Indoor Computational Fluid Dynamics using Smart Proxy Models

Cormac McCumiskey supervised by Dr. Nima Gerami-Seresht

Abstract—**This study explores the prediction of velocity vector fields within an indoor environment through the implementation of Deep-Learning driven Smart Proxy Models (SPMs). The training data was collect through Computational Fluid Dynamics (CFD) and consisted of x,y,z positions, and velocities across a mesh within a 75 m^3 room equipped with a Heating, Ventilation, and Air Conditioning (HVAC) system. Preprocessing included creating spherical vectors to encode the position relative to the HVAC inlet and outlet vents and the six nearest walls, serving as proxies for numerical simulations. The proxies were then used as inputs to a basic Artificial Neural Network (ANN) to segment the velocity flow field into its respective flow regimes of stagnant, laminar, transition and turbulent flow to 98.2% accuracy. The proxies were then used as deep-learning models which were trained separately on each segment of data, upon testing, the models had a Mean Absolute Percentage Error (MAPE) of less than 5% for free-flow regions away from boundary layers. Notably, the inference time for predictions using SPMs were 0.8s on a work laptop with an intel core i7 processor. In comparison, the same data took five hours to generate on Durham Universities Hamilton 8 Supercomputer [1] which uses three compute nodes and 256GB of RAM. This presents a relative time saving of 99.996% and enables the simulation to be run on almost personal computer.**

I. Introduction

INDOOR environmental modeling plays a crucial role in several applications such as energy efficiency optimization, air quality assessment, and pathogenic transmission. While traditional techniques like CFD are widely used for predicting velocity flow fields, they often require domain expertise, high-performance computers, and long run times, particularly in larger indoor environments.

Despite the common misconception that pathogens primarily spread via surface contact, recent research by Zhang et al. [2] indicates that aerosol transmission is one thousand times more likely to spread Covid-19 than contact with surfaces. This highlights the need for a better understanding of aerosol dynamics in indoor spaces. The National Library of Medicine emphasises the potential of CFD in elucidating disease transmission mechanisms but notes a research gap in understanding how room conditions such as windows, doors, and ventilation rates affect pathogen spread [3]. During the COVID-19 pandemic, maintaining well ventilated public spaces became crucial for the reduction of pathogen concentrations. However, if a comprehensive grasp of airflow dynamics is lacking, increased ventilation might inadvertently exacerbate pathogen dispersal. This assertion is supported by Birnir et al. \cite{Birnir}, who conducted an analysis of fluid dynamics in a restaurant setting with recommended air-conditioning rates. Their findings indicate that the risk of transmission when dining at separate tables within fifteen minutes is comparable to that of sitting adjacent to an infected individual.

Bhatia and De Santis [4] used CFD to investigate the dispersion of respiratory droplets in confined indoor spaces, Their findings revealed that approximately 75% of these droplets disperse within the cabin, reaching a maximum distance of two meters behind the infected person within a duration of 20 seconds. This is one of many investigations that have shown how CFD can provide actionable data to reduce pathogenic transmission. Strategies have been proposed to leverage CFD data for reducing pathogen transmission and energy consumption, such as the zonal demand-controlled ventilation strategy developed by Ren et al. [5] which uses cameras to detect occupancy of each room to inform an appropriate ventilation rate. Their findings show that this technique could reduce the infection rate to 2.8% and reduce energy consumption by 34%. According to the World Green Building Council [6] 28% of global carbon emissions come from heating, cooling and powering buildings yet very little is known how real-time control mechanisms can improve operational efficiency. Wu et al. [7] proposed a house with over 300 sensors to act as inputs to a CFD system which would then inform how the actuators should operate to increase efficiency. These sensors would then be combined with local weather forecasts to create a 'digital-twin' of the house and thus actuators could optimise the thermal efficiency of the house. This study was limited by the fact that the actuators cannot react to real-time changes to boundary conditions because long simulation times.

Recognising the limitations of conventional techniques, there is growing interest in SPMs for indoor environmental modeling. SPMs leverage Artificial Intelligence (AI) to overcome the computational barriers associated with traditional CFD methods. While numerical simulations rely on iterative mesh-grid updates, smart proxy models use data-driven approaches to predict complex indoor airflow dynamics. This report presents a successful implementation of SPMs for predicting indoor velocity vector fields, offering potential to produce accurate results in seconds.

The structure of this report is as follows: Section I-A reviews previous work in enhanced CFD, providing insights into the advantages and limitations of existing methodologies. Section II introduces the theoretical framework underpinning the AI-based SPM approach. Section III outlines the methodology, detailing the steps involved in generating training data, model development, and validation. Finally, Section IV presents the results and key findings of the study, highlighting the implications for indoor environmental modeling and related applications.

A. Literature review

In the field of **Numerical Simulation**, the Navier-Stokes equations are commonly discretized in space and integrated over time to accurately capture flow structures. While Direct Numerical Simulation (DNS) offers high accuracy, it demands substantial computational resources. Jeon et al. [8] introduced a novel approach that applies deep neural networks to the finite volume discretisation scheme, achieving alignment with high-resolution data at reduced computational costs. However, this method is primarily used for unsteady simulations and is susceptible to significant error accumulation over time. To address the time dependent performance of the finite volume method, Stevens and Colonius [9] leveraged Long Short Term Memory Networks to enhance simulation accuracy. Another strategy aimed at mitigating computational expenses in high-fidelity simulations involves using machine learning techniques to correct coarse-grid simulation results to match finer grid outcomes. Kochkov et al. [10] demonstrated 40 times less computation time in fluid simulations using grid correction methods. However, despite these advancements, this approach still entails considerable computational expenses and may not offer real-time CFD results.

The **Reynolds-Averaged Navier-Stokes (RANS)** equations stem from the Navier-Stokes equations through a time-averaging process aimed at capturing mean flow behavior while segregating it from turbulent fluctuations. However, this averaging procedure entails the loss of information pertaining to small-scale turbulent structures. These unresolved turbulent fluctuations introduce additional terms in the equations, notably the Reynolds stress tensor, which signifies the correlations among turbulent velocity fluctuations [11]. Machine Learning techniques are frequently employed in this domain to enhance the accuracy of predicting the Reynolds stress tensor. Ling et al. [12] proposed a novel Artificial Neural Network (ANN) architecture for predicting the Reynolds stress tensor. Their approach incorporated a multiplicative layer featuring an invariant tensor basis to encode Galilean invariance to enhance the Networks understanding of the underlying physical equations. This method yielded more precise results compared to traditional RANS models, encompassing both linear and non-linear eddy viscosity models. However, a significant limitation lies in its limited adaptability to diverse flow scenarios. Addressing this challenge may involve enriching the training dataset to encompass a wider range of flow scenarios, potentially enabling the model to capture more comprehensive patterns and generalize across various conditions. Iang et al. [13] introduced a Physics-Informed Residual Network tailored to forecast turbulence across broader and more adaptable flow scenarios. This network aims to understand both the structure and specific parameters governing turbulent flows. By integrating non-unique mappings, it accommodates complex data relationships, while incorporating physical constraints to uphold physically feasible solutions. Additional constraints are integrated to mitigate the impact of noise, enhancing the model's stability against system disruptions. However, a limitation of this model lies in its requirement for large amounts of high-fidelity CFD data to achieve meaningful flow generalisability. **Large Eddy Simulations (LES)** involve filtering the Navier-Stokes equations to simulate the largest scales while modeling the smallest scales with a Sub Grid Scale (SGS) model due to their demanding computational requirements. However, SGS models often lack generalisability in complex flow scenarios or at high Reynolds numbers [14]. Beck et al. [15] employed an ANN based on local convolutional filters to predict the mapping between flow in a coarse simulation and closure terms, using filtered DNS of decaying homogeneous isotropic turbulence. These models necessitate large datasets to exhibit satisfactory generalisability. On the other hand, Novati et al. [16] utilised Multi Agent Reinforcement Learning (MARL) to estimate unresolved subgrid-scale physics. While this unsupervised method offers improved generalization across flow conditions, the control policies trained by MARL may be highly complex, with numerous parameters encoding correlations between inputs and the eddy-viscosity coefficient. Consequently, these policies may not be easily transferable to other flow solvers utilizing different numerical discretisation methods or underlying physics meaning that this solution is very limited in its application.

Reduced Order Models (ROMs) utilise dominant coherent structures within complex flows, extracted via techniques such as Proper Orthogonal Decomposition (POD). While enhancing computational efficiency in CFD simulations, ROMs sacrifice the ability to generalise to new situations. Linear methods like POD and Principal Component Analysis struggle with the nonlinear nature of fluid flows. Deep learning frameworks like autoencoders distill essential data features into a reduced latent space, capturing nonlinear patterns more effectively. Lee and Carlberg demonstrated that deep convolutional autoencoders can outperform linear ROMs [17]. Deep learning frameworks can capture the nonlinear behaviors present in fluids but come with a larger computational expense and can overfit to training data. **Smart Proxy Models (SPMs)** harness AI to develop efficient surrogate models for complex numerical simulations, particularly excelling in scenarios involving the intricate simulation of spatio-temporal flow structures within fluids. These models bypass computationally intensive tasks by learning patterns in simulation data, providing quick yet accurate approximations. Aboaba et al. [18] employed SPMs to replicate thermal-flow patterns of pressure, temperature, and species concentrations (nitrogen, oxygen, and carbon dioxide) to less than 10% error. Their approach also had faster execution times compared to numerical CFD simulation methods. These results were achieved with a limited amount of training data, and the computations took 5-6 minutes on an inexpensive laptop, contrasting with the 24 hours needed to collect the training data on a high-performance computer. Researchers addressed the need for extensive data requirements by utilising **feature engineering**, defining each node point as a function of distances to each boundary condition. Additionally, they employed fuzzy clustering to assess the importance of each feature to each cluster. Shahkarami and Mohaghegh [19] utilised SPMs to expedite subsurface modeling for enhanced oil recovery, achieving highly accurate results with 98.9% less computational time. However, this investigation required careful feature selection and preprocessing steps. Wui et al.

[20] introduced SPMs to streamline the modeling of fractured reservoirs, notorious for their complexity. By analysing a database derived from numerical simulations, SPMs identify patterns, reducing computational demands. These models accurately predicted production rates, including fuel and water oil, in a fraction of the time compared to traditional methods, demonstrating their efficiency and effectiveness. Haghshenas et al. [21] used SPMs to predict optimal water injection sites, aiming to boost oil recovery rates. They assembled a training dataset using Latin Hypercube Sampling to evenly distribute oil injection sites across the search space. Additionally, they performed linear and non-linear input analyses for dimensionality reduction, eliminating redundant variables. Introducing hyper-features as linear functions of multiple features addressed non-zero values and reduced skewness. Normalisation of each feature enhanced the training efficiency of the ANN. Despite these efforts, they observed that inherent errors within the ANN accumulated over time, resulting in a decrease in model performance even with a large dataset size. **Ensemble learning** is a machine learning technique that involves combining multiple individual models to produce a stronger and more accurate predictive model. While it can be challenging to make a single ANN generalisable to different flow scenarios, an ensemble network can provide higher accuracy with a reduced computational load. For instance, Bussov and Nättilä [22] used computer vision to predict turbulent fluid flow, integrating an ensemble of models to extract various shapes in the images and combine them to construct large-scale structures in the data. **Transfer learning** applies knowledge from solving one problem to address related ones. Yu et al. applied transfer learning to effusion cooling, achieving 40-60% more accuracy than using a single ANN model [23].

In summary, modeling airflow in indoor environments presents significant challenges as low air viscosity leads to the formation of turbulent structures at low velocity. Achieving accurate predictions using traditional CFD techniques necessitates a fine mesh to capture small-scale features, this process is computationally expensive and slow. AI enhanced CFD techniques have been researched extensively in literature and focus on accelerating certain processes in computational modelling such as predicting the Reynolds stress tensor or improving the mapping of LES models. These strategies still require expensive meshing processes and often fall short in delivering a critical combination of efficiency and generalisability. Despite the development of instant inference time fluid models tailored for the petroleum industry, there is a notable absence of research exploring their application to indoor airflow simulations. This gap is particularly significant given that research indicates CFD informed ventilation strategies can substantially reduce building energy consumption and minimise the risk of pathogenic transmission. The primary challenge lies in achieving prediction accuracy across multiple orders of magnitude of velocity, crucial for applications such as pathogenic transmission and energy dispersion. Addressing these limitations will require careful data preprocessing and ANN architecture planning to provide the best results.

II. Theory

A. Artificial Neural Networks and Deep Learning

AI and machine learning have witnessed widespread adoption in recent years, because of the growing accessibility of Graphical Processing Units (GPUs) and data storage. At the core of machine learning is supervised learning, a prevalent algorithm that enables machines to learn from labeled input data. This approach involves learning a function that effectively maps a set of inputs to desired outputs. Typically, the input layer comprises features that encapsulate the variability within a system. Meanwhile, hidden layers consist of weighted connections with non-linear activation functions, facilitating the capture of intricate relationships between inputs and outputs. Deep Learning, a subset of machine learning, extends this paradigm by incorporating multiple hidden layers, enabling hierarchical feature extraction and the acquisition of complex patterns.

B. Evaluation Metrics

In the training process of ANNs, an initial set of weights is randomly assigned, and the error between the network's predictions and the ground truth is calculated. Subsequently, this error is propagated backwards through the network via a process known as backpropagation. Through backpropagation, adjustments are made to the weights and biases to iteratively reduce the error over time. The performance of the ANN is evaluated to minimise the average error across all outputs, aiming to create a well-rounded predictive tool.

Equation 1 defines the Mean Absolute Error (MAE), which quantifies the average magnitude of errors between predicted values ($\hat{y}i$) and actual values (yi) [24]. However, it cannot simultaneously optimise for multiple outputs on different scales.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

Equation 2 represents the Mean Absolute Percentage Error (MAPE), which assesses the average absolute percentage difference between predicted and actual values [25]. Being a relative error metric, it treats all scales equally. However, it tends to yield disproportionately large errors when dealing with very small actual values ($y_i << 1$). This means that in physical systems with noise, small absolute variations saturate the output space.

$$MAPE = \frac{100}{n}\sum_{i=1}^{n}|\frac{y_i - \hat{y}_i}{y_i}| \tag{2}$$

Equation 3 defines the Mean Absolute Arctangent Error (MAAE), which effectively evaluates the arc-tangent of the MAPE [26]. Although less intuitive than MAPE percentages, MAAE is advantageous in that it is less sensitive when predicting values close to zero. This reduced sensitivity arises because the error is confined within the range of 0 to $\frac{\pi}{2}$. The addition of ϵ on the denominator also serves to prevent computational errors when using this loss function.

$$MAAE = \frac{1}{n}\sum_{i=1}^{n}arctan(|\frac{y_i - \hat{y}_i}{y_i + \epsilon}|) \tag{3}$$

C. Smart Proxy Models and coordinate transformation

SPMs represent a category of surrogate models that harness AI techniques to accurately and efficiently approximate complex computational simulations. These models utilise feature engineering to transform physical features into inputs that algorithms can learn from. For instance, equations 4, 5, and 6 comprise the complete set of equations required to convert Euclidean coordinate systems to Spherical coordinate systems. In these equations, r denotes the radial distance or velocity, θ represents the polar angle measured from the positive z-axis (vertical), and ϕ denotes the azimuthal angle measured in the x,y plane (horizontal). It is important to note that the calculation of ϕ requires the use of the atan2 function (or a similar four-quadrant arctangent operation) as the angle spans 2π radians of rotation.

$$r = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2} \tag{4}$$

$$\theta = \text{arcos}\left(\frac{\Delta z}{\Delta r}\right) \tag{5}$$

$$\phi = \text{atan}\left(\frac{\Delta y}{\Delta x}\right) \tag{6}$$

D. Fluid Dynamics

In fluid dynamics, the behavior of fluid flow can vary widely depending on factors such as velocity, viscosity, and geometry. Laminar flow is defined by smooth, orderly movement of fluid particles along parallel paths, dominated by viscosity, ensuring stability. In contrast, turbulent flow is chaotic and typically occurs at higher speeds, small localised swirling eddies promote intense mixing and velocity fluctuations. During the transition from laminar to turbulent flow, small localised eddies begin disrupt ordered laminar flows. Stagnant flow occurs when fluid particles stop at obstacles or due to the non-slip boundary condition imposed by walls.

Different flow regimes are governed by distinct equations and exhibit unique phenomena. CFD flow solvers often utilise Reynolds Averaged Navier Stokes (RANS) equations to represent a time-averaged form of the Navier-Stokes equations. In RANS equations, flow variables are decomposed into mean and fluctuating components. Typically, the mean flow is further separated into laminar and turbulent components. Turbulent flow is then handled as a statistical phenomenon.

Equation 7 provides the formula for the Reynolds number of fluid flow where ρ is the density of the fluid ($\frac{kg}{s}$) is the fluid velocity ($\frac{m}{s}$), L is the characteristic dimension (m), and μ is the kinematic viscosity ($\frac{Ns}{m^2}$). Typically a Reynolds number of less than 2,000 will indicate laminar flow and greater than 4,000 indication turbulent flow.

$$Re = \frac{\rho u L}{\mu} \tag{7}$$

Equation 8 provides a formula for finding the volumetric flow rate of air ($\frac{l}{s}$), where $\dot{m}$ is the mass-flow rate ($\frac{kg}{s}$), ρ is the density of air ($\frac{kg}{m^3}$)

$$\dot{Q}(L) = 1,000\frac{\dot{m}}{\rho} \tag{8}$$

III. Methodology and development

A. CFD Model Development

A simple HVAC model was created to simulate indoor air conditioning systems, using 400mm by 400mm ducts to maintain a comfortable airspeed of 0.2m/s [27]. Experimentally, this velocity corresponds to a mass flow rate of 0.015 kg/s using equation 8. With a recommended air mass flow rate of 2.5l/s per occupant [28], the system can accommodate four people in a 5m x 5m x 3m room, typical for office spaces.

The chosen model was created using Ansys Discovery which is a Computer Aided Design software. The 'named-selection' tool extracted the boundary-conditions of the model such as walls, inlets and outlets. A smooth-transition boundary layer has been introduced along all four walls to enforce a no-slip boundary condition, an approximation for the transition from stagnation flow on the walls to the free-stream flow in the body of the room. The volume enclosed by the surface was given the thermodynamic properties of air at 25°C, the four walls were given the surface roughness properties of concrete to replicate a typical indoor environment. A mesh independence study was then used to determine how fine a mesh is required to capture the fluid flow, a maximum mesh-size of 0.025m was chosen such that the percentage difference between x,y,z velocities between trials varied by less than 5%.

For this investigation, Ansys Fluent utilises the steady-state solution scheme, ideal for scenarios where HVAC rates remain constant for extended periods, resulting in stable velocity-flow fields within rooms. Although predominantly laminar, turbulent regions around HVAC outlets significantly affect fluid velocity profiles. Hence, the $k - \omega$ turbulence model is chosen to represent turbulent flow accurately. Fluent solves an augmented version of the Navier-Stokes equations at each node in the volume mesh, incorporating turbulent kinetic energy and frequency terms. The solver iterates by initially guessing the flow field and measuring differences in computed values (e.g.pressure and temperature) compared to the previous iteration, storing them as residuals. Convergence of these residuals throughout the simulation ensures the solver approaches the theoretical solution. After completion, equations 7 and 8 were used to verify that the Reynolds number and velocity of the air at the HVAC inlet align with theory.

Following the simulation's completion, data consisting of x, y, z positions and velocities at each node in the mesh were exported into a CSV file totaling 3.4GB. This file was then transferred to Google Drive for preprocessing. For the investigation, Google Colab was used as the development environment as it has free access to GPUs that accelerate the development of ANNs and it links seamlessly with google drive to aid in data transfer.

B. Initial development

In an unoccupied room, airflow tends to be uniform and stagnant, with zero velocity throughout. However, the introduction of boundary conditions, such as HVAC systems, disrupts this uniformity, leading to airflow patterns that eventually settle into a steady-state equilibrium. As a result, the velocity at any point within the room becomes a function of its position.

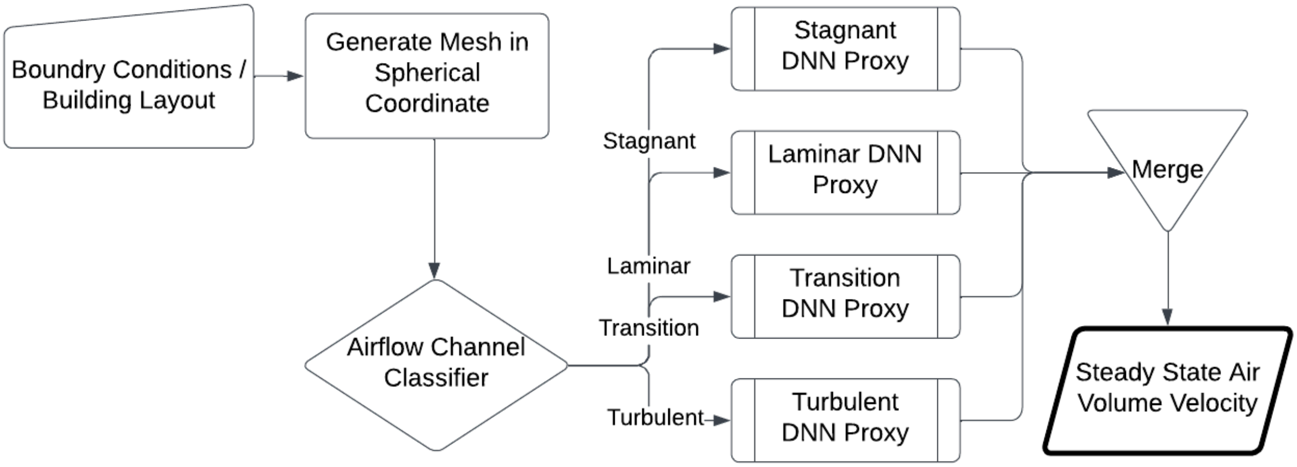
Boundry Conditions / Building Layout
Generate Mesh in Spherical Coordinate
Airflow Channel Classifier
Stagnant
Laminar
Transition
Turbulent
Stagnant DNN Proxy
Laminar DNN Proxy
Transition DNN Proxy
Turbulent DNN Proxy
Merge
Steady State Air Volume Velocity

Fig. 1: Design Overview of Smart Proxy Model

To create an SPM model which can generalise well, it must be provided with input features that are applicable across any room size and inlet/outlet configuration. Table I presents the initial set of input features utilised in the analysis. The positions relative to HVAC inlets and outlets were represented using both coordinate differentials (Δx, Δy, Δz) and absolute distances (r). This dual representation is necessary because HVAC air jets typically exhibit relatively consistent velocities directly beneath them, which then diminish rapidly as one moves away from the source across the room. The SPM outputs comprise the separated x, y, and z components of velocity (vx, vy, vz) as they fully describe the velocity flow field within the room.

Walls	Ceiling	Floor	Right	Left	Front	Back
Inlet	Δx	Δy	Δz	$r = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$		
Outlet	Δx	Δy	Δz	$r = \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$		
Output	vx	vy	vz			

TABLE I: Table showing input and output features of the Euclidean SPM

The challenge with velocity vectors in Euclidean space arises from the independent variation of velocity components along different axes, often resulting in significantly different scales. This discrepancy led to instabilities during the learning process of the initial ANN, hindering its ability to discern any meaningful patterns. To tackle this challenge, an alternative approach involves representing velocity in a spherical coordinate system using equations 4, 5, and 6, and replacing Δx, Δy, Δy with vx, vy and vz. In this system, the velocity components are combined into a single radial magnitude, while θ and ϕ represent the inclinations of the vector to the vertical and horizontal axes, respectively. This conversion not only alleviates scaling issues but also enhances the memory efficiency of the SPM by reducing the number of features required, as illustrated in Table II. Additionally, Transforming the coordinate distances to the HVAC inlet and outlet into spherical vectors offers significant advantages for the learning process of ANNs. By consolidating the coordinate distances (Δx, Δy, Δz) into a single distance magnitude, denoted as r, the distance becomes strictly positive, providing a more intuitive representation for the ANN. Moreover, incorporating the reciprocal of the distance reflects the physical phenomenon where velocity tends to increase as the distance to the vent decreases. However, to avoid numerical instability, distances less than 1mm were standardised to exactly 1mm, effectively bounding the feature within a range of 0 to 1,000 and stabilising the ANN during training. Fluid velocity near boundaries tend to be zero, however the direction of velocity also changes around walls due to the recirculation of air being deflected. This relationship between fluid velocity and wall-distance is therefore complex as thus the inputs were left as raw distances.

Walls	Ceiling	Floor	Right	Left	Front	Back
Inlet	$1/r$	$\theta(0,\, \pi/2)$	$\phi(0, 2\pi)$			
Outlet	$1/r$	$\theta(0,\, \pi/2)$	$\phi(0, 2\pi)$			
Output	r	$\theta(0,\, \pi)$	$\phi(0, 2\pi)$			

TABLE II: Table showing the inputs and outputs of the spherical SPM

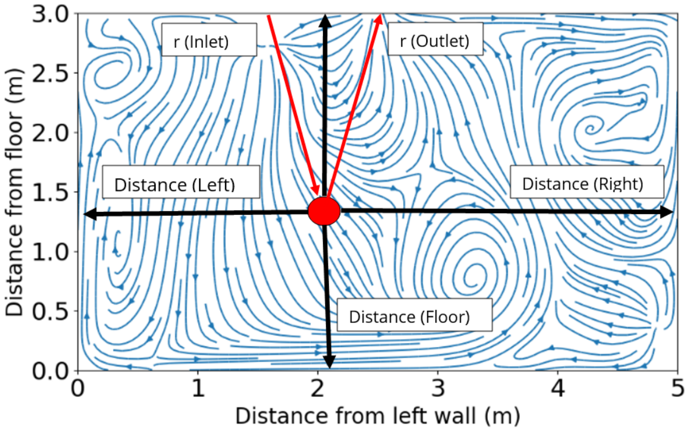

Fig. 2: Graphic showing the inputs of the spherical SPM with fluid flow

In the context of spherical vectors to the inlet and outlet, the polar angle (θ) denotes the angle between the positive z-axis (vertical), while the azimuthal angle (ϕ) signifies the angle on the x-y plane. To align with the natural airflow throughout the room, angles were computed from the inlet to the reference point and from the reference point to the outlet, as depicted in Figure 2 illustrating the inputs to the ANN and airflow streamlines. These angles were represented in radians, as a degree variations have a much larger magnitude and the ANN would assign greater significance to the angular input than the velocity magnitude input. Furthermore, standardising the angles to begin from 0 radians ensures that there are no negative values, which could potentially cause the ANN to assign hierarchical significance to negative values. This adjustment helps maintain consistency in the representation of angles and avoids any unintended biases in the learning process of the ANN.

C. Loss function testing and evaluation

Initially, the Mean Absolute Error (MAE), as depicted by equation 1, served as the error metric. However, due to the significantly larger magnitude of angles compared to velocities, the model tended to prioritize minimising angle errors. Consequently, this led to velocity predictions becoming random and inaccurate. To address this issue, the Mean Absolute Percentage Error (MAPE), illustrated by equation 2, was introduced. However, when predicting velocities as low as 1×10^{-7}, velocity errors which are practically negligible may result in percentage errors of several thousand percent and dominate the learning process. As a solution, the Mean Absolute Arctangent Error (MAAE), represented by equation 3, was adopted. This metric calculates the arctangent of the percentage error, confining the error between 0 and 2π radians. This ensures that large errors cannot dominate the learning process and as such the error becomes representative of model performance.

D. Deep Learning

Initially, a simple ANN was constructed, comprising a 12-node input layer followed by two hidden layers with 32 and 64 nodes, respectively. Each hidden layer used the Rectified Linear Unit (ReLU) activation function to detect data non-linearities. The output layer consisted of three nodes to predict velocity magnitude, polar angle and azimuthal angle. The results were quite poor and the ANN showed less deflection in the training curve than expected, this suggested that the model was under-fitting. To rectify this, a deeper ANN was created to see if it could get a better fit to the dataset. The ANN had seven hidden layers with node counts increasing in a pyramidal arrangement: 32, 64, 128, 256, 128, 64, and 32 nodes in each respective layer. This architecture enables the extraction of increasingly abstract features at intermediate layers, which are then distilled down to the three output nodes, proving effective for regression tasks. The loss curves for each design iteration are depicted in Figure 3. It's evident that transitioning from a simple 3-hidden layer arrangement to 7-hidden layers didn't immediately enhance performance at the beginning of training. However, as training progressed, the deep-learning curve managed to achieve a lower loss than the simple ANN around the 20-epoch mark, indicating the impact of complex pattern recognition. Hyperparameter optimization notably reduced model loss from the outset of training, consequently accelerating the learning process. This performance improvement can primarily be attributed to adjusting the learning rate from 0.001 to 0.0017. The higher learning rate prevents the algorithm from getting stuck in local minima, leading to more efficient convergence towards better solutions.

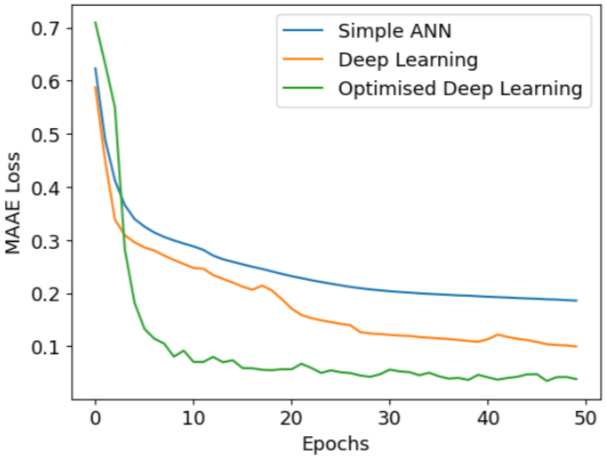

Fig. 3: Loss curves for each model design iteration

E. Flow Regime Segmentation ANN

Analysis of error maps revealed that velocity vector orientations (θ and ϕ) and magnitudes were highly sensitive to changes in position which was almost impossible for the ANN to predict and hence resulted in large errors. To prevent large errors from dominating the learning process, stagnant air data was segregated from higher velocity data. Additionally, splitting the dataset into four segments representing their respective flow regimes led to a significant decrease in the average error for all segments. Different flow regimes are governed by different physical properties, hence an ANN specialised in predicting each regime independently will yield the best results. Flow regime change can be caused by many factors including Reynolds number, surface roughness of each wall, and geometry changes. However, for simplicity, classification will be based on the Reynolds number using Equation 7. To perform flow regime segmentation, a simple ANN was created and trained on the inputs described by Table II. The ANN was optimised using grid-search, resulting in a model with four layers, each with 64, 128, 256, and 512 nodes respectively. The output layer consisted of four nodes representing each flow regime class. Each output used a softmax activation function to ensure the model's predictions sum to 1, enabling a probabilistic interpretation of the output. For assessing the model's performance, categorical cross-entropy loss was used as the loss function, it evaluates the dissimilarity between true and predicted probabilities, providing a robust measure of the model's accuracy in classifying the data. The learning rate had a significant impact on model performance, with the best value determined to be 0.001. The results of segmentation are displayed in Table III. The simple ANN was able to assign flow regimes to each point in the room with a weighted average accuracy of 98.2%.

	Turbulent	Transition	Laminar	Stagnant
Percentage of cells (%)	2.53	7.23	88.21	2.03
Min-Velocity (m/s)	0.021	0.01	0.0002	0
Min Reynolds Number	4000	2000	100	0
Prediction Accuracy (%)	99.13	96.41	98.65	86.24

TABLE III: Table showing the flow regime segmentation

Figure 4 shows a cut-plot of the room at the location of the air-mass flow inlet. This diagram shows the physical location that data for each dataset is extracted from. The turbulent dataset is mainly composed of flow below the mass-flow inlet and outlet vents, the transition region typically outlines any turbulent regions or sharp changes in geometry. Stagnant flow predominantly occurs due to the boundary layer of each wall.

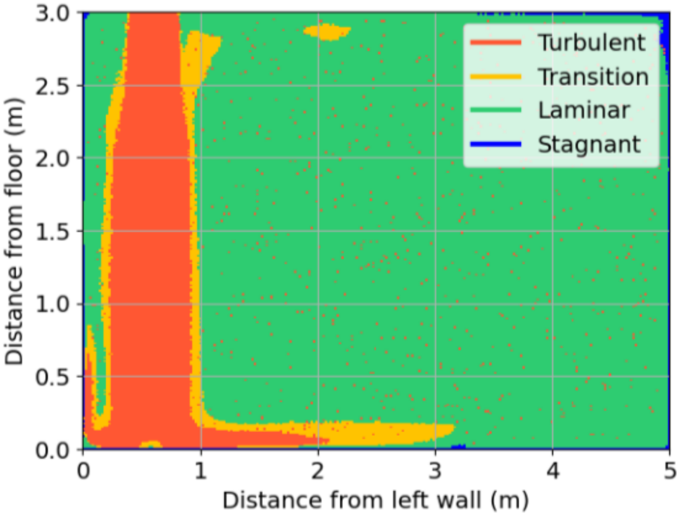

Fig. 4: Graphic showing the fluid flow regimes in a cut section of a room with a HVAC system

F. Design Overview and Testing

Figure 1 shows the architecture of the proposed SPM model, for each room the location of inlets, outlets and walls is recorded in a arbitrary coordinate system. A Boundary-Condition Vector Generation algorithm then constructs proxies for the SPM at a specified number of sample points within the room. A simple ANN then receives this information and classifies each sample-point into a separate flow regime. The velocity vectors in each flow regime are then predicted by separate Deep Learning regression models and the total velocity flow field is predicted. Conventional CFD relies on a dense mesh across the entire room to accurately predict flow fields. Both mesh creation and simulation are both time consuming processes. In contrast, SPMs are not constrained by the number of points they predict, meaning that the SPM inference time could be accelerated to milliseconds if an equally as accurate but less dense flow field is required. During training, data was generated from two rooms on Ansys Fluent and a training, validation and test split of the dataset was created 70%, 15% and 15%, the model was trained and then tested on the unseen test set to validate the models ability to predict unseen data. The MAAE for each split was recorded and converted in to MAPE which allows a more intuitive analysis of results.

IV. Final Results

A. Evaluation metrics overview

The results depicted in Figure IV illustrate that the SPM predicts turbulent and transitional flow to a test MAPE of less than 2%. The accuracy was very high as turbulent flow is generally clustered in very close proximity to the HVAC inlets and outlets and thus are highly influenced by the input features. These results are very promising as flow from these regimes can have a velocity magnitude of up to 1000 times the magnitude of a velocity in laminar flow and thus will affect the entire velocity flow field much more. On the other hand, the laminar flow model has a relatively higher test error of 6.8%, this discrepancy could be attributed to the dispersed nature of laminar flow throughout the room, leading to more complex flow patterns that are inherently harder to predict accurately. The stagnant flow model performs very poorly with a MAPE of 68%, as discussed earlier, at low velocity small, even random variations in conditions can lead to large relative changes in angle and velocity which are impossible to account for. The test MAPE is consistently as low as the training MAPE which demonstrates that the model performs well with unseen data. The Inference time refers to the time taken for a laptop using an 9th gen Intel Core I7 processor with 16GB of RAM to predict the velocity vectors at every grid point in a simulation split across each regime. The total time was around 0.8s (including regime classification) which takes 99.996% less time than using Durham Universities Hamilton 8 Supercomputer which takes around 5 hours to run using three compute nodes at 256GB of RAM.

Seen Room	**Turbulent**	**Transition**	**Laminar**	**Stagnant**
Training MAAE	0.0250	0.0218	0.0551	0.5881
Training MAPE (%)	2.50	2.18	5.52	66.30
Test MAAE	0.0190	0.0184	0.0458	0.5820
Test MAPE (%)	1.90	1.84	4.58	65.80
Inference Time per room(s)	0.0352	0.0496	0.6414	0.0193

TABLE IV: Table summarising the evaluation metrics for Smart Proxy Model

B. *Discussion of performance*

During the optimisation stage of the SPM, the SPM was tested with the same inputs but only tasked with predicting the velocity magnitude which led to very poor results. This indicates that the neural network weights necessary for forecasting azimuthal and polar angles are crucial in accurately predicting velocity magnitude, this highlights the importance of a complete output basis withing SPM models. Indoor airflow is heavily governed by eddies and swirls generated by the deflection of air from walls and edges and as such the orientation of velocity tells the SPM whether the airflow will be stopped by an obstacle or accelerated. The model was initially trained and tested on one simulation and then the training set was expanded to include two simulations. As the dataset was expanded, the model training and test accuracy increased significantly suggesting it scales well to a more expansive dataset. Deep learning models are essentially non-linear interpolation models and as such they require data points which are similar to which they are being tested, the model will become better at generalisation as the training set expands. Figures 5 and 6 show the predicted velocity streamlines produced by Ansys Fluent and SPM respectively. The plot is a cut-section of the room which intersects with the HVAC inlet. In these plots each line represents the streamline of fluid flow and longer lines tend to represent larger absolute velocities with continuous flow. This plot was generated by converting the spherical velocity back into the original euclidean velocity. Some noticeable discrepancies between plots are in the bottom right-hand corner, where the air streamlines interact with the right-wall and the floor, presenting a challenge for the SPM due to the sharp change in geometry. This difficulty could stem from the SPM's poor performance in modeling the stagnation regime, resulting in a decrease in accuracy when predicting flow behavior around boundary layer regions. One potential solution to address this issue is to refine the training data for the stagnation regime within the SPM. By smoothing out the angle changes within the stagnation regime training data, only the predominant flow directions would prevail which would reduce anomalous errors and stabilise training. Moreover, further discrepancy appears evident between the free stream jet at the HVAC inlet and the surrounding flow, indicating a discontinuity between predictions made by turbulent and transition models. Given that the turbulent model yields the most accurate predictions, the SPM could gradually integrate its predictions from one model to influence the predictions at the boundary of the next model, allowing the SPM to iteratively build the flow field. Such a strategy has the potential to mitigate errors across transition, laminar and stagnant flow regimes.

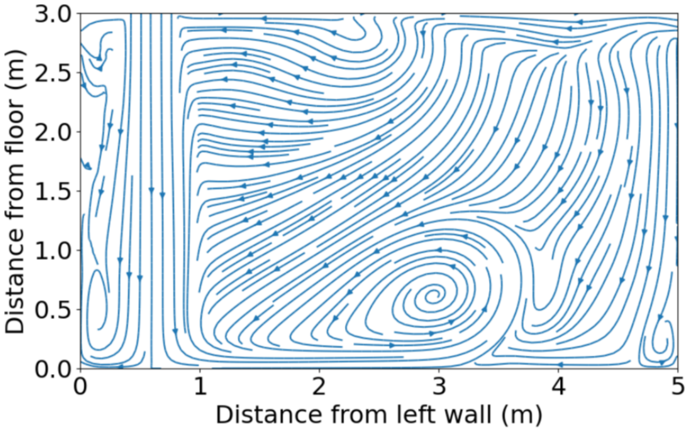

Fig. 5: Standing view streamplot of the HVAC inlet generated by Smart Proxy Model

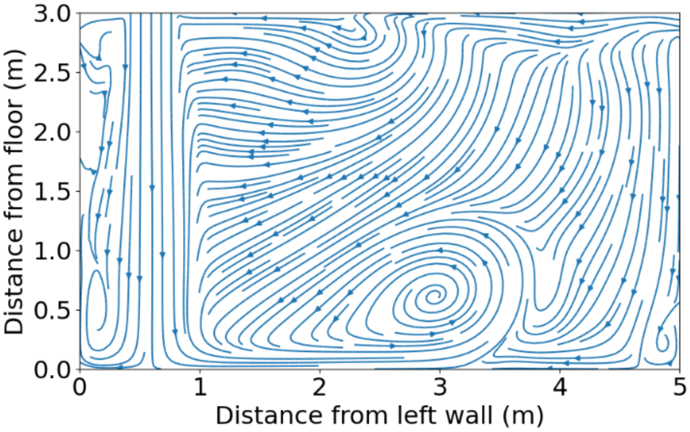

Fig. 6: Standing view streamplot of the HVAC inlet generated by Ansys Fluent

Figures 7 and 9 show heat-maps from the SPM of the predicted velocity magnitude, azimuthal angle and polar angle for the two simulations covered by the training set. The yellow jet in the velocity magnitude plots represents the flow coming from the HVAC inlet. The yellow square in the polar angle plot also represents the inlet for which the velocity vectors pointing vertically downwards from the ceiling to the floor. The spiral structure at the bottom of the polar angle plots shows the swirling vortex created by the suction of the HVAC outlet. In the azimuthal angle plots, the presence of red indicates that the velocity vector aligns positively along the back wall. Conversely, green-blue hues signify that the velocity vector aligns oppositely along the back wall. The distinction between setting up simulation 1 and 2 was a minor adjustment, involving a slight shift of the HVAC inlet away from the outlet. However, the azimuthal angle plots reveal the emergence of distinct flow fields for each boundary condition. This illustrates the significant influence of boundary condition positioning on airflow patterns within a room. In dynamic environments like public offices or hospitals, where conditions are in constant flux, traditional numerical CFD methods struggle due to their computational complexity. Nevertheless, the dynamic nature of flow fields also implies that SPMs would require training on a diverse range of simulations to generalise well to entirely new room configurations. Figures 8 and 10 are heat maps showing the MAAE of the SPM's output features. The main source of error when predicting the velocity magnitude seems to arise from the corners of the room which suggests that the SPM struggles with predicting stagnant and boundary layer velocities due to small absolute differences in velocity having large relative impacts. The SPM seems to predict the polar angle quite well but develops larger errors near the HVAC inlets and outlets when the air-velocities switch direction from upwards to downwards. By contrast, the error is much more prevalent in the azimuthal angle predictions. Figure 10 shows that when the velocity switches from aligning parallel with the back wall (red) to the opposite direction (green) the error increases substantially. Evidently the SPM still struggles with some of the sharper transitions in direction.

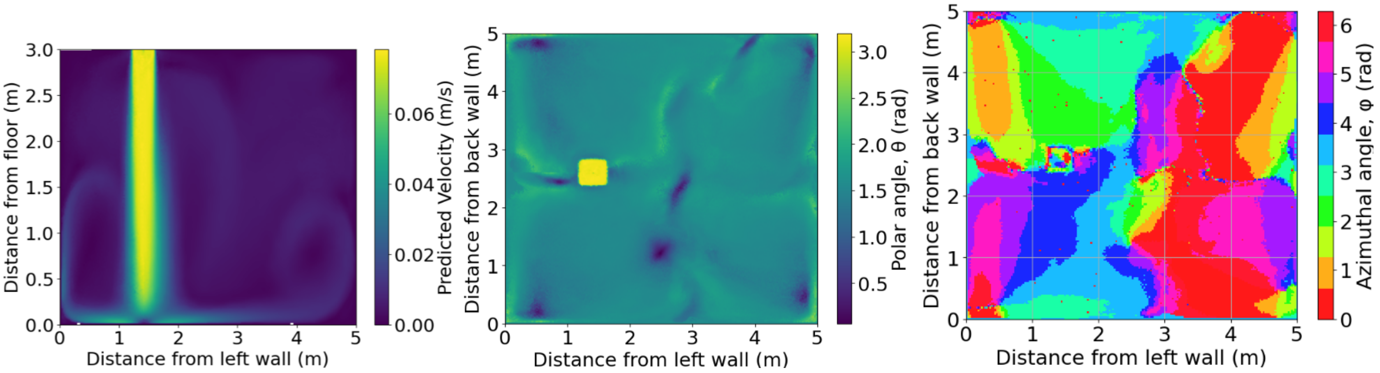

Fig. 7: Left: Magnitude of velocity (standing view), Middle: Polar angle (θ) (birds-eye view), Right: Azimuthal (ϕ)
Plots showing predicted heat-map cut-sections of each output feature. (Simulation 1)

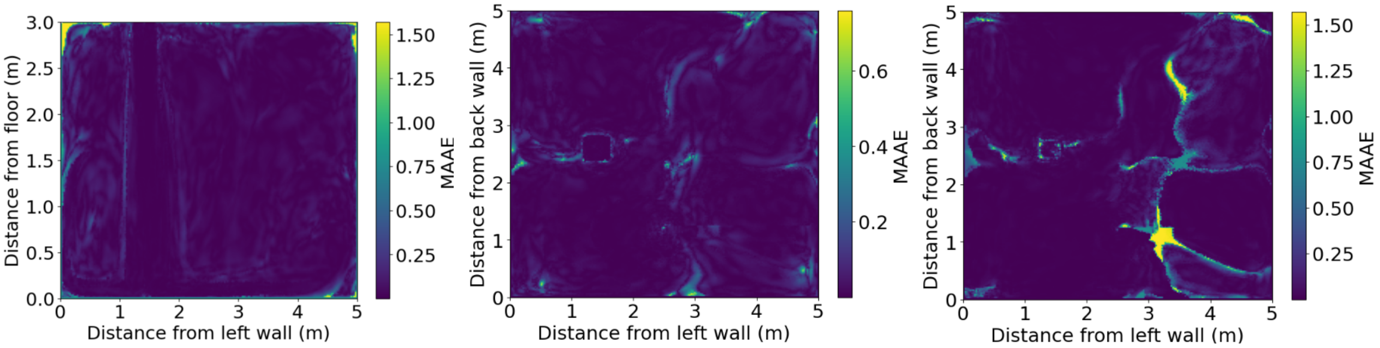

Fig. 8: Left: MAAE of velocity magnitude(standing view), Middle: MAAE of polar angle (θ) (birds-eye view), Right: MAAE of azimuthal angle (ϕ) (birds-eye view)
Plots showing heat-map cut sections of the MAAE of each output feature. (Simulation 1).

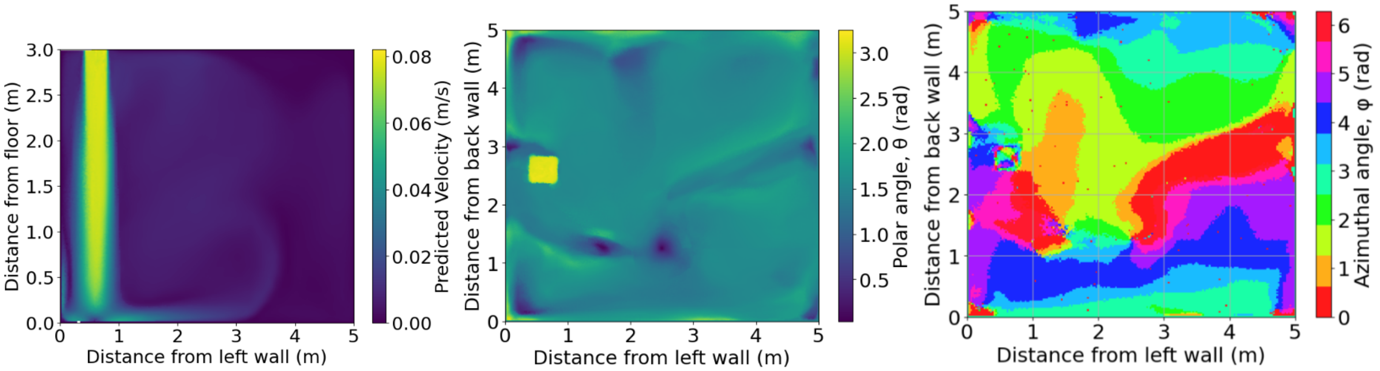

Fig. 9: Left: Magnitude of velocity (standing view), Middle: Polar angle (θ) (birds-eye view), Right: Azimuthal angle (ϕ)
Plots showing predicted heat-map cut-sections of each output feature. (Simulation 2).

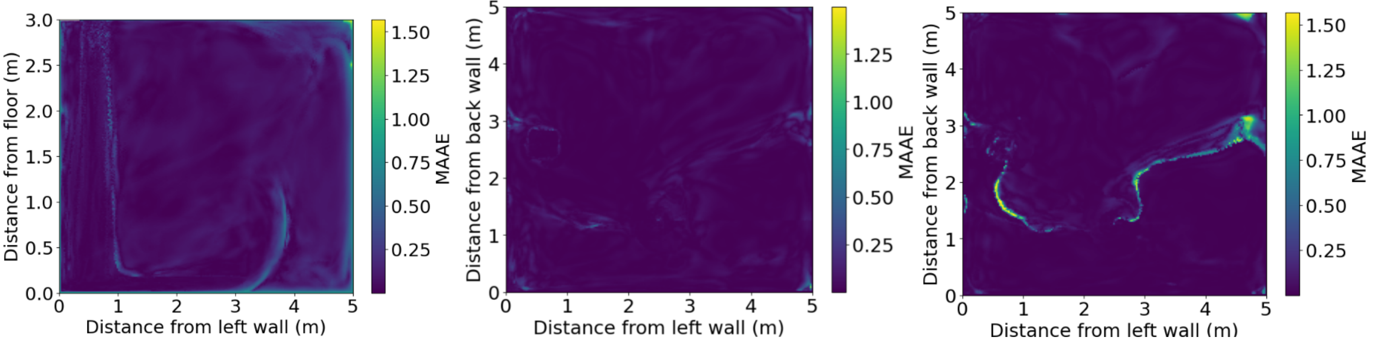

Fig. 10: Left: MAAE of velocity magnitude(standing view), Middle: MAAE of polar angle (θ) (birds-eye view), Right: MAAE of azimuthal angle (ϕ) (birds-eye view)
Plots showing heat-map cut sections of the MAAE of each output feature (Simulation 2)

C. Further Development

This report provides an introduction to the prediction of a velocity flow field using SPMs however, there are many more areas to investigate within this field. In order for SPMs to be able to predict to unseen rooms they will need to be trained on a diverse range of CFD data. Scheme files can be used to automate the CFD simulation process to repeatedly generate new simulations for different boundary conditions. Furthermore, the training data could be selected to contain more examples of flow that the SPM struggles with such as near walls or edges. SPM generalisability could be improved by using Physics Informed Neural Networks which constrain the predictions of the SPM to those that align with physical principles. These additions may make the the SPM slightly more computationally expensive but could also offer substantial increases in generalisability.

V. Conclusion

This research highlights the effectiveness of a Deep Learning-based Smart Proxy Model (SPM) in accurately predicting indoor velocity flow fields. The SPM extracts information from a room such as the building dimensions and location of each air inlet and outlet which it uses as proxies for the simulation. A simple ANN is used to segment the flow field into the respective flow regimes. The segmented dataset is then fed in to four seperate Deep Artificial Neural Networks (ANNs), each designed to forecast turbulent, transitional, laminar, and stagnant flows with respective mean absolute percentage errors (MAPE) of 1.9%, 1.84%, 4.58%, and 65.80%. Crucially, the SPM completes simulations for a typical medium-sized office within 0.8 seconds on a personal laptop with a Core i7 processor, contrasting Computational Fluid Dynamics (CFD) with a 5-hour duration on Durham University's Hamilton 8 supercomputer with 256GB of RAM and three compute nodes.

References

[1] Durham-University, "Hamilton8-sumpoercomputer." [Online]. Available: https://www.durham.ac.uk/research/institutes-and-centres/advanced-research-computing/hamilton-supercomputer/systems/

[2] L. S. X. L. O. Y. A. F. T. D. C. X. R. N. Xin Zhang, Jianfeng Wu, "Monitoring sars-cov-2 in air and on surfaces and estimating infection risk in buildings and buses on a university campus," *Journal of Exposure Science Environmental Epidemiology*, 2022.

[3] B. K. G. S. Aliabadi AA, Rogak SN, "Preventing airborne disease transmission: review of methods for ventilation design in health care facilities," 2011.

[4] D. Bhatia and A. De Santis, "A preliminary numerical investigation of airborne droplet dispersion in aircraft cabins," *Open Journal of Fluid Dynamics*, vol. 10, no. 3, pp. 198–207, 2020.

[5] C. Ren, H. Yu, J. Wang, H.-C. Zhu, Z. Feng, and S.-J. Cao, "Zonal demand-controlled ventilation strategy to minimize infection probability and energy consumption: A coordinated control based on occupant detection," *Environmental Pollution*, vol. 345, p. 123550, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0269749124002641

[6] "Bringing embodied carbon upfront." [Online]. Available: https://worldgbc.org/advancing-net-zero/embodied-carbon/

[7] W. Wu, B. Wang, A. Malkawi, N. Yoon, Z. Sehovic, and B. Yan, "A method toward real-time cfd modeling for natural ventilation," *Fluids*, vol. 3, no. 4, 2018. [Online]. Available: https://www.mdpi.com/2311-5521/3/4/101

[8] L. J. . K. S. J. Jeon, J., "Finite volume method network for the acceleration of unsteady computational fuid dynamics: Non-reacting and reacting fows," *Int. J. Energy Res*, 2022.

[9] T. Stevens, B. Colonius, "Enhancement of shock-capturing methods via machine learning," *Teor. Comput. Fluid Dyn. 34, 483–496*, 2020.

[10] D. e. a. Kochkov, "Machine learning-accelerated computational fuid dynamics." *Proc. Natl Acad. Sci. USA 118*, 2021.

[11] K. S.-L. Kudzai Chipongo, Mehdi Khiadani, "Comparison and verification of turbulence reynolds-averaged navier–stokes closures to model spatially varied flows," *Scientific Reports*, 2020.

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

[13] C. Jiang, R. Vinuesa, R. Chen, J. Mi, S. Laima, and H. Li, "An interpretable framework of data-driven turbulence modeling using deep neural networks," *Physics of Fluids*, vol. 33, no. 5, p. 055133, 05 2021. [Online]. Available: https://doi.org/10.1063/5.0048909

[14] N. Georgiadis, D. Rizzetta, and C. Fureby, "Large-eddy simulation: Current capabilities, recommended practices, and future research," *AIAA Journal*, vol. 48, pp. 1772–1784, 08 2010.

[15] A. Beck, D. Flad, and C.-D. Munz, "Deep neural networks for data-driven les closure models," *Journal of Computational Physics*, vol. 398, p. 108910, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999119306151

[16] d. L. H. . K. P. Novati, G., "Automating turbulence modelling by multi-agent reinforcement learning." *Nat Mach Intell 3*, 2021.

[17] K. Lee and K. T. Carlberg, "Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders," *Journal of Computational Physics*, vol. 404, p. 108973, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999119306783

[18] A. Aboaba, Y. Martinez, S. Mohaghegh, M. Shahnam, C. Guenther, and Y. Liu, "Smart proxy modeling application of artificial intelligence machine learning in computational fluid dynamics." [Online]. Available: https://www.osti.gov/biblio/1642460

[19] A. Shahkarami, , and S. Mohaghegh, "Applications of smart proxies for subsurface modeling," *Petroleum Exploration and Development*, vol. 47, no. 2, pp. 400–412, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S187638042060057X

[20] M. G. O. T. C Wui, A Ghahfarokhi, "Smart proxy modeling of a fractured reservoir model for production optimization: Implementation of metaheuristic algorithm and probabilistic application," *Natural Resources Research*, 2021.

[21] S. A. Yousof Haghshenas, Mohammad Emami Niri and R. A. Kolajoobi, "Developing grid-based smart proxy model to evaluate various water flooding injection scenarios," *Petroleum Science and Technology*, vol. 38, no. 17, pp. 870–881, 2020. [Online]. Available: https://doi.org/10.1080/10916466.2020.1796703

[22] M. Bussov and J. Nättilä, "Segmentation of turbulent computational fluid dynamics simulations with unsupervised ensemble learning," *Signal Processing: Image Communication*, vol. 99, p. 116450, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923596521002150

[23] H. Yu, J. Lou, H. Liu, Z. Chu, Q. Wang, L. Yang, and Y. Rao, "A transfer learning method to assimilate numerical data with experimental data for effusion cooling," *Applied Thermal Engineering*, vol. 224, p. 120075, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1359431123001047

[24] Y. M. Q. W. Yingjie Tian, Kun Zhao, "A comprehensive survey of loss functions in machine learning," *Annals of Data Science*, 2022.

[25] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, 2016, advances in artificial neural networks, machine learning and computational intelligence. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231216003325

[26] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *International Journal of Forecasting*, vol. 32, no. 3, pp. 669–679, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0169207016000121

[27] ASHRAE, "Thermal enviornmental conditions for human occupancy," ASHRAE, Tech. Rep. ISSN 1041-2336, 2020. [Online]. Available: https://www.ashrae.org/technical-resources/bookstore/standard-55-thermal-environmental-conditions-for-human-occupancy

[28] ASHRAE, "Ventilation for acceptable indoor air quality," ASHRAE, Tech. Rep. ISSN 1041-2336, 2016.