An Advanced Surrogate Model Approach for Enhancing Fluid Dynamics Simulations

Shubham Kavane, Kajol Kulkarni, Harald Köstler
Chair of System Simulation, Friedrich Alexander Universität Erlangen-Nürnberg
Erlangen, Germany
e-mail: {shubham.kavane, kajol.kulkarni, harald.koestler}@fau.de

***Abstract*—The increasing complexity and computational demands of 3D fluid dynamics simulations highlight the need for advanced surrogate models that strike a balance between predictive accuracy, computational efficiency, and convergence time. Traditional Computational Fluid Dynamics (CFD) methods, while highly accurate, are often resource-intensive and time-consuming. This research presents advanced U-Net-based surrogate models for 3D fluid flow prediction, aiming to achieve faster convergence and more efficient resource utilization while retaining competitive accuracy relative to traditional CFD solvers. We developed a U-Net model featuring an improved architecture utilizing an advanced attention mechanism known as the Convolution Block Attention mechanism. Considering the high computational demands, the model was trained using multiple GPUs, incorporating both model and data parallelism techniques. The model's capability was evaluated through overfitting experiments, where it was trained on a limited dataset to assess its ability to accurately replicate true labels. These findings highlight the promise of advanced surrogate models as a viable alternative to traditional CFD methods, providing faster solutions and reduced computational costs with comparable accuracy. Future research will focus on evaluating the current advanced U-Net model, trained on an extensive dataset of 10,000 samples, against Fourier Neural Operators and traditional CFD solvers in terms of training time, accuracy, and resource utilization, including energy consumption.**

Keywords-Surrogate Models; Computational Fluid Dynamics (CFD); U-Net; Fourier Neural Operators; Model Parallelism;

I. Introduction

The rapid advancements in Machine Learning (ML) and Deep Learning (DL) have transformed various fields by providing innovative solutions to complex problems once considered unsolvable. These technologies have revolutionized applications across various scientific domains [1]. Notably, in fluid dynamics, ML and DL have introduced groundbreaking methods that enhance our ability to understand and tackle intricate challenges, underscoring their profound and far-reaching impact [2].

Traditional fluid flow analysis relies on the Navier-Stokes equations (NSE), which, despite their strong theoretical foundation, are time-consuming and computationally intensive, particularly for complex scenarios. The limited parallelizability and iterative nature of algorithms for solving partial differential equations (PDEs) further complicate achieving convergence and efficient parallelization in real-world, non-convex problems [3]. Recent developments in fluid flow prediction have increasingly shifted toward data-driven methodologies, with deep learning-based surrogate models becoming a robust alternative to CFD simulations. These models are particularly effective in predicting complex, nonlinear fluid behavior across diverse Reynolds numbers, geometries, and flow conditions. They achieve faster convergence and enhanced computational efficiency with minimal compromise on accuracy. By leveraging sufficiently large datasets, surrogate models can recognize patterns without relying on explicit physical laws, making them particularly valuable for modeling turbulent, unsteady, or multiphase flows where traditional methods struggle. These innovations enhance fluid dynamics research and enable more sophisticated and efficient solutions for critical engineering applications [4].

The integration of data-driven surrogate models with deep learning has significantly enhanced both the precision and efficiency of fluid dynamics simulations. Nonetheless, challenges such as model generalization and the handling of high-resolution, large datasets persist, as these models must reliably predict outcomes under novel or previously unseen conditions across a diverse range of fluid flow scenarios. Achieving this level of adaptability requires sophisticated model architectures capable of accurately capturing the intricate flow dynamics observed in real-world conditions. As a result, these models often become highly complex, with millions of trainable parameters, necessitating the use of multiple GPUs to optimize training time and computational resources effectively [5].

we propose an advanced U-Net-based surrogate model specifically designed to predict complex fluid dynamics scenarios. We have employed highly optimized multi-GPU training strategies, such as DeepSpeed ZeRO, to maximize computational efficiency. The primary research goals and objectives of this work are as follows:

- Develop an advanced U-Net-based surrogate model and train it on a multi-GPU setup using data and model parallelism techniques to predict complex flow scenarios.
- Compare the performance of the U-Net model with Fourier Neural Operators when trained on a large dataset of 10,000 samples.
- Evaluate the advanced surrogate model against traditional CFD solvers by assessing convergence time, accuracy, and resource utilization, including energy consumption.

The rest of the paper is organized as follows: Section 2 explains the training data generation process and preprocessing techniques. Section 3 offers an overview of the standard U-Net model, emphasizing the enhancements in the advanced U-Net. Section 4 discusses the necessity of multi-GPU training and compares the model's results. Finally, Section 5 concludes the paper and outlines future research directions.

ISBN: 978-1-68558-192-3

II. Training Data Generation:

Our study adopts a comprehensive approach to generate generalized datasets for CFD applications, emphasizing the need for geometric and positional diversity as underscored by [6]. Utilizing Python 3.8 and CadQuery 2.1, we employed a custom-developed Python script to generate a diverse array of three-dimensional shapes within a rectangular channel domain [7]. The shapes include cubes, cuboids, cones, cylinders, spheres, torus, and wedges, varying in size and orientation to create a versatile dataset suitable for a broad range of CFD studies.

To delineate regions within and around these geometries, we used signed distance functions, which provide spatial context for the geometries to the network. For simulations, we utilized the in-house developed WalBerla software, which is based on the lattice Boltzmann method (LBM) to generate true labels for supervised training. The automation script, coupled with the Fritz HPC clusters, facilitates the parallel generation of numerous simulations across multiple cores.

For data preparation, we applied rigorous preprocessing techniques, including standard scaling and min-max normalization. We observed that standard scaling was more effective for our application compared to min-max normalization. The signed distance functions will be used as inputs for the U-Net model, while the WalBerla simulations will provide the true labels for supervised learning, ensuring precise and efficient model training. In total, we have generated 10,000 training samples for extensive training of the advanced U-Net model.

For simulations, a D3Q27 lattice model, employing a cumulant collision operator, was utilized for the simulations. These simulations were conducted within a domain of size 2048 x 512 x 512. The Reynolds number was varied from 50 to 10,500.

III. Methodology

In the following section, we provide a brief overview of the standard U-Net, outlining its key components, and explore how the Advanced U-Net extends these foundations with significant enhancements.

A. U-Net

The U-Net architecture, initially designed for biomedical image segmentation, is known for its effectiveness in complex tasks due to its U-shaped structure with contracting and expanding paths [8]. It has since been enhanced and adapted, including to 3D volumes and various fields like fluid flow prediction [9], demonstrating its broad versatility and impact.

Figure 1 illustrates the standard U-Net architecture, distinguished by its unique U-shaped configuration that includes an encoder (contracting path), a bottleneck, and a decoder (expanding path). This innovative design is notable for employing an extensive number of feature channels in the upsampling section, facilitating the propagation of contextual information to higher resolution layers.

The U-Net architecture comprises three main sections: the encoder, the bottleneck, and the decoder.

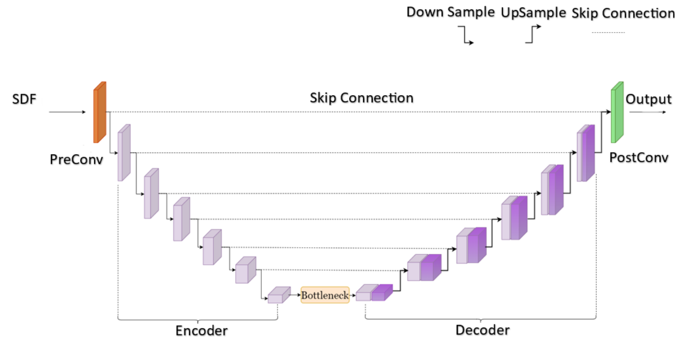

Figure 1. Schematic diagram of standard U-Net architecture

- Encoder: This initial stage, featuring convolutional layers, max pooling, activation functions, and batch normalization, reduces spatial dimensions while enhancing feature depth, and capturing critical and abstract features.
- Bottleneck: At the network's lowest resolution, the bottleneck connects the encoder and decoder, using multiple convolutional layers to handle abstracted data and integrate complex contextual information.
- Decoder: This stage reconstructs high-resolution data by increasing spatial resolution and reducing feature channels. It includes skip connections that merge upsampled outputs with encoder feature maps, reintroducing spatial details for accurate predictions.

B. Advanced U-Net Architecture and Its Component

In the following section, only the improved features over standards U-Net architecture are highlighted.

1) Repeating selected Encoder Layers without down sampling: The use of repeated encoder layers without down sampling in advanced U-Net architectures offers several significant benefits. Firstly, it enhances feature extraction by allowing the network to iteratively process and refine feature information. This iterative approach helps capture both low-level details and high-level abstractions, resulting in a more nuanced and accurate representation of the input data. Secondly, maintaining consistent spatial dimensions and input/output channels throughout these layers preserves important spatial details. This preservation is crucial for accurately representing the structure and features of the input, which is essential for tasks that require detailed spatial understanding. Additionally, the repeated encoder layers improve the network's contextual understanding by enabling it to build a more comprehensive view of complex features and relationships within the data. This leads to more precise and detailed interpretations, enhancing the overall effectiveness of the network.

2) Increasing the number of input layers and numbers of channels: Incorporating additional input layers and increasing the number of channels in a convolutional neural network significantly enhances its ability to process complex data. This increased depth enables the network to capture and analyze finer details, leading to a more sophisticated understanding of intricate features. By scaling the number of channels in encoder layers up to 2048 or even 4096, the network achieves a hierarchical feature representation. Initial layers focus on basic patterns, while deeper layers with more channels interpret these patterns in nuanced contexts, revealing detailed structures and semantic information. This hierarchical approach is particularly advantageous in applications such as high Reynolds number flows, where understanding high-frequency patterns and complex interactions is crucial.

However, expanding the network's depth and channel count introduces challenges, such as an increased risk of overfitting and higher demands on computational resources and memory during training. Managing these complexities requires careful balancing of network architecture to optimize both performance and practical feasibility.

3) Varying Kernel Sizes: The selection of kernel size is crucial in convolutional neural networks (CNNs) for effective feature extraction. Smaller kernels, such as 3×3×3, are adept at capturing fine details, such as small eddies and turbulent flow scales, which is essential for accurately predicting intricate flow scenarios. Medium-sized kernels, such as 5×5×5 and 7×7×7, strike a balance by capturing a wider range of patterns and contextual information, thereby enhancing the model's versatility. Larger kernels, like 9×9×9, are employed to cover more extensive portions of the input, enabling the identification of large-scale patterns and structural elements while maintaining global consistency in predictions. By incorporating a range of kernel sizes, CNNs can effectively capture both detailed and broad features, which is particularly advantageous for U-Net models in performing comprehensive data analysis. This varied approach enhances the network's capability to interpret complex input data across multiple scales.

4) Use of residual connection in the encoder and decoder Block: Residual connections play a crucial role in deep networks by mitigating the vanishing gradient problem, which can impede training by causing gradients to diminish through multiple layers. They preserve information by maintaining a continuous flow across layers, merging initial inputs with subsequent outputs to retain essential features. This capability enhances model convergence, as residual connections enable more effective gradient flow and faster convergence. Additionally, these connections are vital for constructing deep architectures, allowing networks to learn complex patterns without the issues typically associated with deeper models.

5) Use of advanced attentions Mechanism: Convolution Block Attention module: The Convolutional Block Attention Module (CBAM) [10] significantly enhances neural networks by focusing attention sequentially on both channel and spatial dimensions. First, the Channel Attention module compresses spatial information into a channel descriptor using global average pooling, which highlights important features and applies a ReLU activation followed by sigmoid to generate a channel attention mask. This mask refines feature importance on a channel-by-channel basis. Subsequently, the Spatial Attention module identifies critical spatial regions by pooling features across channels and combining them with a convolutional layer to create a spatial attention map, which directs the network's focus to essential areas. This dual attention mechanism enables CBAM to selectively emphasize vital features, improving the network's ability to represent complex data and enhance overall performance.

IV. Multiple GPU training of advanced U-Net model

The enhanced model iteration offers a significant improvement over the standard U-Net by incorporating additional encoder layers and expanding the number of channels, leading to enhanced feature extraction and prediction accuracy. While retaining the core methodologies of the traditional U-Net, this iteration increases both depth and analytical capability. It integrates the Convolutional Block Attention Mechanism (CBAM) and introduces residual connections within and between the encoder and decoder blocks, optimizing data processing and learning efficiency. As a result, the number of trainable parameters has increased from 80 million in the standard U-Net to 511 million in advanced U-Net, contributing to the model's complexity [11] [12].

Due to these advancements, the heightened computational demands pose challenges for training on a single GPU. The increased model complexity necessitates substantial processing power and optimal use of high-performance computing (HPC) resources. To address these challenges and enhance training efficiency, the deployment of multiple GPUs is essential. Employing PyTorch's Distributed Data Parallel (DDP) alongside DeepSpeed's ZeRO-2 [13] model parallelism strategy has facilitated effective parallel processing, resulting in a significant reduction in training time—approximately 4-5 times faster per epoch. This approach has also been instrumental in identifying the optimal resources required for training the advanced U-Net model.

A. Results and Analysis of Advanced U-Net Model:

1) Model Capacity Evaluation through Overfitting: In deep learning, particularly for complex tasks like predicting fluid velocity, assessing a model's capabilities is essential before engaging in extensive training. One effective method is to test the model's ability to overfit on a small, representative dataset. This approach helps determine if the model can accurately capture complex data patterns by minimizing loss on this subset. For evaluating a U-Net architecture, the model is deliberately overfitted on a carefully selected small dataset to drive the loss near zero compared to true labels, indicating its capability to replicate intricate details accurately. Successful overfitting, evidenced by significantly reduced loss, suggests that the model can encapsulate detailed flow dynamics. If the model fails to achieve satisfactory loss reduction, it may require architectural enhancements.

In this study, we trained both the standard and an advanced U-Net model for 500 epochs on a relatively small dataset consisting of 16 samples. The performance of the models was evaluated using the L1 loss, which measures the absolute difference between the predicted labels and the ground truth.

 ISBN: 978-1-68558-192-3

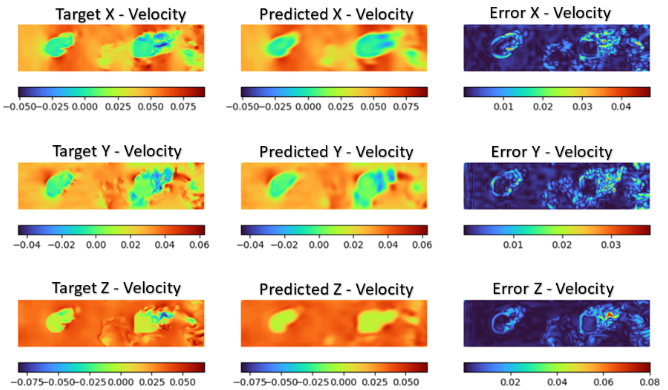

Figure 2. Comparison of Target Velocity, Predicted Velocity, and Absolute Error for each component, based on a Model trained with 16 samples for Standard U-Net Model

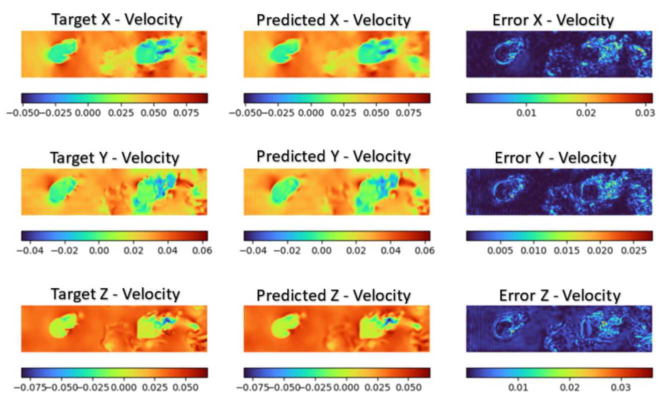

Figure 3. Comparison of Target Velocity, Predicted Velocity, and Absolute Error for each component, based on a Model trained with 16 samples for Advanced U-Net Model

The standard U-Net model yielded an L1 loss of 0.32, whereas the advanced U-Net achieved a markedly lower L1 loss of 0.09.

In terms of predictions, the standard U-Net model exhibited some resemblance to the target velocity fields but was unable to capture the finer details of the flow as seen in Figure 2. Its predictions for the X, Y, and Z velocity components appeared overly smoothed, with regions of high-velocity present but not sharply defined. For example, in the predicted X-velocity field, the regions with higher velocity values are visible, but their contours are not well defined, resulting in an oversimplified representation of the complex flow behavior. This lack of precision indicates the base model's limitations in capturing intricate velocity variations, particularly in areas where the flow is rapidly changing.

Conversely, the advanced U-Net model produced predictions that were much more aligned with the ground truth. Its predictions for the X, Y, and Z components were sharper and better represented the spatial complexity of the flow, especially in regions with high-velocity magnitudes as seen in Figure 3. Notably, the X-velocity predictions of the advanced model show a much closer match to the target, especially in areas where the flow exhibits more complex behavior. This highlights the advanced model's superior ability to capture finer details and dynamic variations in the velocity fields.

These results suggest that the modifications introduced in the advanced architecture—such as increasing the number of channels and encoder layers, as well as integrating an advanced attention mechanism—contribute significantly to its improved performance. Further evidence of the advanced U-Net's superior predictive capability is illustrated in the velocity plots comparing true labels to predictions, as presented in Figures 2 and 3.

Evaluating a model's capacity through overfitting on a small dataset effectively assesses its initial ability to predict flow with high accuracy. However, this method does not evaluate the model's performance on unseen data. Therefore, after confirming the model's capacity, it is crucial to apply regularization techniques during training on larger, more comprehensive datasets to prevent overfitting. This approach helps maintain the model's appropriate level of complexity and optimizes its effectiveness, thereby avoiding the inefficiencies associated with overly complex models. Additionally, to ensure robust performance across various flow scenarios, including both laminar and turbulent conditions, the model must be trained on an extensive dataset that encompasses all these variations. Only a sufficiently large and diverse training dataset can enable the model to learn and generalize effectively across different flow patterns.

2) Evaluation of the Advanced U-Net Model with 1000 samples: We trained both the standard U-Net and the advanced U-Net models on a relatively large dataset comprising 1,000 samples to evaluate their performance under more realistic conditions. While the standard U-Net model was able to make predictions of the velocity fields, the quality of these predictions was comparatively poor when evaluated against the advanced U-Net model as seen in Figure 4 and 5. The standard model struggled to accurately capture the flow dynamics, particularly in regions with more complex or high-velocity patterns, leading to oversimplified predictions that lacked detail and precision.

In contrast, the advanced U-Net model showed improvements in its predictive capabilities. However, its performance was still below what might be expected given its architectural advantages. Although it did not overfit the data, the predictions were not as sharp or detailed as those observed in the overfitting experiments, where the model had demonstrated the ability to perfectly capture the flow patterns on a smaller dataset. This performance gap suggests that the advanced model, while more powerful, requires further training on an even more extensive and diverse dataset to fully realize its predictive potential and generalize well to unseen data.

One of the key factors contributing to these results is that the model needs to encounter a wide range of scenarios during training in order to develop a more robust understanding of flow dynamics. With a limited number of samples, even though 1,000 represents a substantial increase over smaller datasets,

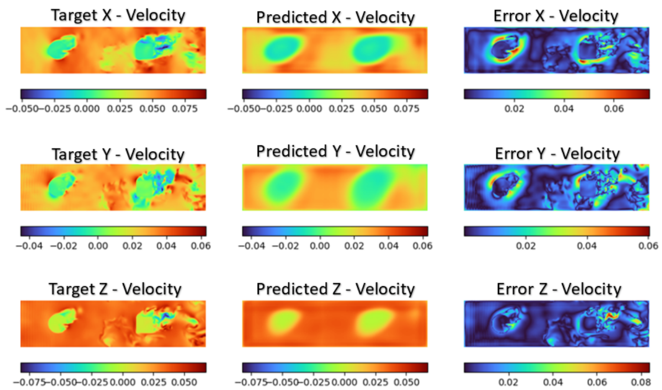

Figure 4. Comparison of Target Velocity, Predicted Velocity, and Absolute Error for each component, based on a Model trained with 800 samples and validated against 200 samples for standard U-Net Model

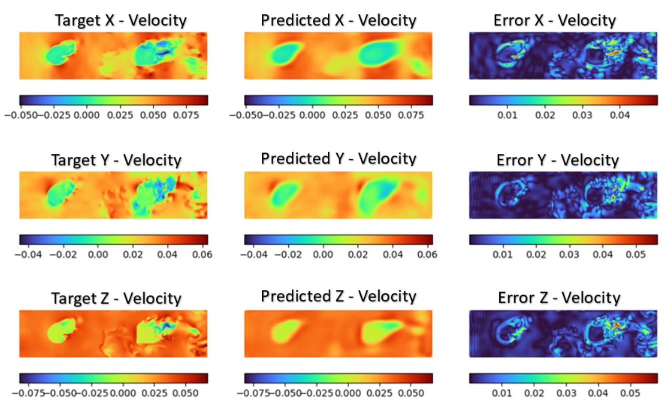

Figure 5. Comparison of Target Velocity, Predicted Velocity, and Absolute Error for each component, based on a Model trained with 800 samples and validated against 200 samples for Advanced U-Net Model

the model has not yet been exposed to all possible variations in the flow patterns. This lack of comprehensive exposure results in suboptimal generalization, especially when confronted with new, unseen data.

While the model began to learn the flow dynamics, its prediction accuracy was notably lower compared to the performance observed during the overfitting experiments. This discrepancy highlights a gap between the model's ability to fit a small dataset and its performance on a larger, more diverse set. Figure 5 illustrates the model's predictions when trained on the 1,000 samples.

V. Conclusion and Future Work

In this work, we developed an advanced U-Net architecture for fluid flow prediction in complex geometries and domains. We conducted a comparative analysis between the advanced U-Net and the standard U-Net, focusing on overfitting experiments and training performance on a dataset of 1,000 samples. The architectural improvements in the advanced U-Net enabled it to capture intricate flow patterns more effectively, resulting in a 71% improvement in overfitting performance compared to the standard U-Net. Additionally, while neither model performed as expected on the 1,000-sample dataset, the advanced U-Net demonstrated superior accuracy. These results underscore the need for further training on a larger dataset to fully realize the potential of the advanced U-Net model.

While working with the advanced U-Net model, we encountered several challenges. One significant issue is that, even though the model is adept at predicting complex fluid flows, it struggles to generalize effectively when trained on larger datasets. This difficulty underscores the importance of implementing careful and specialized training strategies to ensure accurate performance across extensive datasets.

Another issue arises when an object within a channel is relatively small, as the flow variations become concentrated around the object. In contrast, the larger portions of the channel exhibit minimal variation and are relatively simpler. This uneven distribution of flow complexity can complicate the evaluation of model performance. Specifically, using traditional loss metrics like the L1 error can produce misleading results. The L1 error might indicate a falsely reduced error if the model accurately predicts the simpler, less complex regions of the channel while failing to capture the intricate flow patterns near the object. This is because the accurate predictions in the less complex regions can overshadow the errors in the more complex regions near the object.

To mitigate this problem, one approach is to modify the error calculation by incorporating weights that emphasize the accuracy of predictions near the object. By prioritizing errors in these critical regions, this weighted error calculation helps to avoid misleadingly low error values and provides a more accurate assessment of the model's performance around complex areas.

In future we plan to extent the work with following aspects,

1) Training on a Larger Dataset: To thoroughly assess the generalization capability of the advanced U-Net model, we plan to train it on progressively larger datasets, scaling up to 10,000 samples. The aim of this step is to systematically test the model's ability to learn complex flow dynamics when exposed to a broader range of scenarios. A larger dataset will help mitigate the potential for overfitting and allow the model to generalize better to unseen cases.

In addition to this, we will conduct an ablation study to analyze the contributions of different architectural elements (e.g., the number of encoder layers, attention mechanisms, and increased channels). This will allow us to determine the importance of each feature and guide further optimization. Specifically, we will remove or alter these components one at a time to observe their direct impact on prediction accuracy, training time, and error rates. Such an approach will provide insight into which elements are critical for performance and which might be redundant or unnecessary.

2) Performance Comparison and Hybrid Model Development: We will conduct a detailed performance comparison between the advanced U-Net and the Fourier Neural Operator (FNO). The rationale behind comparing these models stems from their fundamentally different architectures: the U-Net excels in capturing local spatial features due to its convolutional nature, while the FNO is designed to efficiently model global patterns using Fourier transforms. This comparison will focus on aspects such as:

- Accuracy in capturing fine flow structures (especially in complex, high-velocity regions).
- Computational efficiency, particularly in terms of training time and resource consumption.
- Scalability with respect to dataset size and prediction time for large-scale problems.

Based on the insights from this comparison, we propose the development of a hybrid model that integrates the strengths of both architectures. The hybrid model will leverage U-Net's ability to accurately capture local features with the FNO's capacity to model large-scale, global flow dynamics. Specifically, we envision an architecture that uses U-Net layers for feature extraction at finer scales, followed by FNO layers to capture overarching patterns and relationships. This approach should improve both the accuracy and efficiency of the predictions, especially in challenging fluid dynamics simulations.

3) Evaluation of Surrogate Models: We will evaluate the performance of the surrogate model (based on the advanced U-Net or the proposed hybrid model) against traditional CFD methods. This evaluation will focus on several key performance indicators:

- Accuracy: We will measure the difference in prediction accuracy between the surrogate model and CFD simulations, focusing on both average error and maximum error in critical flow regions.
- Convergence time: Surrogate models are expected to converge much faster than conventional CFD methods. We will document and compare convergence times, particularly in simulations requiring iterative solutions over complex domains.
- Computational resources: The analysis will include detailed assessments of the computational power required by each approach, such as CPU/GPU usage, memory consumption, and overall energy expenditure. The goal is to quantify the potential cost savings of using surrogate models.

In addition, we will explore real-time applications of the surrogate model in industrial settings, where rapid simulations are often required for optimization, design iteration, or operational decision-making. The ability of the surrogate model to provide high-fidelity predictions in a fraction of the time typically required by CFD will be a significant aspect of this evaluation.

Acknowledgment

Innovation Study FLOWGEN has received funding through the Inno4scale project, which is funded by the European High-Performance Computing Joint Undertaking (JU) under Grant Agreement No 101118139. The JU receives support from the European Union's Horizon Europe Programme.

References

[1] S. L. Brunton, J. N. Kutz, K Manohar, A. Y. Aravkin, K. Morgansen, J. Klemisch N. Goebel, J. Buttrick, J. Poskin, A. Blom-Schieber, T. Hogan, and D. McDonald 'Data-Driven Aerospace Engineering: Reframing the Industry with Machine Learning," ArXiv, vol.24, pp. 417-438, 2005.
[2] M. R. Bachute, and J. M. Subhedar, 'Autonomous Driving Architectures: Insights of Machine Learning and Deep Learning Algorithms," Machine Learning with Applications, vol. 6, pp.100164,issn 2666-8270, April 2024.
[3] I. I. Albarreal, M. C. Calzad, J. L. Cruz Soto and E. Fern, 'Time and space parallelization of the navier-stokes equations,"Computational & Applied Mathematics,vol 24, N. 3, pp. 417–438, 2005.
[4] O. Hiroto, and A. Takeshi, 'Prediction of steady flows passing fixed cylinders using deep learning,"Scientific Reports,vol 12, pp. 437-449, 2022.
[5] M. Telgarsky, 'Benefits of Depth in Neural Networks,"Annual Conference Computational Learning Theory, pp. 1517-1539, 2016.
[6] K. Otness, A. Gjoka, J. Bruna, D. Panozzo, B. Peherstorfer, T. Schneider, and D. Zorin, 'An Extensible Benchmark Suite for Learning to Simulate Physical Systems,"ArXiv,pp. 137-143, 2021.
[7] S. Jacob, M. Mrosek, and C. Othmer, 'Deep Learning for Real-Time Aerodynamic Evaluations of Arbitrary Vehicle Shapes,"SAE International Journal of Passenger Vehicle Systems,vol 15, pp. 77-90, March 2022.
[8] O. Ronneberger, P. Fischer, T. Brox, 'U-Net: Convolutional Networks for Biomedical Image Segmentation,"Medical Image Computing and Computer-Assisted Intervention – MICCAI ,vol 4, pp. 234-241, 2015.
[9] N. Thuerey, W. Konstantin and P. Lukas, 'Deep Learning Methods for Reynolds-Averaged Navier-Stokes Simulations of Airfoil Flows,"AIAA Journal,vol 58, pp. 25-36, 2020.
[10] S. Woo, J. Park, and J. Y. Lee, and I. S. Kweon1, 'CBAM: Convolutional Block Attention Module,"ECCV Journas, pp. 3-19, l2018.
[11] O. Hennigh, S. Narasimhan, M. A. Nabian, A. Subramaniam, and K. Tangsali , 'NVIDIA SimNet™: An AI-Accelerated Multi-Physics Simulation Framework, pp. 447-461, 2021.
[12] S. Li, Y. Zhao, R. Varma, O. Salpekar, P. Noordhuis, T. Li, A. Paszke, J. Smith, B. Vaughan, P Damania, and S. Chintala, 'PyTorch Distributed: Experiences on Accelerating Data Parallel Training,"Proceedings of the VLDB Endowment, Volume 13, Issue 12, pp. 3005-3018.
[13] S. Rajbhandari, J. Rasley, O. Ruwase, and Y. He, 'ZeRO: Memory Optimizations Toward Training Trillion Parameter Models,"ArXiv, Volume 13, Issue 12, pp. 1-16, 2020.