Proceedings of the 1st International Symposium on AI and Fluid Mechanics
Paper No S5 P6

Accelerating Evolutionary RANS Turbulence Modelling through Transformer-based Augmentation

M Reissmann[1], Y Fang[1], A S H Ooi[1], R D Sandberg[1]

[1]Department of Mechanical Engineering, University of Melbourne, Melbourne, AU

Keywords: Turbulence Modelling, RANS, Symbolic Regression, Transformer

Abstract. For the training process of machine learning (ML)-guided Reynolds-Averaged Navier-Stokes (RANS) closures, the use of computational fluid dynamics (CFD) simulations as feedback can be highly beneficial. However, this results in significant computational costs, especially when considering engineering-relevant and complex flow scenarios. The efficiency of identifying robust and more accurate models depends not only on the computational cost of the RANS calculations but also on the initial distribution of the candidate solutions, particularly the symbolic composition of the expressions. Although previously explored closures from simpler, phenomenologically similar optimizations exist, current optimization workflows largely fail to effectively leverage this information to limit the search space. This study presents a framework that integrates Gene Expression Programming with a transformer architecture, predominantly used in language modeling. We firstly train a transformer on diverse closure models derived from optimizations across various canonical flow scenarios to identify recurring patterns and correlations within these terms. We then use a complex three-dimensional test case to evaluate the benefit of inferring candidate solutions generated by a language model for the initial population. Our results demonstrate that language-model-augmented initial candidate solutions, informed by previously investigated models, enhance convergence rates and yield more accurate closure models.

1. Introduction

Despite the firm reliance on model assumptions, Reynolds averaged Navier-Stokes (RANS) calculations continue to play a crucial role in engineering product development [1]. While computational limitations will likely constrain the execution of numerous high-fidelity simulations for engineering relevant flows [2] in the near future, emerging machine learning (ML) frameworks, in combination with a large volume of available data, offer an immense potential to improve the low computation cost models and enhance the accuracy of solutions obtained from the RANS framework.

Data-driven approaches, including tensor-based neural networks [3] and explicit algebraic Reynolds-stress (EARSM) modeling approaches using symbolic regression techniques such as Gene Expression Programming [4] (GEP) or Sparse Regression of Turbulent Stress Anisotropy [5], have been developed to address limitations of RANS in various flow scenarios. These techniques demonstrate improved performance [6, 7], yet face challenges with ill-conditioning [8] and out-of-distribution prediction, resulting in costly simulation-based optimization cycles [9] or expanded training scenarios [10]. Although thest strategies can yield robust models, the inherent limitations imposed by averaging the flow field raise concerns about generalizability [11, 12], requiring frequent model adaptions or new explorations that scale with the problem complexity.

To accelerate the model exploration, studies like [13, 14, 15, 16] successfully employed pre-trained approaches relying on the transformer architecture heavily utilized for natural language processing. While the authors demonstrate fast inference, the techniques require many synthetic equations (10^5) and an extensive training process, which is not feasible for problems requiring turbulence modeling.

Extensions of these strategies [17, 18, 19] circumvent the training procedure by employing models from OpenAI or Meta and demonstrate strong performance on standard benchmarks. However, the open-source nature of these benchmarks raises concerns about data contamination (benchmark already present in the training dataset of the language model) [20]. Consequently, the outlined performance may indicate the ability to extract existing equations rather than discover unknown ones, thus potentially overstating their practical use for turbulence modeling applications.

The above-mentioned approaches demonstrate either fast inference with high data demands or data

efficiency with costly exploration. This study combines the complementary strengths of integrating transformer-based sequence learning with the flow solver incorporating high-quality physical knowledge, framed as a transfer learning methodology. Here, the language model preserves sequential information from simpler flow cases and augments the initial candidate solutions for subsequent optimization problems. Hence, fewer training iterations are needed to explore and adopt appropriate solutions. To assess the performance of the proposed method, we conduct a validation on different flow scenarios with increasing complexity, ranging from a canonical square duct to a 3D highly bent serpentine duct.
The remainder of this paper is structured as follows: Section 2 outlines the core functionality of the GEP as well as the methodology employed for its application in turbulence modeling. We further explain the core parts of the transformer architecture that have been adapted and how they interact with the evolutionary method. Section 3 demonstrates the application of the framework and exemplifies the performance improvement compared to the standard method. The study concludes with a short reflection and an outline for possible improvements.

2. Methodology

In this section, we first briefly introduce the concept of the GEP and how it is applied to develop a turbulence model, including the underlying model hypothesis and the input features. Subsequently, an overview of the language model-driven transfer learning approach is provided, along with an explanation of the components adapted for interacting with the GEP.

2.1. GEP Modelling Strategy

The GEP developed by Ferreira [21] marks a gradient-free, population-based optimization method following Darwin's principle of the "survival of the fittest". The method randomly initializes several candidate solutions (individuals), each represented as a linear string of symbols (genotype), separated into a head and a tail section. At each optimization step (generation), the linear strings are decoded into algebraic equations (phenotype) and evaluated utilizing a cost function, resulting in a loss value (fitness). Subsequently, a mechanism selects a subset from the best-performing ones, which are then undergoing modifications introduced by genetic operators. This process is repeated until a termination criterion is met, defined as a computational budget (number of iterations, runtime) or by achieving a specified threshold for the cost function value.
As for turbulence modeling we require the approximation of tensorial quantities, we employ the GEP framework introduced by Weatheritt and Sandberg [22]. Here, the optimization problem is formulated based on extending the Boussinesq hypothesis with non-linear anisotropy terms according to Pope [23], to close the Reynolds stress (τ_{ij}) as:

$$\boldsymbol{\tau}_{ij} = \frac{2}{3}k\boldsymbol{\delta}_{ij} - 2\nu_t\mathbf{S}_{ij} + 2k\mathbf{a}_{ij} \tag{1}$$

with k signifying the turbulent kinetic energy, ν_t turbulent viscosity (solved using the $k-\omega$ SST model [24]), $\mathbf{S}_{ij}$ the strain rate, and the anisotropic stress tensor $\mathbf{a}_{ij}$. This tensor can be constructed through a finite polynomial expansion of ten independent basis tensors:

$$\mathbf{a}_{ij} = \sum_{\lambda} G^{(\lambda)} \cdot \mathbf{T}_{ij}^{(\lambda)} . \tag{2}$$

While $\mathbf{T}_{ij}$ for $\lambda \in \{1, \ldots, 10\}$ can be derived from the non-dimensionalized mean strain rate ($\mathbf{s}_{ij}$) and the non-dimensionalized mean vorticity rate $\mathbf{w}_{ij}$, the functional form of $G^{(\lambda)}$ and characterization of the input space remains undefined. Following the literature [22, 9], we incorporate the first and second invariants (I_1,I_2). Additionally, we include flow features belonging to the blending function of the $k-\omega$ SST (N_3), alongside the Reynolds number based on wall distance (N_1) and the streamline pressure

gradient (N_2) as outlined in [10]. In summary, the regression problem can be formulated as:

$$\min_{a_{ij}} \|\boldsymbol{\phi} - \hat{\boldsymbol{\phi}}\|_L \tag{3}$$

whereby $\boldsymbol{\phi}$ signifies a quantity gathered from an experiment or a high-fidelity simulation, representing a scalar, vector, or tensor field, and $\|\cdot\|_L$ indicates an appropriate norm. The variable represented by $\hat{\boldsymbol{\phi}}$ relates to the corresponding quantity obtained from performing a low-fidelity prediction using a candidate model generated by the GEP. Thus, a prediction for each model needs to be performed as an evaluation procedure, rather than a fitting on high-fidelity data, to avoid inconsistencies or to mitigate issues related to ill-conditioning [8, 25].

2.2. Language-model-based Transfer Learning

When applying population-based optimization methods, the initialization procedure significantly impacts the quality and convergence characteristics of the explored solutions. Instead of the typical approach of employing a randomly created start population, we adapt the transfer learning methodology [26, 27] to enhance this process. The technique can be formally described using the terminology of a (D) and task (T). The domain $\mathsf{D} = \{\mathbf{X}, P(\mathbf{X})\}$ comprises a data space $\mathbf{X}$ and its probability distribution $P(\mathbf{X})$, while a task $\mathsf{T} = \{\mathbf{Y}, f(.)\}$ maps from $\mathbf{X}$ to an output space $\mathbf{Y}$ via the decision function f. The approach assumes that candidate solutions obtained from exploring functions f_s in domain D_s (source task) can beneficially inform the approximation of a function f_t in D_t (target task) [28].

Compared to the transfer learning methods mentioned earlier, our framework includes the transformer architecture [29], designed for pattern extraction and learning from symbolic sequences. Figure 1 illustrates the key components involved in the process. The initial development for approximating closure corrections (source task) commonly starts on less complex or computationally less expensive problems (1). This optimization process yields not a single but a diverse set of models with small cost function values. The dataset comprising the expressions for training the language model is obtained by optimizing $\mathbf{a}_{ij}$ for a square duct case with $Re_\tau = 180$ [30] and a periodic hill case with $Re = 10595$ [31]. These cases exhibit phenomena that cannot be handled accurately using the linear eddy viscosity hypothesis due to the occurrence of non-equilibrium turbulence. Within the duct, the phenomenon manifests as two counter-rotating vortices oriented in the corners of the channel perpendicular to the streamwise direction. In the periodic hill case, the flow encounters acceleration over the windward side and hill crest, followed by an expansion on the leeward side, which induces an adverse pressure gradient that leads to flow separation and recirculation. Standard models, such as the $k-\omega$ SST, overpredict the size of the recirculation zone and predict the reattachment to occur further downstream. Studies such as [5, 7] have already demonstrated that fitting $\mathbf{a}_{ij}$ can significantly enhance the performance of the standard $k-\omega$ SST. Consequently, we set up our training data by employing the CFD-driven variant of GEP with the particular fitness function and parametrization as given in [28].

From the explored expressions, a subset is selected satisfying the criterion of $< 1.4 \times \min(fitness)$ to train two distinct language models (identical hyperparameter configuration [28]). The first one is trained utilizing exclusively samples from the duct optimization, denoted as the training state 'duct'. For the second model, the training set is extended to use samples from the optimization of $\mathbf{a}_{ij}$ for both duct and periodic hill cases, referred to as training state 'duct & ph' (2). For the target task of generating improved initial solutions (for complex problems), the framework queries the trained transformer (3) and infers the augmented solution into the GEP optimization cycle (4). Here, we conduct the optimization of $\mathbf{a}_{ij}$ for a serpentine duct with $Re = 3.6 \cdot 10^6$, referred to as 'Military Engine Intake Research Duct (MEIRD)' [32, 33], where details related to the GEP optimization can be found [34]. Due to high computational costs to explore a_{ij} for the MEIRD case, the population size was reduced to 20 and the number of generations to five, and we augment 60% of the start population.

Considering the architecture of the transformer, the training, and the sampling mechanism, three adaptions have been made. These include:

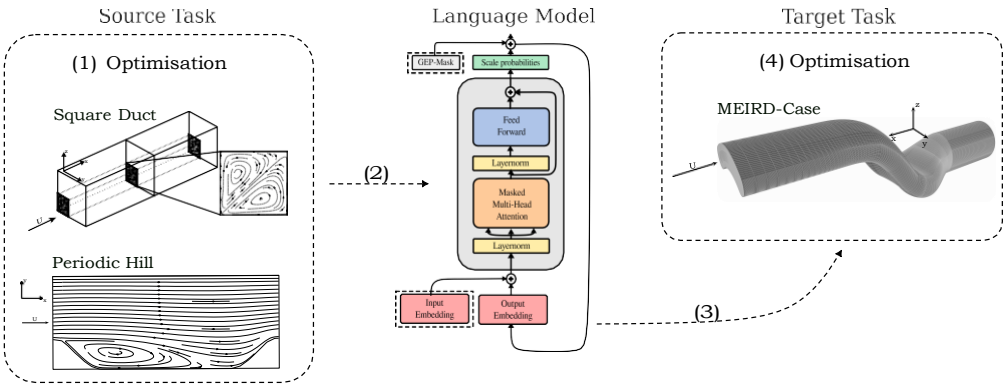

Figure 1: Figure of the language model-based transfer learning methodology. The process comprises: (1) optimization of non-linear corrections a_{ij} for the Reynolds stress for computationally less expensive test cases (duct case & periodic hill); (2) training of the language model with equations from sampled flow field velocity data; (3) augmentation of the initial population through querying the transformer; and (4) subsequent optimization for the target case. Components surrounded by dashed boxes indicate specific adaptations for turbulence modeling.

- Tokenization: This process converts the linearized phenotype into a vector of integer values, which in the context of symbolic regression [14, 15] results in a mapping on character level. To steer the prediction according to the logical segments given by the representation of the expression utilized by the GEP, we add control tokens for segmenting the head and tail sections.
- Creation of the input embedding: The input embedding assigns a vector of float values to each token, further complemented by position information. To relate the information of a flow case, we employ the PointNet [35] (similar to [15]) and infer the information via cross-attention.
- Manipulation of the output probabilities: Each step (t) of the autoregressive prediction produces a probability vector $\mathbf{p}$, enabling the derivation of the next symbol. These predictions yield the possibility of being misguided despite high data quality or extended training cycles, leading to highly nested or invalid expressions. To ensure validity, we add a vector to the i-th output, which is calculated by the encoding scheme of the genotype representation, particularly the head length (H_{len}):

$$\hat{\mathbf{p}}_i = \mathbf{p}_i + \begin{bmatrix} 0 & 0 & \dots & 0 \end{bmatrix}^T \quad \text{for } t < H_{\text{len}}, \qquad \hat{\mathbf{p}}_i = \mathbf{p}_i + \begin{bmatrix} -\infty & -\infty & \dots & 0 \end{bmatrix}^T \quad \text{for } t \geq H_{\text{len}}.$$

3. Results & Discussion

This section presents the performance of the GEP framework with an augmented start population from the trained language models across 10 trials, focusing on the predictive capabilities impacting the pressure distribution along the centerline.

Figure 2 presents the centerline pressure distribution, normalized by total pressure at the inlet, along the streamwise (x) direction, comparing experimental measurements [33] against the results obtained from simulations employing the different GEP techniques (solid lines) as well as the performance of the baseline $k-\omega$ SST [24] (dashed line). The shaded area highlights the performance of a small subset of expressions yielding the best fitness (the best five of each population). The $k-\omega$ SST model shows reasonable agreement with experimental data, yielding minor discrepancies ($error \leq 3\%$) at the lower (LW) and upper wall (UW) up to $x_{rel} \approx 0.65$. Beyond this point, the model exhibits significant deviations, particularly overestimating the size of the separation bubble, resulting in a larger pressure drop ($\approx 10\%$) at the upper wall (UW) and showing delayed pressure recovery ($x_{rel} \approx 0.75$ versus experimental $x_{rel} \approx 0.72$). The best model obtained from the standard GEP (figure 2, first row) utilizing the random initialization achieves slight improvements, reducing the deviation to 0.7% and mitigating

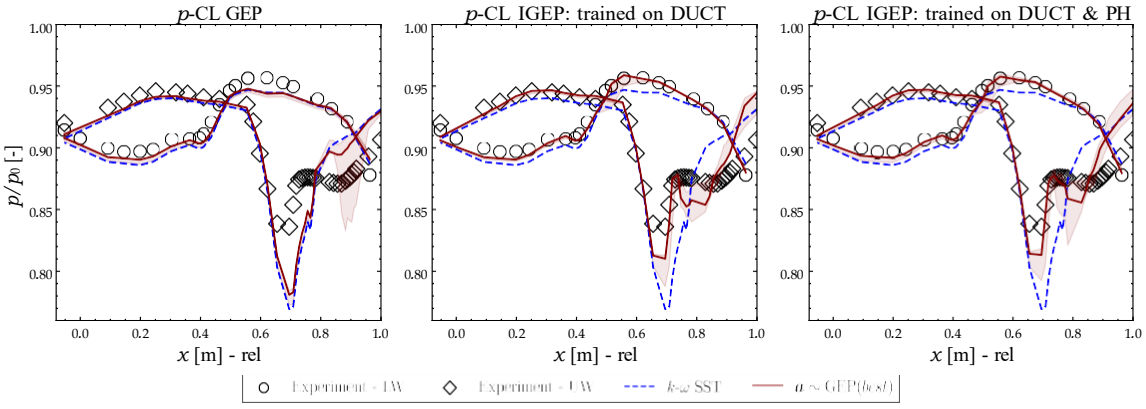

Figure 2: Pressure along the centerline (CL) of the 'MEIRD' normalized by the total pressure at the inlet. Experimental measurements [33] from the lower wall (∘) and the upper wall (◊) are compared with the $k-\omega$ SST [24] and corrections obtained from three methods. These include the standard GEP (left), the IGEP trained on candidate solutions from the square duct, and the IGEP (middle), which additionally incorporates samples from the periodic hill case (right). Shaded regions indicate the range of solutions from top-performing candidates within each population.

the pressure overshoot by 1.4%. Apart from another pressure drop ($0.85 - 0.95$), the confidence of the explored expressions aligns closely with the LEVM.

Augmenting initial solutions with duct-trained models significantly enhances performance, even with limited optimization generations. The pressure drop deviation at UW reduces from 10% to approximately 1.7%, with improved prediction of the separation onset and reattachment. The model exhibits increased accuracy at UW from $x_{rel} \approx 0.18$ to $x_{rel} \approx 0.65$, though the proportion of the best is showing uncertainty in four key regions: the LW-UW pressure crossover ($x_{rel} \approx 0.45$), the UW pressure drop region ($x_{rel} \approx 0.65$-0.75), post-recovery ($x_{rel} \approx 0.76$-0.83), and near the outlet ($x_{rel} \geq 0.93$).

Further improvement is achieved by incorporating expressions from the periodic hill optimization into the training set, particularly enhancing the pressure distribution at LW from $x_{rel} \approx 0.25$ and mitigating the pressure drop after the recovery (UW $x_{rel} \approx 0.72$-0.75).

4. Conclusion

In this study, we introduced the application of a language model-based transfer learning strategy to symbolic regression-based turbulence modeling, effectively combining the complementary strengths of pre-trained approaches with the iterative search capabilities of the GEP.

The framework was tested on a complex serpentine duct case, where it successfully supported the exploration process while surpassing the standard GEP. Moreover, the tests indicate the importance of the training data composition, demonstrating that diverse but physically appropriate training samples yield further potential to accelerate the exploration.

Future work will include the latest architectural enhancements, such as a mixture of expert layers or a second training stage involving reinforcement training.

Acknowledgments

This research was supported by The University of Melbourne's Research Computing Services and the Petascale Campus Initiative. The work was supported by a Melbourne Research Scholarship provided by the University of Melbourne.

References

[1] K. Duraisamy, G. Iaccarino, and H. Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51, 2019. doi: 10.1146/annurev-fluid-010518-040547.

[2] A. Cary, J. Chawner, E. Duque, W. Gropp, N. Wukie, E. Nielsen, and B. Smith. CFD vision 2030 roadmap progress and tech-

nology highlights for 2022. Technical report, AIAA CFD Vision 2030 Integration Committee, December 2022. URL https://www.cfd2030.com/report/CFD_Vision_2030_Roadmap_Update_2022.pdf.

[3] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[4] J. Weatheritt and R. D. Sandberg. A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. *Journal of Computational Physics*, 325:22–37, 2016.

[5] M. Schmelzer, R. P. Dwight, and P. Cinnella. Discovery of algebraic Reynolds-stress models using sparse symbolic regression. *Flow, Turbulence and Combustion*, March 2020. doi: 10.1007/s10494-019-00089-x.

[6] H. D. Akolekar, F. Waschkowski, Y. Zhao, R. Pacciani, and R. D. Sandberg. Transition modeling for low pressure turbines using computational fluid dynamics driven machine learning. *Energies*, 14(15), 2021.

[7] F. Waschkowski, Y. Zhao, R. D. Sandberg, and J. Klewicki. Multi-objective CFD-driven development of coupled turbulence closure models. *Journal of Computational Physics*, 452:110922, 2022. doi: 10.1016/j.jcp.2021.110922.

[8] J. Wu, H. Xiao, R. Sun, and Q. Wang. Reynolds-averaged navier–stokes equations with explicit data-driven reynolds stress closure can be ill-conditioned. *Journal of Fluid Mechanics*, 869:553–586, 2019. doi: 10.1017/jfm.2019.205.

[9] Y. Zhao, H. D. Akolekar, J. Weatheritt, V. Michelassi, and R. D. Sandberg. Rans turbulence model development using cfd-driven machine learning. *Journal of Computational Physics*, 411:109413, 2020.

[10] Y. Fang, Y. Zhao, F. Waschkowski, A. S. H. Ooi, and R. D. Sandberg. Toward more general turbulence models via multicase computational-fluid-dynamics-driven training. *AIAA Journal*, 61(5):2100–2115, 2023. doi: 10.2514/1.J062572.

[11] S. S. Girimaji. Turbulence closure modeling with machine learning: a foundational physics perspective. *New Journal of Physics*, 26(7): 071201, 2024. doi: 10.1088/1367-2630/ad6689.

[12] P. R. Spalart. Philosophies and fallacies in turbulence modeling. *Progress in Aerospace Sciences*, 74:1–15, 2015. doi: 10.1016/j.paerosci.2014.12.004.

[13] L. Biggio, T. Bendinelli, A. Lucchi, and G. Parascandolo. A seq2seq approach to symbolic regression, 2020.

[14] P.-A. Kamienny, S. D'Ascoli, G. Lample, and F. Charton. End-to-end symbolic regression with transformers, April 2022.

[15] M. Valipour, B. You, M. Panju, and A. Ghodsi. Symbolicgpt: A generative transformer model for symbolic regression. *arXiv preprint arXiv:2106.14131*, 2021.

[16] M. Vastl, J. Kulha´nek, J. Kubal´ık, E. Derner, and R. Babus˘ka. Symformer: End-to-end symbolic regression using transformer-based architecture. *IEEE Access*, 2024.

[17] M. Merler, K. Haitsiukevich, N. Dainese, and P. Marttinen. In-context symbolic regression: Leveraging large language models for function discovery. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 427–444, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-srw.49. URL https://aclanthology.org/2024.acl-srw.49/.

[18] Y. Li, W. Li, L. Yu, M. Wu, J. Liu, W. Li, S. Wei, and Y. Deng. Mllm-sr: Conversational symbolic regression base multi-modal large language models. *arXiv preprint arXiv:2406.05410*, 2024.

[19] M. Du, Y. Chen, Z. Wang, L. Nie, and D. Zhang. Large language models for automatic equation discovery of nonlinear dynamics. *Physics of Fluids*, 36(9):097121, 2024. doi: 10.1063/5.0224297.

[20] T. Ye, Z. Xu, Y. Li, and Z. Allen-Zhu. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. *arXiv preprint arXiv:2407.20311*, 2024.

[21] C. Ferreira. *Gene Expression Programming: Mathematical Modeling by an Artificial Intelligence*. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2006.

[22] J. Weatheritt and R. D. Sandberg. The development of algebraic stress models using a novel evolutionary algorithm. *International Journal of Heat and Fluid Flow*, 68:298–318, 2017.

[23] S. B. Pope. A more general effective-viscosity hypothesis. *Journal of Fluid Mechanics*, 72(2):331–340, 1975.

[24] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, 1994.

[25] M. P. Brenner, J. D. Eldredge, and J. B. Freund. Perspective on machine learning for advancing fluid mechanics. *Physical Review Fluids*, 4:100501, 2019.

[26] C. Chen, X. Jin, and H. Li. Discovering governing equations by machine learning implemented with invariance, 2022.

[27] T. T. H. Dinh, T. H. Chu, and Q. U. Nguyen. Transfer learning in genetic programming. In *2015 IEEE Congress on Evolutionary Computation*, pages 1145–1151, 2015.

[28] M. Reissmann, Y. Fang, S. H. Ooi, and R. D. Sandberg. Accelerating evolutionary exploration through language model-based transfer learning. *arXiv preprint arXiv:2407.20311*, 2024.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-December:5999–6009, 2017.

[30] R. Vinuesa, P. Schlatter, and H. M. Nagib. Secondary flow in turbulent ducts with increasing aspect ratio. *Physical Review Fluids*, 3: 054606, 2018. doi: 10.1103/PhysRevFluids.3.054606.

[31] M. Breuer, N. Peller, C. Rapp, and M. Manhart. Flow over periodic hills – Numerical and experimental study in a wide range of Reynolds numbers. *Computers & Fluids*, 38(2):433–457, 2009. doi: 10.1016/j.compfluid.2008.05.002.

[32] J. P. Haug, R. P. M. Rademakers, M. Krummenauer, and R. Niehuis. Validation of RANS simulations of the flow in a short highly bent intake duct. In *13th European Conference on Turbomachinery Fluid Dynamics and Thermodynamics*, 2019.

[33] R. P. M. Rademakers, J. P. Haug, R. Niehuis, and M. Sto¨ßel. Design and development of a military engine inlet research duct. In *30th Congress of the International Council of the Aeronautical Sciences*, DCC, Daejeon, Korea, 2016.

[34] M. C. Hein, M. Reissmann, J. A. Scheibel, M. Sto¨ßel, R. D. Sandberg, and D. Kozulovic. Numerical simulation of a highly bent intake duct using machine learning algorithms to close the RANS equations. *Proceedings of Global Power & Propulsion Society*, 2024.

[35] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*, pages 77–85, 2017.