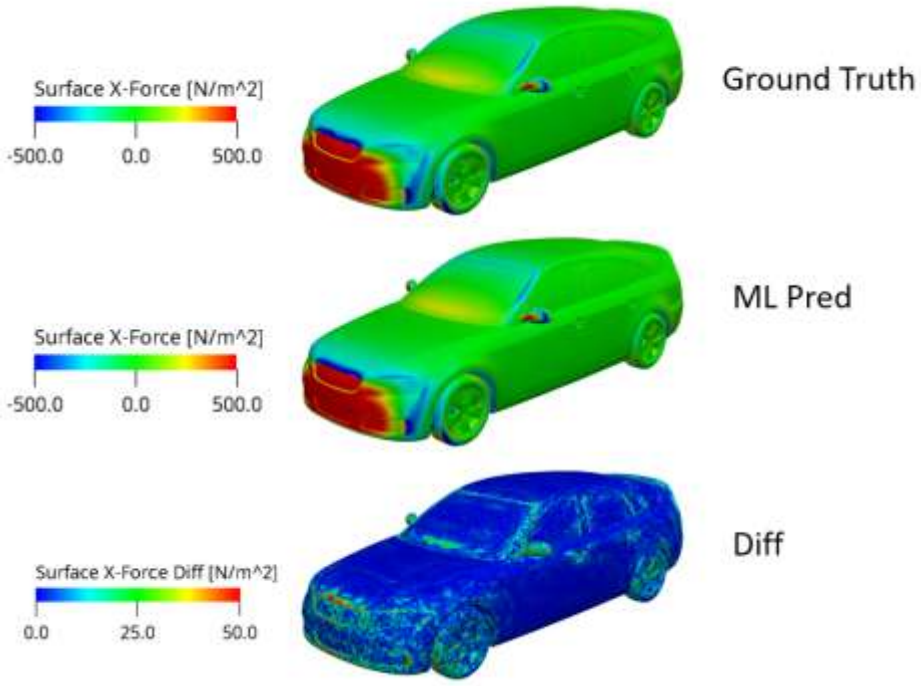

Figure 5: Surface X-Force for Run 42. This is the worst prediction among all of the test points. top: Ground truth, middle: ML prediction, bottom: difference

The overall performance of the ML model is shown in Figure 6. Here, the overall C_D derived from the predicted X-component of surface force contours is plotted against the C_D computed from the ground truth data. A perfect prediction would result in all of the data points falling on the 45-degree line. The mean error for the entire dataset (training points included) is 0.37% and the mean error for the test points only is 0.59%. As shown in Figure 6, all of the test points land on or within the 1% error bars.

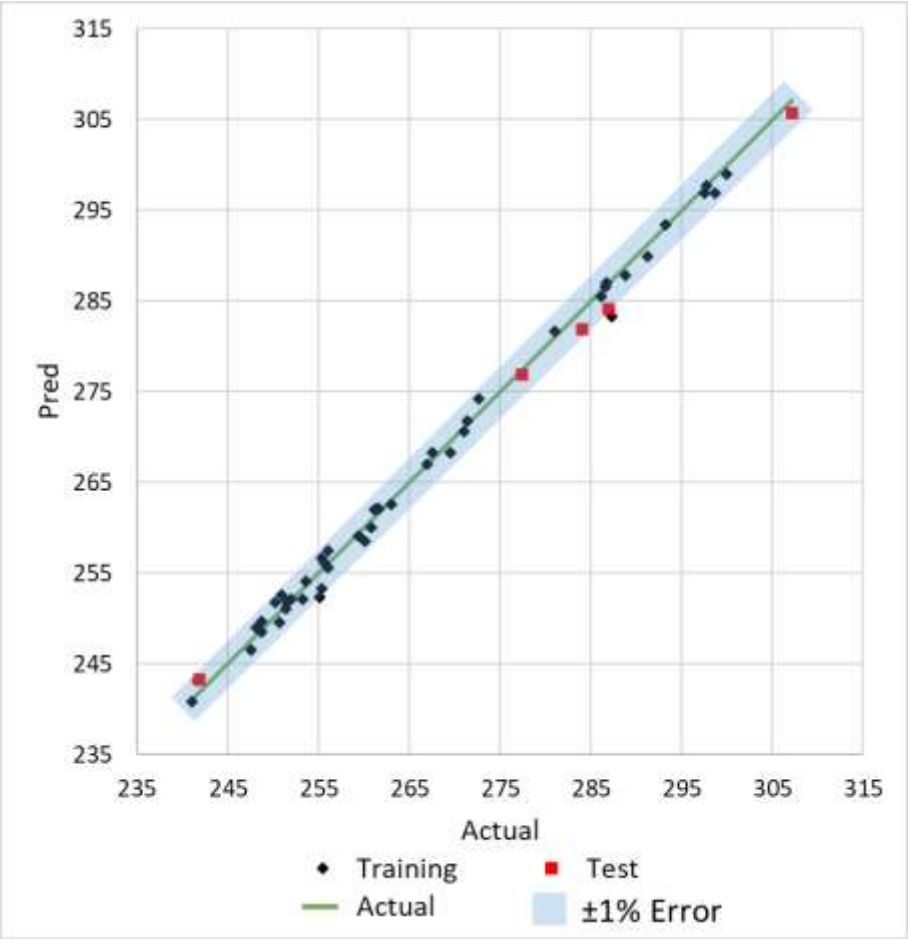

Figure 6: ML predictions (Pred) vs. ground truth (Actual) for overall vehicle drag (C_D) in counts. 1% error bars

It is also important to consider the ranking of designs based on an important metric (C_D in this case). This ensures that the neural network is able to predict trends accurately, which is key during the vehicle design phase. Figure 7 shows a ranking of the C_D for all of the test datasets, ordered from lowest to highest. Here, the ranking remains consistent between ground truth result and ML prediction for all of the test points. This indicates that the trained model can predict trends accurately, even between closely ranked designs like test points 2, 3, and 4.

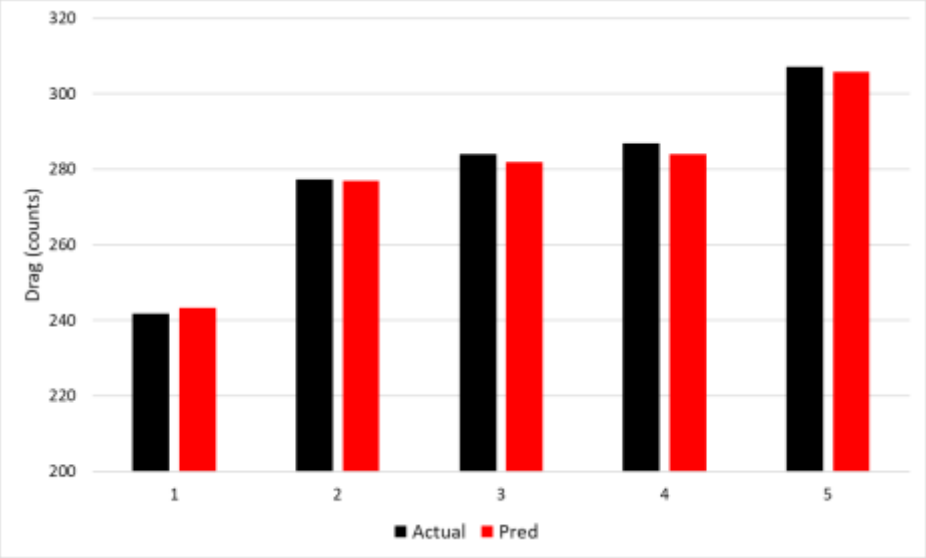

Figure 7: Overall vehicle drag for test runs. Ranked in ascending order based on the ground truth result.

4 Conclusion

Through this study, we have shown the ability for transformer-based neural networks to successfully create surrogate models from high-fidelity LBM CFD simulations. When applied to concept aerodynamics simulations, the key performance indicators of overall drag and surface X-Force contours were predicted by the neural network with a high level of accuracy. Specifically, in this study we utilized a 50-simulation dataset with a 90-10 training-test split, resulting in 45 training points and 5 test points. Training the model on the 3D contours of surface X-force resulted in an integrated drag force mean error of 0.59% for the test points. Moreover, the ranking of the different vehicle designs was accurately predicted by the ML model, ensuring that the methodology presented in this paper offers reliable design direction.

The 3D contour predictions are made in a matter of minutes, instead of the hours required to simulate high-fidelity CFD, providing almost instantaneous design guidance based on vehicle drag performance. This approach will enable designers to explore the aerodynamic impact on a wider variety of vehicle shapes in the early stages of the design process. This was previously not possible with traditional CFD simulation techniques.

Moving forward, the next steps are to evaluate the impact of expanding the training set, which would potentially improve the error even further. We also plan on applying this technique to other vehicles to ensure robustness. Finally, we would like to explore applying this history-based neural network technique to other CFD application areas such as acoustics and thermal applications.

5 References List

[1] X. Du, P. He und J. R. R. A. Martins, “Rapid airfloil design optimization via neural networks-based parameterization and surrogate modeling,” Bd. 113, p. 106701, June 2021.

[2] S. L. Brunton, B. R. Noack und P. Koumoutsakos, “Machine Learning for Fluid Mechanics,” *Annu. Rev. Fluid Mech.*, Bd. 52, Nr. 1, pp. 477-508, Jan. 2020.

[3] O. Obiols-Sales, A. Vishnu, N. Malaya und A. Chandramowliswharan, “CFDNet: a deep learning-based accelerator for fluid simulations,” in *Proceedings of the 34th ACM International Conference on Supercomputing*, Barcelona Spain, 2020.

[4] R. Vinuesa und S. L. Brunton, “Enhancing computational fluid dynamics with machine learning,” *Nat. Comput. Sci.*, Bd. 2, Nr. 6, p. 358–366, Jun. 2022.

[5] J. Ling, A. Kurzawski und J. Templeton, “Reynolds averaged turbulence modelling using deep neural networks with embedded invariance,” *J. Fluid Mech.*, Bd. 807, pp. 155-166, 2016.

[6] A. Beck und M. Kurz, “A perspective on machine learning methods in turbulence modeling,” *GAMM-Mitteilungen*, Bd. 44, Nr. 1, p. e202100002, Mar. 2021.

[7] M. Milano und P. Koumoutsakos, “Neural Network Modeling for Near Wall Turbulent Flow,” *J. Comput. Phys.*, Bd. 182, Nr. 1, pp. 1-26, 2002.

[8] I. E. Lagaris, A. Likas und D. I. Fotiadis, “Artificial neural networks for solving ordinary and partial differential equations,” *IEEE Trans. Neural Netw.*, Bd. 9, Nr. 5, pp. 987-1000, Sep. 1998.

[9] I. E. Lagaris, A. C. Likas und D. G. Papageorgiou, “Neural-network methods for boundary value problems with irregular boundaries,” *IEEE Trans. Neural Netw.*, Bd. 11, Nr. 5, pp. 1041-1049, Sep. 2000.

[10] M. Raissi, P. Perdikaris und 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.*, Bd. 378, pp. 686-707, Feb. 2019.

[11] Y. Bar-Sinai, S. Hoyer, J. Hickey und M. P. Brenner, “Learning data-driven discretizations for partial differential equations,” *Proc. Natl. Acad. Sci.*, Bd. 116, Nr. 31, pp. 15344-15349, Jul. 2019.

[12] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner und S. Hoyer, “Machine learning–accelerated computational fluid dynamics,” *Proc. Natl. Acad. Sci.*, Bd. 118, Nr. 21, p. e2101784118, May 2021.

[13] K. Huang, M. Krügener, A. Brown, F. Menhorn, H. J. Bungartz und D. Hartmann, “Machine Learning-Based Optimal Mesh Generation in Computational Fluid Dynamics,” *arXiv: arXiv:2102.12923*, 25 Feb. 2021.

[14] S. Sanchez-Gamero, O. Hassan und R. Sevilla, “A Machine Learning Approach to Predict Near-optimal Meshes for Turbulent Compressible Flow Simulations,” *Int. J. Comput. Fluid Dyn.*, Bd. 38, Nr. 2-3, pp. 221-245, Mar. 2024.

[15] S. L. Brunton, J. L. Proctor und J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proc. Natl. Acad. Sci.*, Bd. 113, Nr. 15, pp. 3932-3937, Apr. 2016.

[16] K. Fukami, K. Fukagata und K. Taira, “Super-resolution reconstruction of turbulent flows with machine learning,” *J. Fluid Mech.*, Bd. 870, pp. 106-120, Jul. 2019.

[17] K. Fukami, K. Fukagata und K. Taira, “Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows,” *J. Fluid Mech.*, Bd. 909, p. A9, Feb. 2021.

[18] K. Fukami und K. Taira, “Single-snapshot machine learning for super-resolution of turbulence,” *J. Fluid Mech.*, Bd. 1001, p. A32, Dec. 2024.

[19] J. Page, “Super-resolution of turbulence with dynamics in the loss,” *J. Fluid Mech.*, Bd. 1002, p. R3, Jan. 2025.

[20] K. Azizzadenesheli, N. Kovachki, Z. Li, M. Liu-Schiaffini, J. Kossaifi und A. Anandkumar, “Neural operators for accelerating scientific simulations and design,” *Nat. Rev. Phys.*, Bd. 6, Nr. 5, pp. 320-328, Apr. 2024.

[21] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart und A. Anandkumar, “Fourier Neural Operator for Parametric Partial Differential Equations,” *arXiv: arXiv:2010.08895*, 17 May 2021.

[22] L. Lu, P. Jin und G. E. Karniadakis, ““DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators,” *Nat. Mach. Intell.*, Bd. 3, Nr. 3, pp. 218-229, Mar. 2021.

[23] J. Bi, C. Ngo Ngoc, J. Yao und V. Oancea, “Towards 3d interactive design exploration via neural networks,” in *presented at the NAFEMS World Congress 2023*, Tampa, FL, 2023.

[24] Y. Chen, J. Bi, C. Ngo Ngoc, V. Oancea, J. Cagan und L. B. Kara, “Attention to Detail: Fine-Scale Feature Preservation-Oriented Geometric Pre-training for AI-Driven Surrogate Modeling,” *arXiv preprint arXiv:2504.20110*, 3 May 2023.

[25] H. Wu, H. Luo, H. Wang, J. Wang und M. Long, “Transolver: A Fast Transformer Solver for PDEs on General Geometries,” *arXiv: arXiv:2402.02366*, 01 Jun. 2024.

[26] M. Ataei und H. Salehipour, “XLB: A differentiable massively parallel lattice Boltzmann library in Python,” *Comput. Phys. Commun.*, Bd. 300, p. 109187, Jul. 2024.

[27] A. Corbetta, A. Gabbana, V. Gyrya, D. Livescu, J. Prins und F. Toschi, “Toward learning Lattice Boltzmann collision operators,” *Eur. Phys. J. E*, Bd. 46, Nr. 3, p. 10, Mar. 2023.

[28] S. T. Miller, N. V. Roberts, S. D. Bond und E. C. Cyr, “Neural-network based collision operators for the Boltzmann equation,” *J. Comput. Phys.*, Bd. 470, p. 111541, Dec. 2022.

[29] Z. Hao, Z. Wang, H. Su, C. Ying, Y. Dong, S. Liu, Z. Cheng, J. Song und J. Zhu, “GNOT: A General Neural Operator Transformer for Operator Learning,” *arXiv: arXiv:2302.14376*, Feb. 2023.

[30] A. Heft, T. Indinger und N. Adams, “Introduction of a New Realistic Generic Car Model for Aerodynamic Investigations,” SAE Technical Paper 2012-01-0168, 2012.

[31] G. Parenti, G. Martins, A. Martinez und T. Linden, “PowerFLOW© Uncertainty Quantification using a DrivAer model,” in *4th Automotive CFD Prediction Workshop*, Belfast, 2024.

[32] S. Chapman und T. Cowling, “The Mathematical Theory of Non-Uniform Gases,” Cambridge Univerity Press, 1990.

[33] P. L. Bhatnagar, E. P. Gross und M. Krook, “A Model for Collision Processes in Charged and Neutral One-Component System,” *Phys. Rev,* Bd. 94, pp. 511-525, 1954.

[34] H. Chen, “Volumetric Formulation of the Lattice Boltzmann Method for Fluid Dynamics: Basic Concept,” *Phys. Rev. E,* Bd. 58, pp. 3955-3963, 1998.

[35] Y. Zhou, R. Zhang, I. Staroselsky und H. Chen, “Numerical Simulation of Laminar and Turbulent Buoyancy-Driven Flows using a Lattice-Boltzmann based Algorithm,” *Int. J. of Heat and Mass Transfer, 47,* pp. 4869-4879, 2004.

[36] Y. Li, R. Shock, R. Zhang und H. Chen, “Numerical Study of Flow Past an Impulsively Started Cylinder by Lattice Boltzmann Method,” *J. Fluid Mech,* Bd. 519, pp. 273-300, 2004.

[37] H. Chen, S. Orszag, I. Staroselsky und S. Succi, “Expanded Analogy between Boltzmann Kinetic Theory of Fluid and Turbulence,,” *J. Fluid Mech.,* Bd. 519, pp. 301-314, 2004.

[38] H. Chen, S. Kandasamy, S. Orszag und Shock et al., “Extended Boltzmann Kinetic Method For Turbulent Flows,” *Science,* Bd. 301, pp. 633-636, 2003.

[39] H. Chen, C. Teixeira und K. Molvig, “Realization of Fluid Boundary Conditions via Discrete Boltzmann Dynamics,” *Intl. J. Mod. Phys. C,* Bd. 9, Nr. 8, pp. 1281-1292, 1998.

[40] M. Gleason, B. Duncan, J. Walter und et al., “Comparison of Computational Simulation of Automotive Spinning Wheel Flow Field with Full Width Moving Belt Wind Tunnel Results,” *SAE Int. J. Passeng. Cars - Mech. Syst. 8(1),* 2015.

[41] A. Guzman, Y. Cho, J. Tripp und K. Srinivasan, “Further Analyses on Prediction of Automotive Spinning Wheel Flowfield with Full Width Moving Belt Wind Tunnel Results,” *SAE Int. J. Passeng. Cars - Mech. Syst. 10(2),* pp. 600-618, 2017.

[42] K. Sbeih, A. Guzman, D. Barrera Garcia, N. Fougere und et al., “Accurate Automotive Spinning Wheel Predictions Via Deformed Treaded Tire on a Full Vehicle Compared to Full Width Moving Belt Wind Tunnel Results,” SAE Technical Paper 2023-01-0843, 2023.

[43] N. Fougere, M. Demeo, H. Tuit Farquhar, D. Oliveira und et al., “Transient Aerodynamics Simulations of a Passenger Vehicle during Deployment of Rear Spoiler,” SAE Technical Paper 2024-01-2536, 2024.

[44] C. Mockett, T. Knacke und F. Thiele, “Detection of initial transient and estimation of statistical error in time-resolved turbulent flow data,” in *Proceedings of the 8th International Symposium on Engineering Turbulence Modelling and Measurements*, 2010.

[45] W. Schroeder, K. Martin und W. Lorensen, “The Visualization Toolkit, An Object-Oriented Approach To 3D Graphics,” 2006, pp. 359-362.