Physics-Informed Neural Networks for Passive Scalar Emission and Transport

Joshua Ian Rawden,* Christina Vanderwel, and Sean Symon

Aerodynamics & Flight Mechanics Research Group,

University of Southampton, Southampton SO17 1BJ, UK

(Dated: January 19, 2026)

Abstract

Accurate modeling of harmful pollutant concentrations is an important field of interest for protecting public health and the environment. In this study, physics-informed neural networks (PINNs) are applied to a low Reynolds number, time-averaged cylinder wake, which interacts with a variety of different passive scalar regimes. The PINN reconstructs both the time-averaged velocity and scalar concentration from limited measurements. In addition to satisfying the incompressible Reynolds-averaged Navier-Stokes (RANS) equations, the reconstructed fields must also obey the time-averaged advection-diffusion equation. This was done to extend the applicability of mean field reconstruction and to lay the foundations for PINNs to be used in more complex passive scalar modeling in future studies, particularly the prediction of pollutant behavior in urban environments. It was found that the PINN could successfully reconstruct the flow fields on a macro-scale in almost all scenarios, having considerable success with first-order quantities and managing to accurately infer the spatial structure of the unknown closure term of the advection-diffusion equation in all cases. When reconstructing scalar source characteristics, the PINN could identify the source location and size in many cases. The results allowed basic guidelines for optimal sensor placement to be described to inform future studies, and suggests ways in which this technique could be developed to contribute to future air quality forecasting models.

* jr1g20@soton.ac.uk

I. INTRODUCTION

A passive scalar is any concentration that does not influence the physical behavior of a fluid, meaning the velocity fields of a flow with and without a passive scalar are identical. In recent years, the importance of understanding how passive scalars are transported and dispersed in urban environments has increased tremendously. This includes the way we model the transfer of heat, particulates, and disease within closed spaces, as well as outdoor public health in ever-growing cities as the world continues to industrialize. According to the World Health Organization (WHO), 99% of the population breathe air that exceeds WHO guideline limits [1]. The first step toward reducing this number is developing the ability to reliably model the evolution of pollutant plumes. To do so, accurate reconstruction of the passive scalar properties at/near the source(s) is critical, as an understanding of the origin and typical paths of harmful particulates and gases in the atmosphere allows public health measures to be enacted by authorities. These would aim to minimize the impact on the public and appropriately direct the enforcement of environmental protection laws when necessary.

Experiments and high-fidelity computational fluid dynamics (CFD) simulations have been performed to improve understanding of passive scalars in urban pollution, such as the simulations of Refs. [2–4] and the scaled experiments of Refs. [5–7]. While experiments provide physically justified measurements at a discrete number of sampling points, and CFD methods give a modeled solution on a dense mesh, both approaches are expensive and time-consuming to set up while having their own sets of limitations. Sparse field measurements and the complexity of large-scale urban geometries present challenges due to the sharp spatial variations of pollutant concentrations, making optimal placement of sensors difficult [8, 9]. This presents an opportunity for data-driven flow reconstruction methods to demonstrate their effectiveness where the use of CFD or experimentation alone is challenging. The most established data-driven method used in mean-flows is variational data assimilation, which uses observations from experimental/high-fidelity CFD to inform and improve the results of a cheap, low-fidelity CFD simulation (e.g. RANS), often reconstructing features and multi-scale structures not present in either dataset separately [10, 11]. Data assimilation is a fundamental component of modern weather models, and is employed in a variety of roles from large-scale storm monitoring to regional-scale pollution modeling [12, 13]. Data assim-

ilation has also been a useful tool for reconstructing passive scalar fields on smaller scales [9, 14–17], but its effectiveness is highly dependent on the locations of individual sensors and the monitoring paradigm being used, especially when reconstructing source characteristics as shown by Ref. [9].

Physics-informed neural networks (PINNs) are a recent development in the flow reconstruction community, leveraging the rapid progress in machine learning technologies to create a new class of neural network that has a physical understanding of the problem it is solving embedded within its objective function. PINNs were formally introduced by Ref. [18], where they were used in a variety of physical problems to demonstrate their novelty and flexibility. More recently, Ref. [19] used PINNs to reconstruct the mean fields and turbulence statistics around a low-Reynolds number cylinder flow. Reference [20] introduced the Spalart-Allmaras turbulence model to the PINN when solving a periodic hill flow problem where it was found to outperform traditional data assimilation methods. Reference [21] applied PINNs to DNS data of an unsteady, buoyancy-driven Rayleigh-Bénard flow and had success with reconstructing the temperature field. Reference [22] apply PINNs to an urban environment, but did not consider any passive scalar. Several other studies have been performed that also attempt to solve fluid-based problems using PINNs, suggesting that there is an appetite in this technology to push the field of fluid dynamics forward [23–30].

The novelty of this study is to identify the size and location of unknown passive scalar sources by reconstructing the mean velocity and concentration fields with limited measurements using PINNs. This is performed on a well-researched, canonical flow case under a wide range of passive scalar regimes defined using different boundary conditions (BCs) and Schmidt numbers. The PINN's effectiveness will be determined by its ability to reconstruct the mean velocities, the mean scalar concentration, the unknown closure field of the advection diffusion equation, and the properties of the scalar source. If the PINN is successful in completing these objectives, it demonstrates that the methodology may be applicable to more complex problems in the near future. It will also suggest that PINNs can complement with the established variational data assimilation techniques for tackling these problems and may play a foundational role in future air quality forecasting frameworks [31].

The rest of the paper is organized as follows. Section II introduces the physical problem and equations, as well as the mathematics and principles that govern the PINN. Section III describes the generation of the CFD reference data used to train the PINN. Section

IV describes the PINN models in more detail, listing the architectures and training procedures used. Section IV D summarizes the variations in the independent variables for the cases, followed by Sec. V which presents the results. Finally, Sec. VI concludes the study, summarizing the reconstructive qualities of PINNs in passive scalar problems.

II. PHYSICS & PROBLEM DEFINITION

This section will introduce the physical problem, beginning with introducing the governing equations in Sec. II A, the time-averaged versions of which are presented and described more rigorously for the passive scalar and velocity in Secs. II B and II C, respectively, followed by an explanation of the PINN's loss function in Sec. II D.

A. Governing Equations

We start with the vector form of the advection-diffusion equation and the full incompressible Navier-Stokes equations

$$\frac{\partial c}{\partial t} + \boldsymbol{u} \cdot \nabla c - \frac{1}{Re \cdot Sc} \cdot \nabla^2 c = 0, \tag{1a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1b}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \nabla p - \frac{1}{Re} \cdot \nabla^2 \boldsymbol{u} = 0, \tag{1c}$$

where $\boldsymbol{u}$ is velocity, p is pressure, c is passive scalar concentration, $Re = \frac{UL}{\nu}$, and $Sc = \frac{\nu}{\alpha}$. U is the characteristic velocity, L is a characteristic length scale, ν is the kinematic viscosity of the fluid, and α is the molecular diffusivity of the passive scalar. The Schmidt number represents the ratio between momentum diffusivity and molecular diffusivity. Certain flows concerning small heat fluctuations may also be considered passive scalar flows, as buoyancy effects are negligible. In this context, the Prandtl number can be used instead of the Schmidt number. A flow with a high Sc will generally result in a field with higher gradients of scalar concentration, as it follows the velocity field more strictly. In contrast, a low Sc allows the scalar to spread regardless of the advective fluxes, resulting in a generally more diffuse and homogeneous concentration field. For some physical intuition, CO_2 has a molecular Schmidt number in air of 0.94, and air itself has a Prandtl number of 0.708 [32]. Naphthalene (an

organic compound used to make plastics and pesticides) has a higher molecular Schmidt number of 2.35 [33]. These values inform the ranges used in this study.

B. Time-Averaged Advection-Diffusion Equation

To transform the system into a set of time-averaged equations, we use the Reynolds decomposition

$$\boldsymbol{u} = \overline{\boldsymbol{u}} + \boldsymbol{u'}, \tag{2}$$

applied to u, v, p and c, where u is the streamwise velocity and v is the transverse velocity. Here, the $\overline{[\ ]}$ terms are the mean flow quantities and the $[\ ]'$ terms are the fluctuations from the mean flow.

Applying Eq. 2 to Eq. 1a and averaging yields the time-averaged advection-diffusion equation

$$\mathcal{P}_1 = \overline{u} \cdot \frac{\partial \overline{c}}{\partial x} + \overline{v} \cdot \frac{\partial \overline{c}}{\partial y} - \frac{1}{Re \cdot Sc} \cdot \left(\frac{\partial^2 \overline{c}}{\partial x^2} + \frac{\partial^2 \overline{c}}{\partial y^2} \right) + g, \tag{3a}$$

$$g = \nabla \cdot (\overline{\boldsymbol{u'} c'}) = \frac{\partial \overline{u'c'}}{\partial x} + \frac{\partial \overline{v'c'}}{\partial y}, \tag{3b}$$

where g is an unknown closure term, as it requires knowledge of the fluctuations and is defined as the divergence of the mean turbulent scalar flux. Here, we assume a 2D problem (explained in Sec. III), hence only 2 components of velocity are present. In Eq. 3a, the first two terms are the scalar advection terms, and the 3rd is the diffusion term. Since g depends on the turbulent scalar fluxes $\overline{u'c'}$ and $\overline{v'c'}$ (which are absent from the mean flow data), any reconstruction of the macro-level structure of g indicates that the PINN is inferring the divergence of the fluxes from the mean concentration fields, which is a useful property for turbulence modeling. The notation $\mathcal{P}_i$ indicates that the equation will be reduced to a residual which the PINN uses to inform its gradient descent algorithm (see Sec. II D).

C. RANS Equations with Solenoidal Forcing

Similarly to Eq. 3a, applying Eq. 2 to Eqs. 1b and 1c then averaging transforms them into

$$\mathcal{P}_2 = \frac{\partial \overline{u}}{\partial x} + \frac{\partial \overline{v}}{\partial y}, \tag{4a}$$

$$\mathcal{P}_3 = \overline{u} \cdot \frac{\partial \overline{u}}{\partial x} + \overline{v} \cdot \frac{\partial \overline{u}}{\partial y} + \frac{\partial (\overline{p} - \phi)}{\partial x} - \frac{1}{Re} \cdot \left(\frac{\partial^2 \overline{u}}{\partial x^2} + \frac{\partial^2 \overline{u}}{\partial y^2} \right) + f_{su}, \tag{4b}$$

$$\mathcal{P}_4 = \overline{u} \cdot \frac{\partial \overline{v}}{\partial x} + \overline{v} \cdot \frac{\partial \overline{v}}{\partial y} + \frac{\partial (\overline{p} - \phi)}{\partial y} - \frac{1}{Re} \cdot \left(\frac{\partial^2 \overline{v}}{\partial x^2} + \frac{\partial^2 \overline{v}}{\partial y^2} \right) + f_{sv}. \tag{4c}$$

In addition to the mean-flow transformation, new variables ϕ, f_{su} and f_{sv} have been introduced. The terms f_{su} and f_{sv} are derived from the vector $\boldsymbol{f} = \nabla \cdot \boldsymbol{R}$ (which emerges after Reynolds-averaging) through the process described below. Tensor $\boldsymbol{R}$ contains the Reynolds stress components $R_{ij} = \overline{u_i' u_j'}$, and so taking Eqs. 4 and 3a, we have 4 equations for 9 unknowns: $\overline{c}$, $\overline{u}$, $\overline{v}$, $\overline{p}$, $\overline{u'u'}$, $\overline{v'v'}$, $\overline{u'v'}$, $\overline{u'c'}$ and $\overline{v'c'}$. This is an underdetermined system which cannot be solved for a unique solution.

To resolve this, a modified formulation of the RANS equations is to be used, which was first proposed by Ref. [10] for use in variational data assimilation algorithms, and has since been used successfully in studies of mean flow with PINNs [19, 20]. This modification performs a Helmholtz decomposition on $\boldsymbol{f}$, separating it into a potential component and a solenoidal (divergence-free) component:

$$\boldsymbol{f} = \nabla \phi + \boldsymbol{f_s}, \tag{5a}$$

$$\nabla \cdot \boldsymbol{f_s} = 0. \tag{5b}$$

The potential forcing ϕ is combined with the pressure gradient term of equations 4b and 4c. As described in detail in Sec. 2.6 of Ref. [10], this results in the true pressure field being unrecoverable after reconstruction; therefore, the reconstructed pressure fields will be omitted from the results.

The addition of Eq. 5b to the roster as:

$$\mathcal{P}_5 = \frac{\partial f_{su}}{\partial x} + \frac{\partial f_{sv}}{\partial y}, \tag{6}$$

increases the count to 5 equations for 9 unknowns, which is still under determined. One way to make the PDE system tractable is to provide the $\overline{u}$, $\overline{v}$ and $\overline{c}$ fields on a sufficiently resolved grid, allowing the velocity data to fully satisfy Eq. 4a. This reduces the system to 4 equations (Eqs. 4b, 4c, 5b and 3a) for 5 unknowns: $\overline{p} - \phi$, f_{su}, f_{sv}, $\overline{u'c'}$ and $\overline{v'c'}$. As discussed

in [19] and [20], a sufficiently dense velocity data grid will close the RANS equations and result in a unique solution; the expectation is that this principle can also be extended to $\overline{c}$ for the advection-diffusion system. However, this is infeasible in realistic applications and many experimental techniques. Therefore, we rely on the deterministic nature of the idealized velocity and passive scalar fields to allow the amount of necessary data to be reduced to a practical amount while still producing an accurate reconstruction.

D. PINN Problem

With some generalization, neural networks aim to "learn" a relationship between pairs of input/output data with or without training data serving as a reference. For PINNs, this is effectively the minimization of an objective loss function:

$$\mathcal{L}(\boldsymbol{\theta};\boldsymbol{y}) = w_D\mathcal{L}_D(\boldsymbol{\theta};\boldsymbol{y}) + w_B\mathcal{L}_B(\boldsymbol{\theta};\boldsymbol{y}) + w_P\mathcal{L}_P(\boldsymbol{\theta};\boldsymbol{y}), \tag{7a}$$

$$\mathcal{L}_D(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{i=1}^{N}\left[\frac{1}{|\boldsymbol{x^D}|}\sum_{j=1}^{|\boldsymbol{x^D}|}\left[y_i(\boldsymbol{x^D}_j) - y^D_{i,j}\right]^2\right], \tag{7b}$$

$$\mathcal{L}_B(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{i=1}^{N}\left[\frac{1}{|\boldsymbol{x^B}|}\sum_{j=1}^{|\boldsymbol{x^B}|}\left[y_i(\boldsymbol{x^B}_j) - y^B_{i,j}\right]^2\right], \tag{7c}$$

$$\mathcal{L}_P(\boldsymbol{\theta};\boldsymbol{y}) = \sum_{i=1}^{N}\left[\frac{1}{|\boldsymbol{x^P}|}\sum_{j=1}^{|\boldsymbol{x^P}|}\left[\mathcal{P}_i|_{\boldsymbol{x^P_j}}\right]^2\right], \tag{7d}$$

with respect to the set of network parameters $\boldsymbol{\theta}$, where $\boldsymbol{y}$ is the vector of N reconstructed fields $\boldsymbol{y} = \begin{bmatrix} y_1(\boldsymbol{x}) & y_2(\boldsymbol{x}) & ... & y_N(\boldsymbol{x}) \end{bmatrix}^T$ at each domain coordinate $\boldsymbol{x} = \begin{bmatrix} x & y \end{bmatrix}^T$, and $\boldsymbol{x}$ can represent the set of data point, boundary point or collocation point coordinates ($\boldsymbol{x^D}$, $\boldsymbol{x^B}$, $\boldsymbol{x^P}$) depending on which loss term the PINN is solving for. Here, $\mathcal{L}_D(\boldsymbol{\theta};\boldsymbol{y})$ represents the data loss, i.e. the summation of L_2 errors between the training and reconstructed values across each data point within the PINN domain [19, 34] where y^D is the reference data field. Next, $\mathcal{L}_B(\boldsymbol{\theta};\boldsymbol{y})$ is the boundary loss with y^B as the boundary condition values for each field, which sums the L_2 errors across the domain boundary to enforce the specified boundary conditions. Finally, $\mathcal{L}_P(\boldsymbol{\theta};\boldsymbol{y})$ is the physics loss, with $\mathcal{P}_i$ as the residual for each governing equation evaluated at each collocation point within the domain. $\mathcal{L}_P(\boldsymbol{\theta};\boldsymbol{y})$ is the primary means by which the PINN is able to enforce the physics of the problem when producing

a solution. w_D, w_B and w_P are the respective data, boundary and physics loss weights, which can be specified by the user to bias the PINN to conform more toward the data or the equations. Training to minimize $\mathcal{L}(\boldsymbol{\theta}; \boldsymbol{y})$ is performed iteratively through numerical optimization of the parameter space.

III. GENERATION OF CFD REFERENCE DATA

OpenFOAM [35, 36] was used to simulate a canonical 2D cylinder flow. This flow is well described in literature and serves as the foundation onto which the novel PINN is applied. Three different flow cases were considered, the boundary conditions/initial conditions (BCs/ICs) for which are given in Sec. III B: a cylinder with passive scalar emission from the surface (hereafter referred to as the *Surface* case), a cylinder with passive scalar emission from a point source upstream of the cylinder (hereafter referred to as the *Upstream* case), and a cylinder with passive scalar emission from a point source within the recirculation region of the cylinder wake (hereafter referred to as the *Wake* case). From these three cases, datasets were generated for each case at Schmidt numbers of 0.1, 0.3, 1.0, 3.0, and 10.0 to give variation in the diffusive behavior of the scalar.

A. Domain & Meshing

The CFD domain extends from $-15 < \frac{x}{d} < 40$, $-10 < \frac{y}{d} < 10$ with a cylinder obstacle of diameter $d = 1$ located at (0, 0). The z dimensionality of the domain is superficial, as the mesh only contains a single cell along the z axis, and the front/back boundary conditions are set to empty. 'Empty' is a type of boundary condition in OpenFOAM reserved for reduced-dimension cases, as OpenFOAM can only operate on 3D domains [37]. This boundary condition type, along with the single cell in the z-direction, enforces uniformity in the spanwise solution and makes the domain effectively 2D. The mesh was generated using BlockMesh within OpenFOAM. The mesh is composed of 6 individual blocks that were refined independently: the pre-block, 4 obstacle blocks surrounding the cylinder, and the post-block. The obstacle blocks were the only blocks to be refined, and were refined using the simpleGrading function with a refinement factor of 80 approaching the cylinder surface. This resulted in a total cell count of 51,000. BlockMesh's internal mesh checking function

verified that the cell aspect ratios and skewnesses are satisfactory.

For the surface case, the passive scalar is injected into the domain by specifying a Dirichlet boundary condition of magnitude 1.0 on the cylinder surface (see Sec. III B). The point source upstream and wake cases use an additional OpenFOAM class called a cellZone to apply special properties to a group of cells without imposing strict boundary conditions that influence other properties of the flow. These cells are assigned a constant value of 5.0 for the scalar concentration which is maintained throughout the simulation. For the upstream case, the cellZone was placed at (-1.6, 0.3) and applied to 2 cells. For the wake case, the cellZone was placed at (1.0, 0.3) and applied to 3 cells. This ensures that the point source cases produce scalar concentration fields with sufficient magnitude to be reconstructed by the PINN, since the total number of cells emitting passive scalar is far lower than in the surface emission case. The difference in source strength does not affect the results since the investigation that compares these cases is concerned with the impact of specifying BCs and not the strength of the source.

B. Boundary/Initial Conditions & Solver

The solver and solution parameters for all three cases were the same except for details of the scalar injection. The pisoFoam solver within OpenFOAM was used to solve the unsteady Navier-Stokes equations transiently without a turbulence model. The solver used a second-order backward time differentiation scheme, a Gauss linear gradient scheme, and a Gauss linear corrected Laplacian scheme. For all cases, a uniform inlet is prescribed. No-slip Dirichlet boundary conditions are applied to the cylinder surface, symmetric boundary conditions at the upper and lower boundaries, and advective conditions to the outlet. The Reynolds number based on the diameter was $Re = 150$.

The solution was advanced in time up to 500 units with $\Delta t = 0.001$ to give 500,000 individual time steps. At a freestream velocity of 1, this corresponded to a maximum CFL number of around 0.15 throughout the simulation which ensured temporal stability and justified the use of the backward time scheme. In the surface case, the scalar was emitted from the cylinder surface at a constant rate throughout the entire simulation. In the upstream and wake cases, the point-source emission cellZone was switched on at $t = 100$ to allow the velocity fields to settle into a periodic vortex shedding state before the scalar

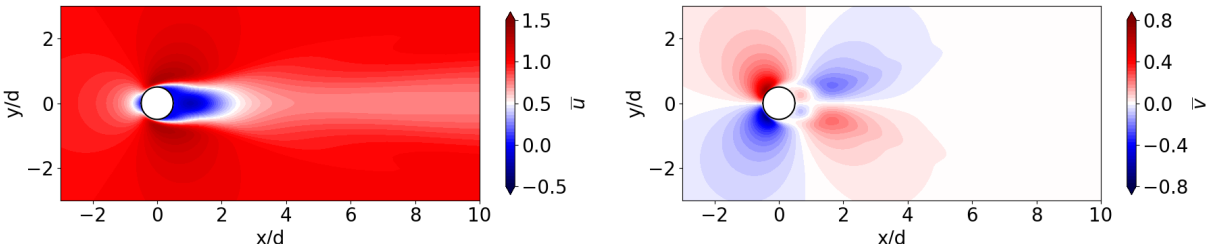


FIG. 1: Reference fields of $\overline{u}$ and $\overline{v}$.

was introduced.

Following the simulations, the data were time-averaged to provide mean fields for the PINN. Time-averaging was performed after the fluid had reached periodic stability. This was found to be at $t \approx 150$; therefore, averaging was performed for 350 snapshots at intervals of 1 between $150 < t < 500$. The g fields were computed using the turbulent scalar fluxes taken from the transient data from this same period, and the results were then averaged to give g as per Eq. 3b. To validate the CFD data, the frequency of shedding was calculated to be $St = \frac{fL}{U} = 0.186$, which is the expected frequency for $Re = 150$ [38]. The mean fields were confirmed to be in agreement with those of other studies [10, 19], and are presented in Figs. 1 and 2. Note that only $\overline{u}$, $\overline{v}$, and $\overline{c}$ are explicitly given to the PINN, as g is an inferred term and will be reconstructed by the PINN.

IV. DETAILS OF PINN MODELS & EXPERIMENTS

This section introduces the procedure used to create the PINN models used throughout the investigation, giving their architectures and hyperparameters, the training protocol, and the independent variables that were varied over the course of the study.

A. PINN Model Generation Program

The program used to generate the PINN models was written in Python and was developed from the code made publicly available in Ref. [19]. The program makes use of the DeepXDE package [34] - a library that interfaces with commonly used deep-learning packages.

To provide the PINN with the reference CFD training data, the data at each cell within the CFD mesh were linearly interpolated to a uniform grid with spacing h, which indicates

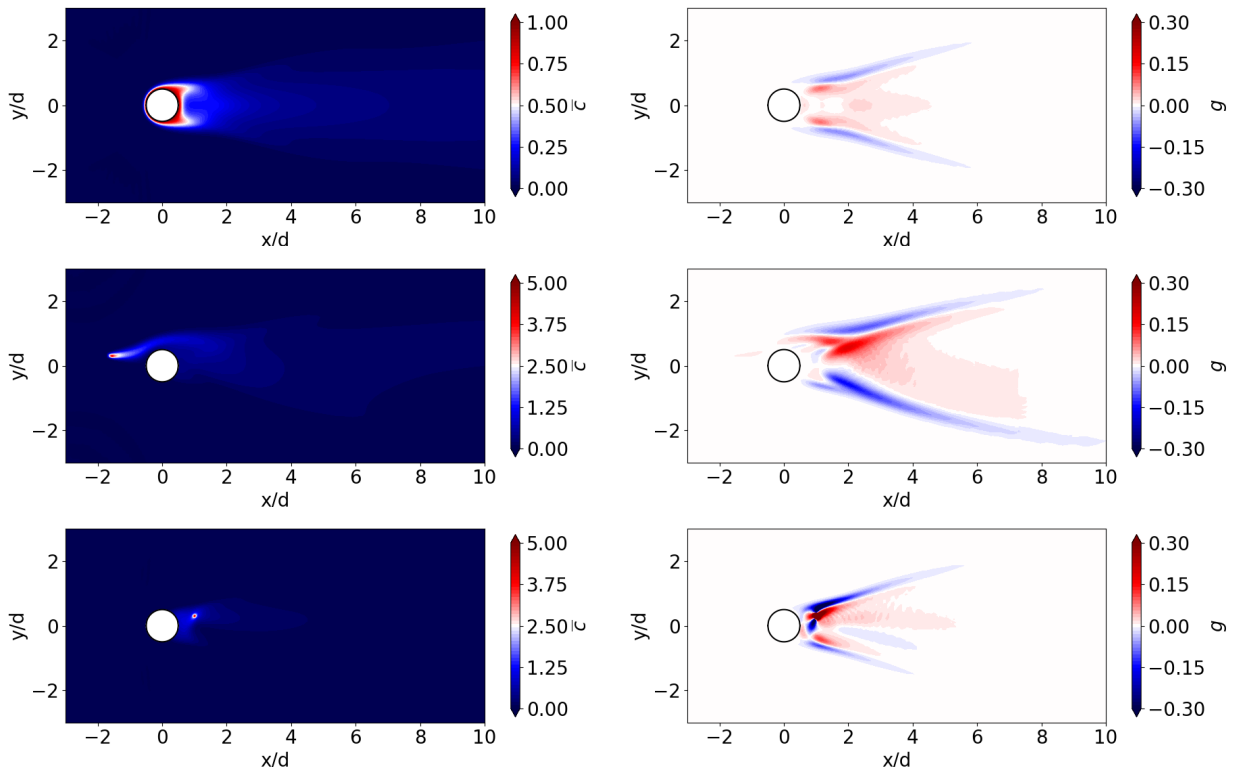


FIG. 2: Surface, upstream and wake case CFD fields at $Sc = 1$; $\overline{c}$ provided as a reference field to the PINN and g to be reconstructed by the PINN.

the number of points per cylinder diameter, e.g. $h = 2$ represents a data point spacing of $d/2$. h with no subscript indicates the velocity and scalar data were supplied on the same grid. The PINN models used the following BCs for the cylinder surface for all cases: Dirichlet no-slip conditions for $\overline{\boldsymbol{u}}$ and $\boldsymbol{f_s}$, Neumann conditions for g, Dirichlet $\overline{c} = 1$ in the surface case, and Neumann conditions for $\overline{c}$ in the upstream and wake cases. All models used the same 5 residual equations within their partial differential equation (PDE) models with partial derivatives computed using automatic differentiation (AD). This allows the PINN to be continuous and differentiable, meaning that no underlying grid/mesh is necessary.

B. PINN Architecture & Parameters

All PINN models used the same architecture: an input layer of width 2, 7 hidden layers of width 100, and an output layer of width 7, which maps $\begin{bmatrix} x & y \end{bmatrix}^T \rightarrow \begin{bmatrix} \overline{u} & \overline{v} & \overline{p} - \phi & \overline{c} & f_{su} & f_{sv} & \overline{g} \end{bmatrix}^T$. The impact of varying the number of nodes per layer and number of layers is presented in

the appendix, but the network size was the same for all models considered in the results section. Each layer used the *tanh* activation function and the *Glorot uniform* initialiser function. The relative loss weights were the same as in Ref. [20], and were set as: $w_D = 10$, $w_B = 10$, $w_P = 1$.

The full domain collocation points were defined using a custom function to cluster points near the cylinder surface and along the flow centerline. This was done using a lognormal distribution in x and a normal distribution in y. Figure 3 shows the extent of the full PINN domain and the collocation point distribution described above, with the dimensions normalized by the cylinder diameter d. The custom distribution is expected to improve convergence as the PINN can more effectively resolve the PDEs in regions of the domain that experience high gradients. The number of domain collocation points was around 58,000, and the number of boundary collocation points was set to 1500. 50 testing points were randomly selected to update the loss function each iteration to give a fairly uniform distribution throughout the domain without hindering the optimizer loop with excessive passes. The testing points do not influence the loss function seen by the optimizer and therefore do not inform the PINN training process. Including testing collocation points is useful for monitoring the testing loss to determine whether the PINN model is overfitting to the training data and failing to adhere to the physics, which was not an issue for any models used in this study.

C. PINN Training Protocol

All models were trained in two stages: an initial stage using the *Adam* optimizer [39] which is based on stochastic gradient descent, followed by a secondary longer training stage using the *L-BFGS-B* optimizer [40, 41]. This order was decided due to *Adam's* ability to explore and smoothen the PINN's rough initial parameter space, which prepares the model for a longer training period with *L-BFGS-B* to significantly reduce the losses [34]. This procedure is consistent with other studies that investigated similar fluid cases [19, 20, 24, 25, 42]. The PINN was trained using *Adam* for 25,000 epochs at a learning rate of 0.001, followed by *L-BFGS-B* up to a maximum of 60,000 total epochs, or until the optimizer converged. The convergence criterion is set as the DeepXDE default of $\leq 1 \times 10^{-8}$ for the maximum component of the projected descent gradient. A typical loss graph is given in Fig. 4. Training

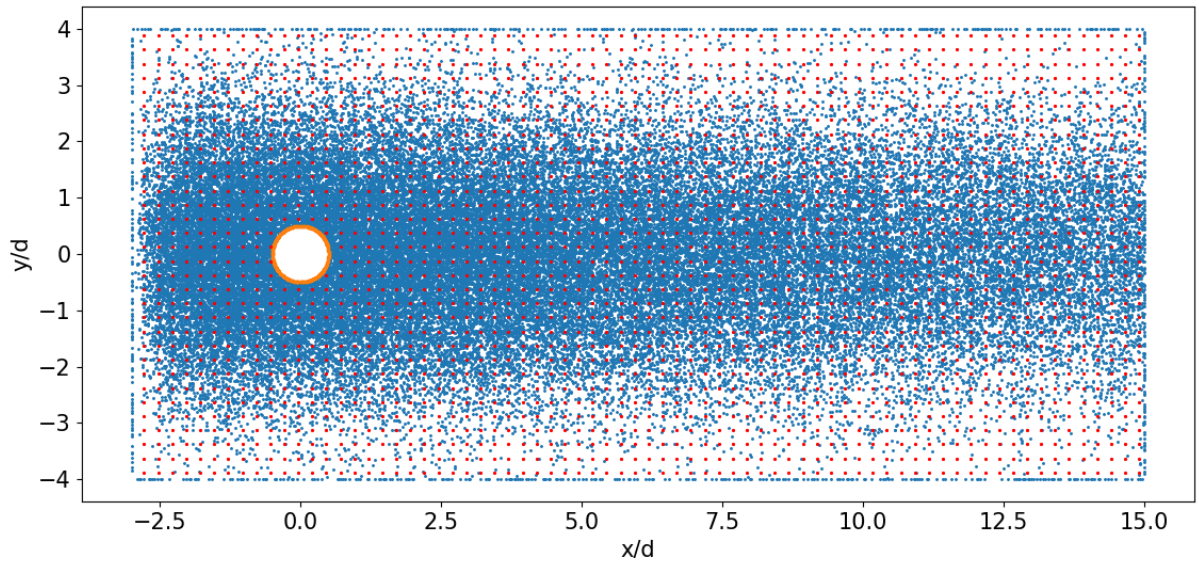


FIG. 3: Collocation and typical data point distribution: domain points (blue), boundary points (orange), data points (red).

was performed using the *Iridis X* High-Performance Computing cluster at the University of Southampton, utilizing NVIDIA A100 enterprise GPUs. The total training time was heavily dependent on the number of data points presented to the PINN, but a typical training cycle for a single model was on the order of 2-4 hours. In total, the time taken to train all models used in this investigation was around 7-9 days. All investigations required many models to be trained with slight variations to the network conditions. This process was automated using a Python loop.

The code used to train the PINN models and some example models used in this study can be found at `https://github.com/roardon/PassiveScalarPINNs` or at `https://doi.org/10.5258/SOTON/D3810`.

D. Experimental Conditions & Independent Variables

The investigations performed can be divided into two categories: investigation of the full domain and investigation of the cropped domain as seen in Fig. 5. The separation of the investigations allows reconstruction of the macro-level structure of the fields to be considered independently of the identification of the source characteristics. The full domain investigations were performed on all three flow cases and consider a smaller region of the CFD domain by cropping unimportant (freestream) parts of the flow. This was done to

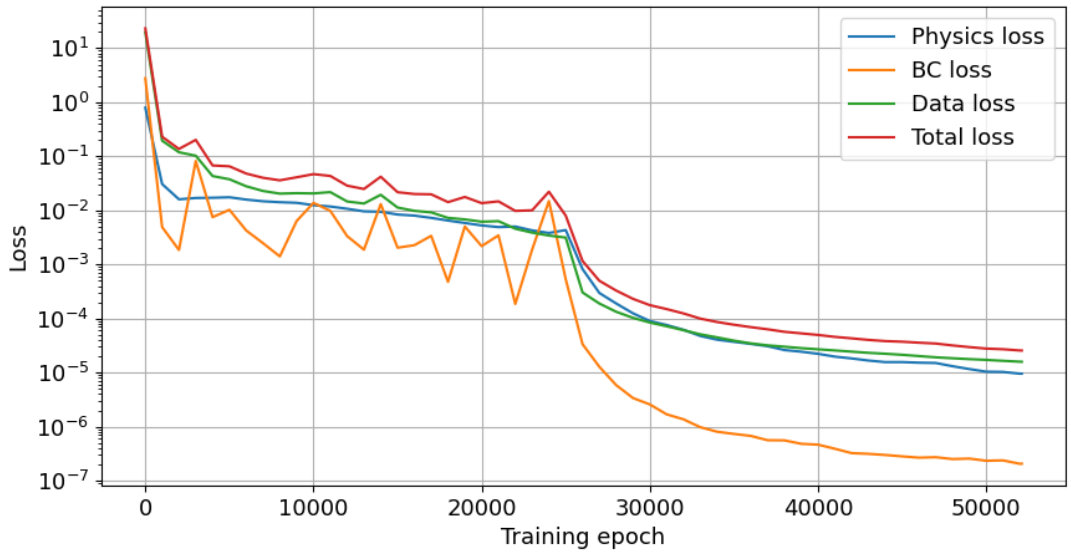


FIG. 4: Typical loss throughout the training cycle; optimiser switches from *Adam* to *L-BFGS-B* at 25,000 epochs.

allow the data/collocation points to be placed within the more sensitive areas of the PINN domain to make the most efficient use of computational resources. The full domain for the PINN extends from $-3 < \frac{x}{d} < 15$, $-4 < \frac{y}{d} < 4$ (green in Fig. 5). The investigations in the cropped domain were performed on the two point source cases (upstream & wake) and reduce the reconstruction region to a smaller window of size 1×1, with its location dependent on the flow case but ensuring that the passive scalar point source was located at the center of this window (blue in Fig. 5). In the cropped domain investigations, the scalar data were supplied within the aforementioned 1×1 window, while the velocity data were supplied within a larger window of size 6×6 from $-3 < \frac{x}{d} < 3$, $-3 < \frac{y}{d} < 3$ which includes the entire cylinder obstacle (see orange in Fig. 5). Focus was placed on identification of the source characteristics (size, location) because they represent important properties influencing the structure and evolution of a passive scalar plume. The reduced domain size was chosen because it allows for a more efficient use of computational resources, which means that more models could be trained in the same amount of time, resulting in a more robust investigation where a wider range of values for h could be tested. We found that reducing the size of the domain had no impact on the conclusions of this experiment compared to using the full domain. Passive scalar flows are highly sensitive to changes in the source properties; therefore, this knowledge is critical when attempting to accurately predict dispersion over

large, complex domains or long time horizons [9]. Within these two categories, investigations were performed to assess the impact of adjusting a single PINN parameter or flow constant on the quality of the reconstruction. Details of the cases are presented in Table I.

Case	Variable range	Variables fixed
Full-domain		
Data sparsity	$1 \leq h \leq 10$	$Sc = 1.0$
Schmidt number	$0.1 \leq Sc \leq 10$	$h = 4$
Cropped-domain		
Data sparsity	$3 \leq h_c \leq 100$	$Sc = 1.0$, $h_{\boldsymbol{u}} = 6$
Schmidt number	$0.1 \leq Sc \leq 10$	$h = 10$

TABLE I: Case details.

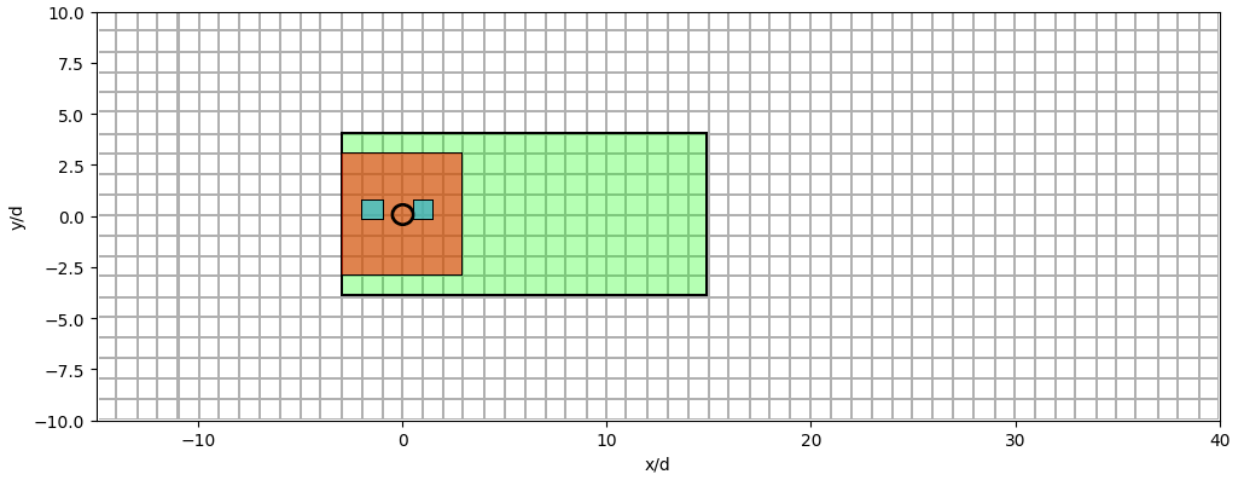


FIG. 5: Subdomains of CFD domain. Green: full PINN domain. Orange: cropped-domain velocity data range. Blue: cropped-domain scalar data ranges.

In the cropped-domain data sparsity investigation, the data were supplied on separate velocity and scalar grids whose densities could be varied independently, given by $h_{\boldsymbol{u}}$ and h_c, respectively. The 'variable range' for this investigation refers to the range of h_c used, while the velocity grid is kept constant at $h_{\boldsymbol{u}} = 6$. An investigation was performed for the inverse case (fixed h_c with varying $h_{\boldsymbol{u}}$), but this did not have any noticeable impact on scalar reconstruction, and so it will be omitted from the results. A PINN model was trained for

the combinations of parameters listed in Table I, giving a total number of models used in the results of 116.

V. RESULTS

The results will be presented as ordered in Table IV D, starting with the full-domain cases in Sec. V A, followed by the cropped-domain investigations in Sec. V B.

A. Full-Domain Investigations

Within the full-domain investigations, a volume-weighted L_2-norm error metric is used as the primary measure for assessing the accuracy of the reconstructed fields relative to the baselines. This metric is defined as:

$$\varepsilon_2 = \sqrt{\sum_{i \in \Omega} [\omega_i \times (y_i(\boldsymbol{x}_i) - y_i^D)^2]}, \tag{8}$$

where ω_i is an individual cell volume inside the domain Ω. This metric gives a single scalar value as a measure of the reconstructed field's accuracy, and is not normalized with respect to the scale of the original data. Therefore L_2 norm values for different fields (e.g. $\overline{u}$, $\overline{c}$) are not intended to be compared to one another.

1. Velocity Reconstruction & Influence of Passive Scalar Data

Since the velocity fields are identical among all cases, the influence of data sparsity on these fields is presented first in Fig. 6. Reference [19] performed a similar analysis, so this section aims in part to validate the the current PINN framework as well as to extend the application to passive scalar cases, as per the investigation objectives.

Figure 6 shows that a data grid density of $h = 1$ is inadequate for reconstructing the velocity fields around the cylinder as the PINN does not have enough information in the regions of importance to produce a unique solution. These 'regions of importance' are typically identified to be areas of the domain that exhibit high spatial gradients of the reconstructed fields. Furthermore, we have found by comparing this model to an additional model with a specified inlet BC of $\overline{u} = 1$, $\overline{v} = 0$, that this results in a substantial improvement

to the reconstruction both upstream of the cylinder and within the near-wake when $h = 1$. Despite this, the influence of specifying the inlet BCs was found to rapidly diminish beyond $h \geq 2$, suggesting that under these conditions the training data and governing equations are sufficient to produce a unique solution, a similar result to Ref. [19]. For these reasons, we believe that inlet BCs are unnecessary for most of the cases in our results and rely on external knowledge about the flow case that may not be available in practical settings, but we maintain their importance when using extremely sparse training data.

Increasing toward $h = 10$, we see greatly improved quality of reconstruction at the expense of requiring a higher measurement window resolution and increased computational resources. The velocity absolute error fields' structures and magnitudes are very similar to those for the first-order velocity statistics presented in Ref. [19] for $h = 50$. This suggests that reconstruction accuracy saturates with data at around $h = 10$, as the error reduces minimally beyond this point. We see that there is reduction in error of 2 orders of magnitude between just $1 < h < 2$. This result is consistent with Ref. [19], and we believe that this rapid increase in accuracy is due to both the convergence to a unique flow solution in the presence of additional data, as well as the PINN more accurately reconstructing the size and shape of the cylinder recirculation bubble.

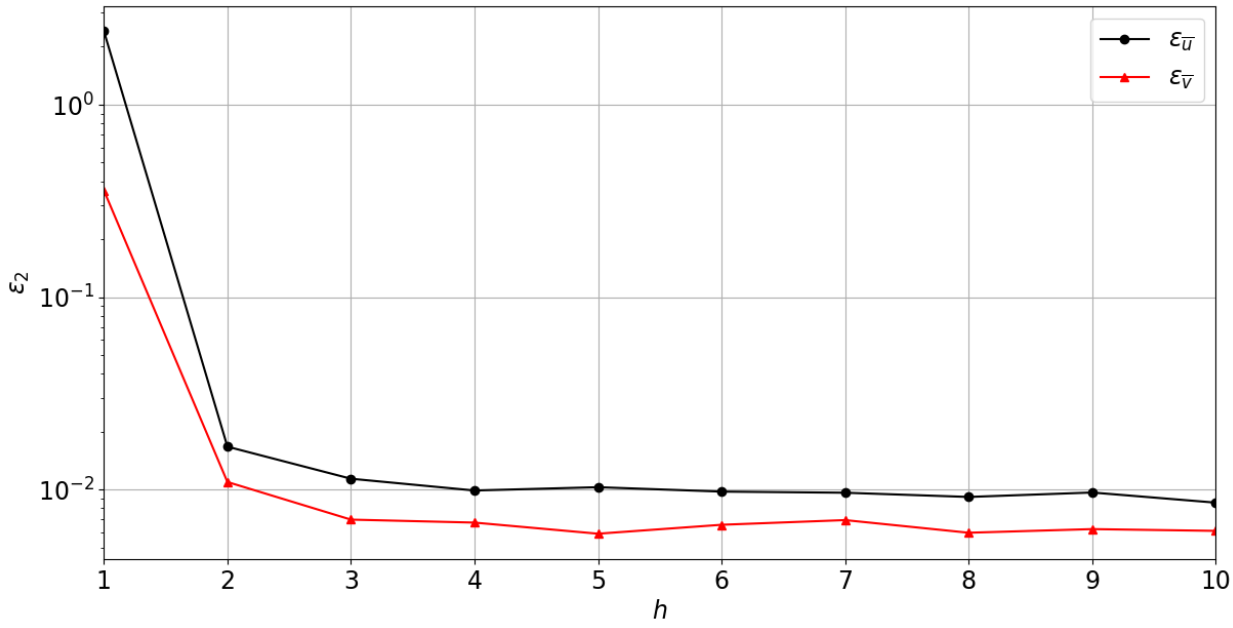


FIG. 6: L_2 norm versus h for velocity fields $\overline{u}$ and $\overline{v}$.

Next, the effects of introducing the passive scalar transport system are investigated when

applied to the velocity PINN model. This was done to determine whether knowledge of the passive scalar distribution within the domain would feed back into the RANS equation system via the advection-diffusion equation to improve the velocity reconstruction. In this investigation, the velocity data grid is kept constant at $h_{\boldsymbol{u}} = 1$, while the scalar data ($Sc = 1.0$) is supplied on a separate grid that varies in density between $1 < h_c < 10$. These results help to satisfy the objectives of accurately reconstructing the passive scalar fields, and are shown in Fig. 7, where it is clear that the addition of the advection-diffusion equation system and uniformly distributed scalar data produces more accurate reconstructions of the velocity fields. This effect is most pronounced in the surface case, where the passive scalar is specified through a boundary condition, but minor improvements can also be seen for the upstream and wake cases. It is likely that the surface case benefits the most from this change since accurate information about the passive scalar is provided on the cylinder surface - the region of the domain in which the velocity error is highest. Within the flow reconstruction framework, information that improves the solution of one equation can implicitly influence the solutions of other equations in the system. It is probable that solving the advection-diffusion equation (Eq. 3a) near the cylinder surface resulted in more accurate reconstructions of $\overline{u}$ and $\overline{v}$ by constraining the range of values that these quantities can take, since $\overline{u}$ and $\overline{v}$ appear in Eq. 3a, but $\overline{c}$ does not appear in Eq. 4. It is also noted that the error of $\overline{u}$ reconstruction was affected more than $\overline{v}$ by this change. This result illustrates that knowledge of passive scalar boundary conditions can greatly improve reconstructions of local velocities.

2. Effect of Data Sparsity on Full-Domain Scalar Reconstruction

The remainder of the results will focus only on reconstruction of the scalar quantities $\overline{c}$ and g. Reconstructed fields of $\overline{c}$ and g for the 3 flow cases at $Sc = 1$ are displayed in Fig. 8. For the sake of clarity and conciseness, only the dense data grid reconstructions ($h = 10$) are shown as the sparser reconstructions showed similar trends to the velocity fields (Sec. V A 1). Unlike $\boldsymbol{f}$, we expect that the reconstructions of g will resemble the CFD data since the term has not been modeled or decomposed in any way. Figure 8 confirms that this is the case since the reconstructed g fields match closely with the CFD data presented in Fig. 2 for all 3 cases throughout the majority of the domain. This is a significant result, as it

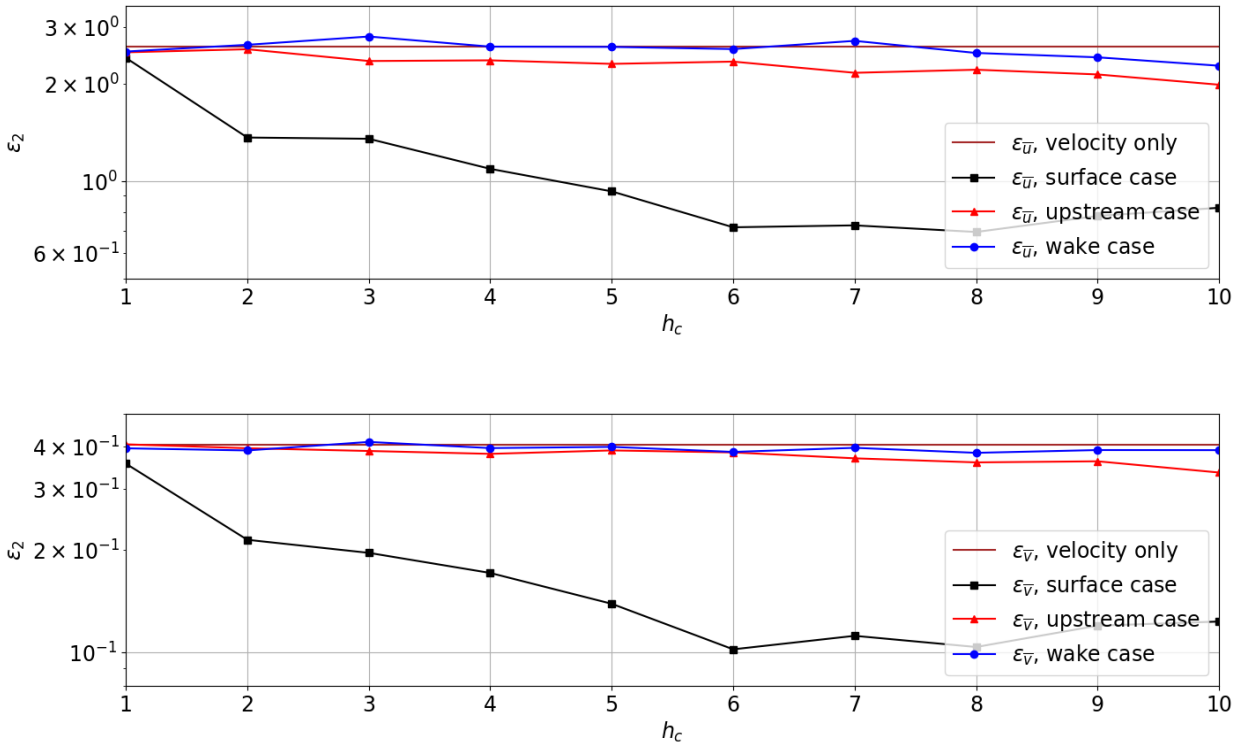


FIG. 7: L_2 norm versus h_c for velocity fields (top) $\overline{u}$ and (bottom) $\overline{v}$.

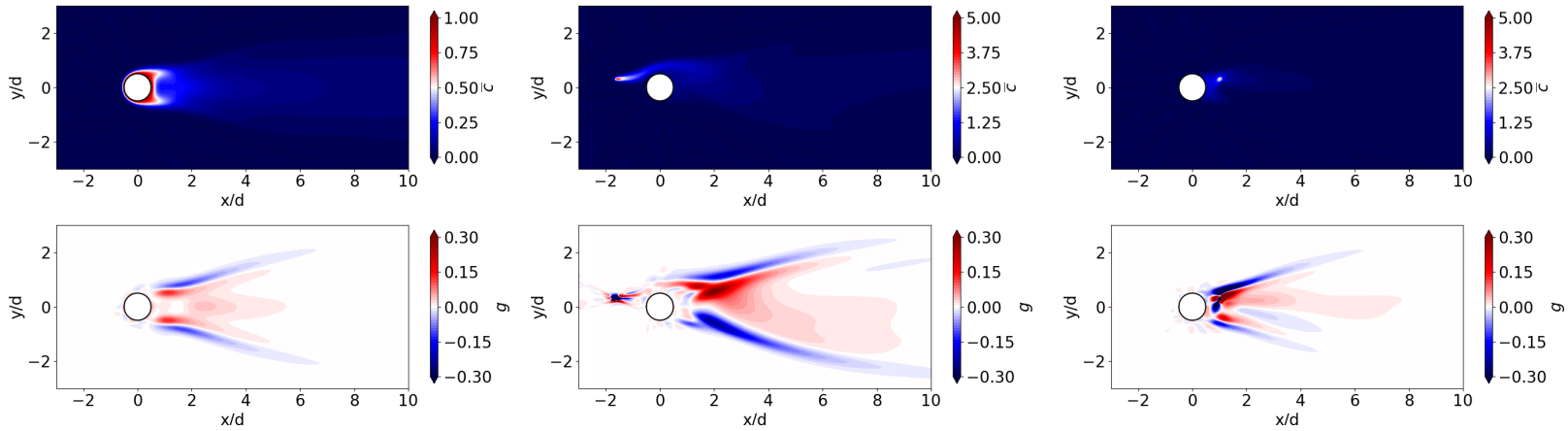
demonstrates that the PINN is inferring the true solution to the time-averaged advection-diffusion equation in the presence of limited data. The areas that exhibit the highest absolute error are found near the passive scalar sources, particularly in the upstream and wake cases in which the scalar source was intentionally not given as a boundary condition to study the PINN's response. In these cases and in the source region, the absolute error of g reaches magnitudes of > 10, suggesting that the PINN struggles with the high spatial gradients of $\overline{c}$ present in this part of the domain. As discussed in Sec. II B, g depends on the spatial gradients of the turbulent scalar fluxes $\overline{u'c'}$ and $\overline{v'c'}$, which are at a maximum at the point-source due to the step-function nature of the $\overline{c}$ field at this point. This causes the PINN to overestimate the magnitude of g at the source, and this behavior is only seen in the two point-source cases since in the surface case (where $\overline{c}$ is given as a boundary condition on the cylinder surface), the scalar gradients and fluxes are well-defined which results in an absolute error of near-zero. However, the regions of high absolute error in Fig. 8 seen downstream of the cylinder surface for the surface case is due to rapid dissipation of the passive scalar. The large decrease in magnitude of $\overline{c}$ challenges the PINN's optimization of the data loss in

this region, resulting in a less accurate reconstruction.

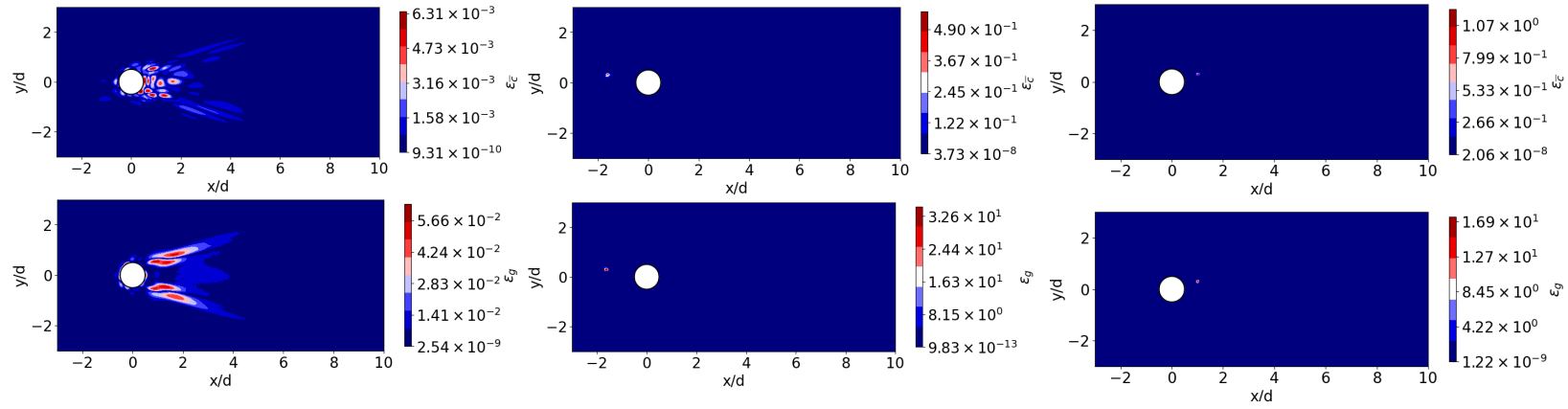
For the point-source cases, the overestimation behavior could be remedied through a relaxation function or constraint placed within the PINN. The reconstructed $\overline{c}$ fields show more predictable behavior, where the absolute error is highest around the scalar sources and far lower in the rest of the domain for similar gradient-dependent reasons as mentioned previously. Once again, the absolute error is the lowest in the surface case. The reader may notice asymmetry about the streamwise centerline in the absolute error fields of the surface case in Fig. 8, as well as figures in the previous section. This may be unexpected, since the RANS solution for this case should be entirely symmetric, yet the PINN reconstructions exhibit some asymmetries. This feature is a residual symptom of the PINN's random initialization (see Sec. IV B). The architecture of the PINN and its parameters do not have any direct relationship to the structure of the flow domain, meaning that the fully-connected network architecture has no mechanism by which to identify or respect spatial symmetries in the flow. Any learned symmetrical structures are purely a result of the PDE system and training data. Enforcing symmetry in the PINN solution is possible through applying Neumann BCs at the flow centerline, but we have chosen to avoid this approach in this investigation as is relies on a priori knowledge of the flow conditions - information that would be unavailable in practical applications.

The L_2 norms for all 3 cases are presented in Fig. 9, which also illustrates the differences between supplying $\overline{c}$ as a boundary condition versus only as data, as well as the differences between the ability of the PINN to reconstruct $\overline{c}$ and g. Figure 9 shows that the surface case is the most responsive to changes in h, where the reconstructions of $\overline{c}$ and g reach plateaus at $h \approx 5$ to 6. This is due to the additional reinforcement of the emissive boundary conditions on the cylinder surface. The cylinder surface is a region of importance (as discussed in Sec. V A 1) due to the high spatial gradients present, which is made even more critical in the surface case due to passive scalar emission from this boundary. These factors increase the sensitivity of the PINN to additional flow information and result in rapid improvement in reconstruction accuracy.

In contrast, the upstream and wake cases are slower to benefit from additional data. Nevertheless, the L_2 errors of the $\overline{c}$ fields are seen to improve by a factor of 10 between $1 < h < 10$, and show no signs of plateauing or saturating unlike the surface case or the velocity fields analyzed previously. On the other hand, $\varepsilon_2(g)$ is seen to diverge as the data



(a) Top: reconstructed fields of $\overline{c}$. Bottom: reconstructed fields of g. Left: surface case. Middle: upstream case. Right: wake case.



(b) Top: absolute error fields of $\overline{c}$. Bottom: absolute error fields of g.

FIG. 8: Reconstructions of $\overline{c}$ and g at $Sc = 1$ and $h = 10$.

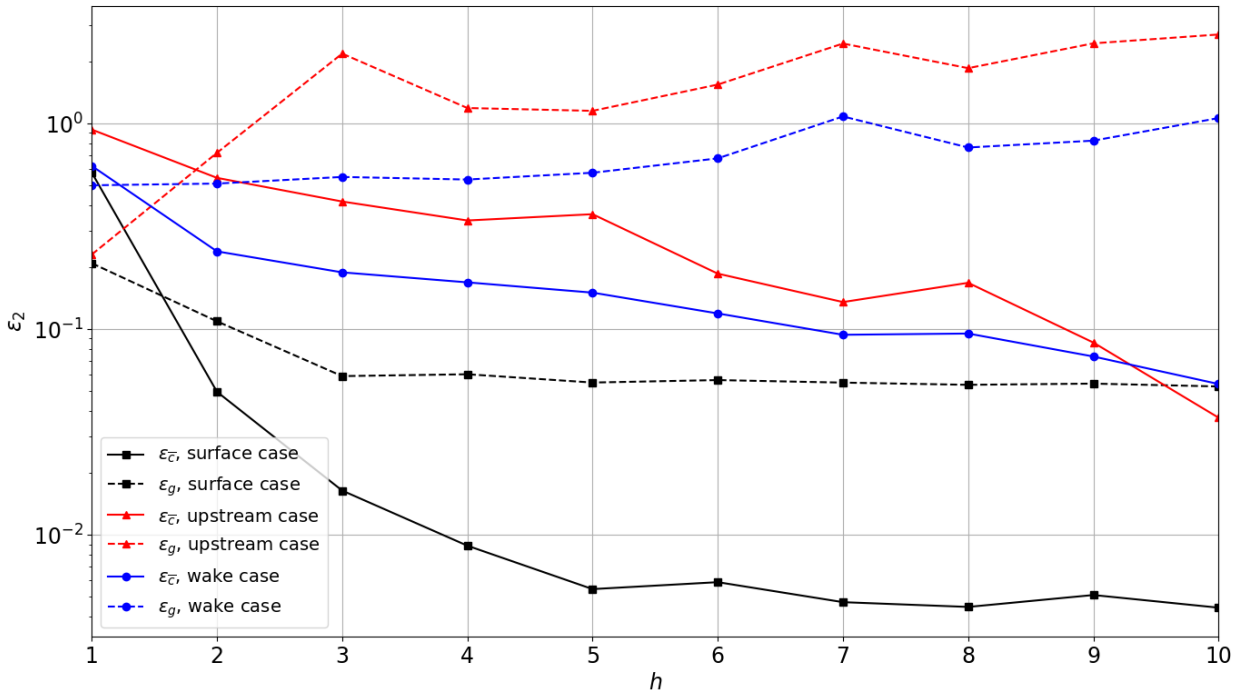


FIG. 9: L_2 norm versus h for scalar quantities $\overline{c}$ and g for all 3 flow cases at $Sc = 1$.

grid density increases in the upstream and wake cases. When observing Fig. 8b, it is clear again that the source region contributes almost entirely to this behavior. The reasons for this were previously discussed and it can be seen that this does not affect the reconstruction of $\overline{c}$ to the same degree, which responds positively to additional data. This may be due to differences in the PINN's reconstruction of directly accessible fields ($\overline{u}$, $\overline{v}$, $\overline{c}$) versus inferred fields (f_{su}, f_{sv}, g). Despite these limitations, the reconstructions of g remain remarkably accurate and well-structured away from the locations of the point-sources (see Fig. 2 to compare), demonstrating that the PINN can successfully reconstruct g on a macro scale even with relatively sparse data.

3. Effect of Schmidt Number on Full-Domain Scalar Reconstruction

In this investigation, the data point grid density was set as $h = 4$ and the Schmidt number of the flow was varied according to Table IV D. This data point grid density was chosen based on Fig. 9, ensuring the reconstructed fields would be sufficiently accurate while providing room for improvement without risk of overfitting to the different Sc datasets. Different Schmidt numbers were considered to determine how generalizable the PINN is to different

passive scalar regimes, and to identify any bias it may have to one regime versus another.

The L_2 norms for each Schmidt number are shown in Fig. 10, allowing comparison between the PINNs' reconstructive capabilities for the three flow cases at the five simulated Schmidt numbers. The reconstruction of $\overline{c}$ is seen to behave uniquely in all three cases. For the surface case, $\varepsilon_2(\overline{c})$ is lowest at $Sc = 0.1$ and reaches a maximum between $3 < Sc < 10$. For the upstream case, $\varepsilon_2(\overline{c})$ once again rises as Sc increases, but at a slower, steadier rate than for the surface case. For the wake case, $\varepsilon_2(\overline{c})$ remains near constant and appears to be reconstructed to the same accuracy independently of the Schmidt number.

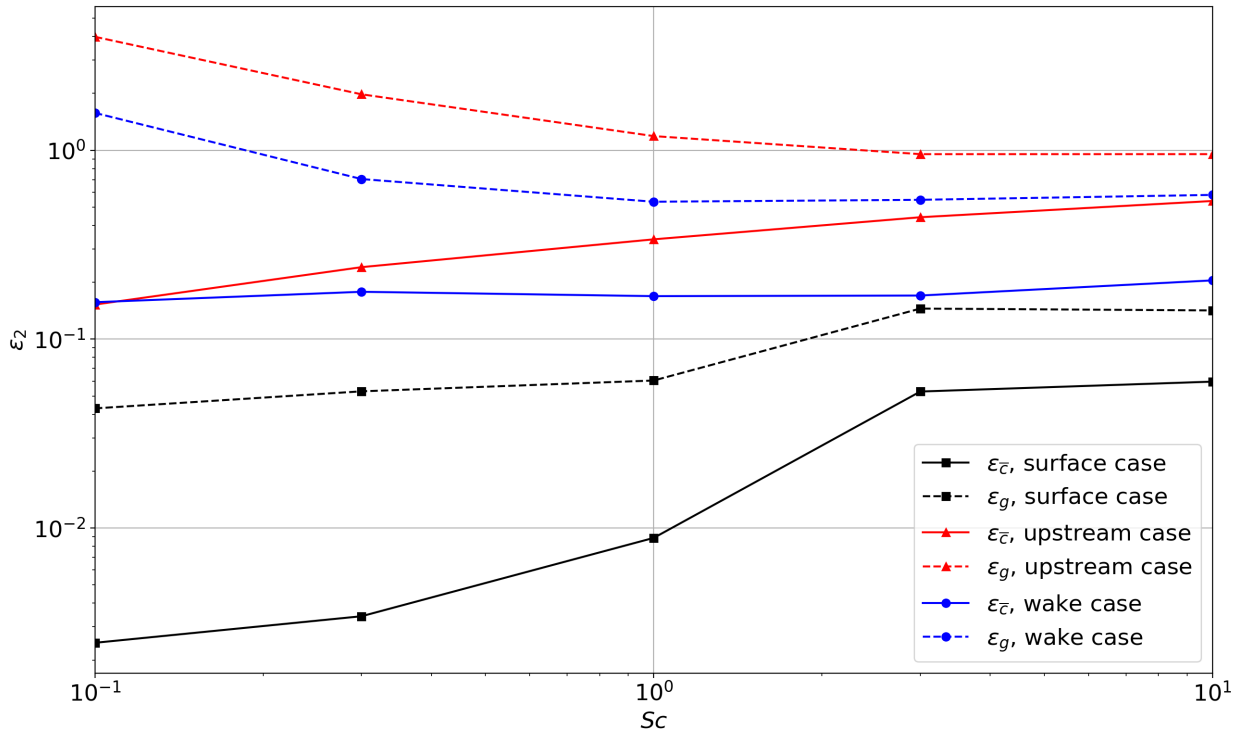


FIG. 10: L_2 norm versus Schmidt number for scalar quantities $\overline{c}$ and g for all 3 flow cases at $h = 4$. Solid lines: $\varepsilon_2(\overline{c})$. Dashed lines: $\varepsilon_2(g)$. Black: surface case. Red/triangle: upstream case. Blue/circle: wake case.

In the surface case, the increased diffusivity of the passive scalar at a low Sc complements the boundary conditions and helps the PINN compute smooth concentration gradients around the cylinder region. This advantage is lost as Sc increases, as the steeper gradients do not benefit as much from the boundary conditions and require more data to be resolved adequately since most of the diffusive activity of the scalar takes place over a very small region of the domain. A similar explanation can be applied to the upstream case, albeit

with a less severe loss of accuracy as this case never had any $\overline{c}$ boundary conditions to use as a basis for the PINN reconstruction.

For the wake case, it is possible that a different mechanism results in a Sc-independent error field. The point source for this case is located at (1.0, 0.3), which is near the center of the cylinder recirculation bubble. The oscillatory vortex shedding behind the cylinder, when averaged, results in values of $\overline{u}$ and $\overline{v}$ of near zero (see Fig. 1), which results in advective fluxes ($\overline{uc}$, $\overline{vc}$) which are also near zero in magnitude. This means the transport of the passive scalar is almost entirely dominated by diffusion (see Eq. 1a), resulting in a circular plume. We can infer from Fig. 10 that the PINN can reliably model purely diffusive behavior of $\overline{c}$, and that the error fields are dependent on the balance of advection to diffusion within the flow. The results suggest that the most challenging case for the PINN's reconstruction of $\overline{c}$ is an advection-dominated flow (high advective fluxes and a high Schmidt number corresponding to low diffusivity). This result is intuitive, as accurate modeling of the advective term of Eq. 1a is conditional on a well-resolved grid of both $\overline{c}$ as well as $\overline{u}$ and $\overline{v}$ data, meaning that error in the PINN's estimation of either field will be compounded when solving for the residual of Eq. 3a.

Interestingly, as Sc increases, $\varepsilon_2(g)$ is seen to decrease for the point-source cases, and increase for the surface case. The point-source region is responsible for the majority of the accumulated error. It is possible that a reduction in the effective source size as Sc increases results in a smaller region for $|g|$ to be incorrectly overestimated by the PINN, which reduces the total error of the reconstruction. The behavior of $\varepsilon_2(g)$ in the surface case is likely due to the steepening of scalar concentration gradients near the cylinder surface, as previously discussed with respect to $\varepsilon_2(\overline{c})$ in this case.

B. Source Identification

The cropped-domain investigations focus only on the upstream and wake cases in which $\overline{c}$ was not given to the PINN as a boundary condition, allowing the study of the PINNs' reconstructions when presented solely with sampled measurement data. Within these investigations, we instead use a more specific 'source metric' to quantify the PINN's accuracy when reconstructing the passive scalar source properties in order to satisfy the objectives. This metric is based on a *modified scalar concentration*, c^* proposed in Ref. [9] and applies

the following scaling to the scalar concentration field:

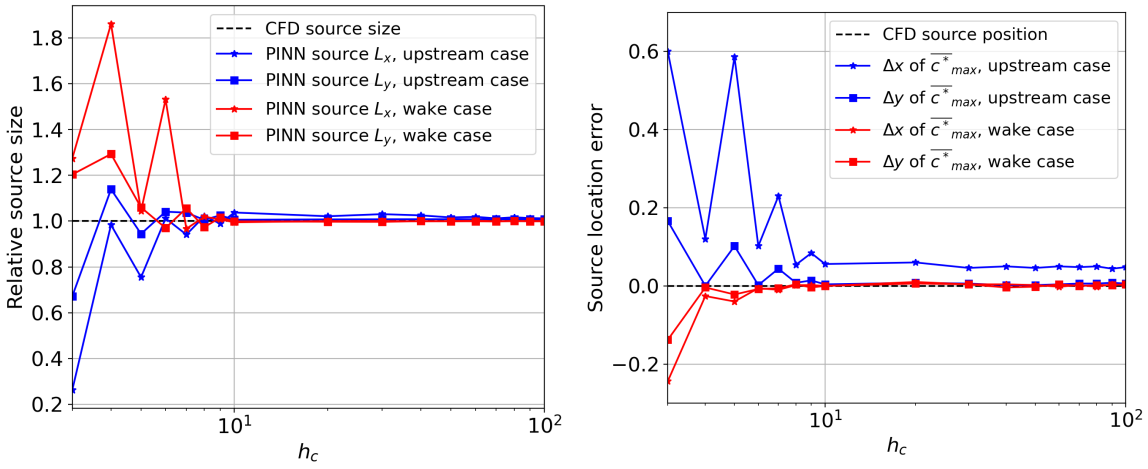
$$c^* = \begin{cases} log_{10}(10c) & c \geq 0.0001 \\ c & c < 0.0001 \end{cases}. \tag{9}$$

This scaling prevents the high concentrations at the source from overpowering the plots and helps to preserve the relations between small values of $\overline{c}$ while also eliminating the hassle of performing logarithms on values that approach machine precision. The quantity $\overline{c^*}$ is used to generate isocontours of the scalar concentration field localized around the source, which allows the dispersion of the source to be quantified as well as the location of highest concentration. An isocontour was produced at levels of $\overline{c^*} = 1.0$ and $\overline{c^*} = 1.2$ for the data sparsity and Schmidt number investigations, respectively. These levels were chosen to give the PINN a flexibility margin in the event that it underestimated the source strength of $\overline{c} = 5$. Similarly, a value of $\overline{c^*}$ too low would produce an isocontour that encompasses all of the source domain, and would therefore be impractical for assessing the PINN's source prediction capacity. The source properties are then compared to the reference fields. The dispersion is calculated by measuring the distance between the two furthest points on the isocontour in x and y to give the measures L_x and L_y which are normalized relative to the corresponding case's CFD source size. The location of highest concentration is simply taken as the location of the maximum value within the $\overline{c^*}$ array and is given as a distance from the normalized true source location as Δx, Δy.

1. Effect of Data Sparsity on Source Reconstruction

This section presents the results for the cropped-domain data sparsity investigation, in which the grid density of the $\overline{c}$ data points was varied between $3 < h_c < 100$, while the velocity data (supplied on a separate grid as described in Sec. IV D) remained at a constant density of $h_{\boldsymbol{u}} = 6$. The Schmidt number was set as $Sc = 1$ since it is within the range of typical experimental Schmidt numbers. The results illustrating how L_x, L_y, Δx and Δy vary as a function of the density of $\overline{c}$ data are shown in Fig. 11.

Figure 11a shows the convergence trend of L_x and L_y to the CFD reference values as h_c increases. The manner in which this occurs is different between the cases. For the upstream case, sparse $\overline{c}$ data is shown to result in an underestimate for both L_x and L_y,



(a) Predicted source dispersion L_x and L_y versus h_c.

(b) Location error of predicted maximum scalar concentration $\overline{c^*}_{max}$ versus h_c.

FIG. 11: Impact of h_c on reconstructed source properties for upstream and wake cases at $Sc = 1$.

while the inverse is true in the wake case which can also be seen for Δx and Δy in Fig. 11b. Furthermore, the upstream case (which is advection-dominant) exhibits a small steady-state error in L_x beyond a data point grid density of around 10 points per unit, in which the PINN overestimates the source size by a small margin. This trend is also visible in Fig. 11b, showing that the PINN predicts the source location to be around 0.05 units downstream of the true location. This error is only present for L_x in the advective case, which is the regime under which the corresponding advective flux $\overline{uc}$ is significantly higher than in other cases. Supplying additional data to the PINN does not improve the discrepancy, suggesting once again that high advective fluxes pose difficulties when solving Eq. 3a and accurately resolving regions of high advective transport. It is very possible that providing the PINN with a finer grid of the velocities (particularly $\overline{u}$) would remedy this. For all other cases in Fig. 11, the source properties converge to their respective CFD reference values before or around 10 points per unit. These results demonstrate that the PINN is capable of identifying the correct source properties from a density of data points similar to that which is needed to accurately reconstruct the entire flow field. Performance may be improved further by scaling the scalar concentration data to reveal smaller spatial variations in the concentration field, which could push the PINN toward identifying the correct source location under advective

regimes. Finally, Fig. 11 illustrates a 'zig-zagging' pattern of convergence for all cases, which is a consequence of the distribution of points within the data point grid. It is clear when considering Fig. 11 that the quality of the reconstructed field can vary significantly between neighboring models, and that there is an element of chance of where high-importance data points will be placed on the reconstructed field.

2. Effect of Schmidt Number on Source Reconstruction

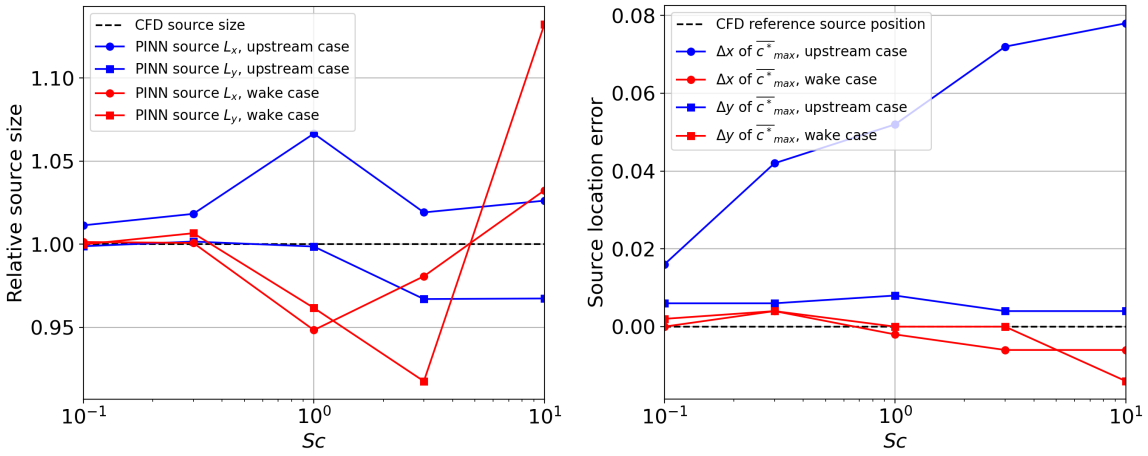
The final investigation focused on the influence of varying the Schmidt number between $0.1 < Sc < 10$ on the source reconstruction characteristics. The $\overline{c}$, $\overline{u}$ and $\overline{v}$ data grids had a coincident, uniform grid density of $h = 10$.

Figure 12 presents the relative reconstructed source sizes and locations for all Schmidt numbers. A noticeable difference between Fig. 12 and Fig. 11 is that the adjustment of the Schmidt number appears to be much less influential on source reconstruction than the density of the data point grid. The maximum discrepancy of the source size in Fig. 11 was around 90% (reducing until $h \approx 8$ to 9), while in Fig. 12 the error in the source size does not exceed around 14%, occurring at the extreme Schmidt number of $Sc = 10$. This suggests that data point placement is a hyperparameter that (when set optimally) can mitigate the uncertainty in performance when other properties of the physical system are adjusted.

Figure 12a shows that the PINN has difficulty reconstructing the source size at high Schmidt numbers, as seen by the more erratic behaviors of L_x and L_y beyond $Sc = 0.3$. Divergent behavior is seen in Fig. 12b for Δx of the upstream case, which is shown to increase linearly with Sc up to a maximum of $\Delta x = 0.08$, and is representative of the PINN's bias toward downstream prediction of source location in advection-dominated flows - a phenomenon also seen in Fig. 11b. All other cases in Fig. 12b do not stray beyond $\Delta = 0.01$, suggesting that the Schmidt number does not influence the prediction of the source location in diffusive regimes.

VI. CONCLUSIONS

This article has introduced passive scalar transport systems as another application of PINNs. This was done by investigating the mean fields of three classic cylinder flow cases



(a) Predicted source dispersion L_x and L_y versus Sc.

(b) Location error of predicted maximum scalar concentration $\overline{c^*}_{max}$ versus Sc.

FIG. 12: Impact of Schmidt number on reconstructed source properties for upstream and wake cases at $h = 10$.

at a Reynolds number of 150 and Schmidt numbers of $0.1 < Sc < 10$, with a passive scalar source active in a different part of the domain and under different boundary conditions for each case. The focus was on reconstructing the terms of the time-averaged advection-diffusion equation, the mean velocities, and identifying the source characteristics. This was done to assess the effectiveness of PINNs for solving the inverse problem of modeling passive scalar dispersion using sparse field measurements.

A PINN model trained using sparse velocity measurements was provided with passive scalar data on an increasingly dense uniform grid, where it was found that the additional scalar data resulted in improvements to the velocity reconstructions. These improvements were amplified when the PINN was given specified boundary conditions for the scalar. This trend was present throughout the first half of the results, where the implementation of Dirichlet scalar boundary conditions was found to produce very accurate reconstructed fields that performed better than the two point-source flows in all cases. This strongly suggests that any knowledge of scalar boundary conditions is greatly beneficial to PINN accuracy, and this information should be sought after in real-world applications to make the most effective use of PINNs. This result has implications for future applications of PINNs, particularly when training data for certain fields have differing degrees of availability.

The PINN was found to be able to reliably infer and reconstruct the unknown closure term g of the advection-diffusion equation, producing reconstructed fields that are remarkably accurate across the domain as a whole. However, g was found to be very sensitive in regions of high turbulent fluxes, requiring scalar boundary conditions to be reconstructed accurately in these areas. Changing the Schmidt number of the flow demonstrated that the PINN can be used at a wide range of Schmidt numbers, but generally performs better at low-intermediate Schmidt numbers ($0.3 < Sc < 3$) where diffusion dominates. The reconstruction of g was found to improve for the two point-source cases as Sc increases, but reduce for the surface case. The L_2 errors for the surface case remained lower than the point source cases at all Schmidt numbers. The PINN's ability to infer g from mean-flow measurements of scalar concentration and velocity, while simultaneously reconstructing and refining these fields, suggests advantages over simpler interpolation-based methods for modeling this type of problem.

When focusing on reconstruction of the scalar source properties (source size/location), the PINN reached peak accuracy at $h_c \approx 10$ and showed little improvement beyond this point. The source size and location predictions were found to exhibit a small steady-state error for properties parallel to the freestream flow direction, representing uncertainty in the sources' sizes and locations along the streamlines. The oscillatory nature of the reconstructed properties highlighted the importance of considering data point placement in high-sensitivity areas of the domain to ensure that gradients and fluxes can be resolved adequately. Nevertheless, the PINN was broadly successful in identifying the scalar source properties when provided with an amount of data similar to that which is necessary to reconstruct the entire flow field. This result is useful for applications regarding urban pollution control and modeling, and should be pursued further.

The use of PINNs should also be extended to more passive scalar transport cases, including complex geometries, 3-dimensional cases, as well as the use of experimental data [5, 6] instead of idealized CFD simulations. Studies could be performed using transient data instead of mean-flow fields, as it is possible that a time-resolved PINN solver is more likely to produce a unique solution than one trained on mean fields [21]. Other neural network architectures could be employed to solve this problem, such as variational autoencoders for model order reduction, or transformers to leverage long/short-term network attention [43].

Appendix: Network Hyperparameter Sensitivity Analysis

Due to the increased complexity of the physical systems in this study compared to previous work [19, 24, 42], a sensitivity study was conducted to determine the optimal network size for the physical system being investigated. As described in Sec. IV B, the hidden network size used in this article was 7 layers of width 100 ($N_L = 7$, $N_N = 100$). In this sensitivity study, additional PINNs were trained with varying increases in hidden layers and widths as presented in Table II.

Network size	**Network 1**	**Network 2**	**Network 3**	**Network 4**	**Network 5**
$\boldsymbol{N_L}$	7	8	9	8	9
$\boldsymbol{N_N}$	100	100	100	120	120

TABLE II: Network architectures for sensitivity investigation.

The PINN models reconstructed the $\overline{u}$, $\overline{v}$, $\overline{c}$ and g fields for the two point source flow cases (upstream and wake) at $h = 4$ and $Sc = 1.0$. The L_2 errors for the reconstructions were computed as shown in Fig. 13. It can be seen that increasing the network size through both an increase in N_L and N_N provides different improvements to the reconstructed fields. The $\overline{c}$ and g fields showed improvements of $< 3\%$ as the network size was increased. For $\overline{u}$ and $\overline{v}$ on the other hand, more tangible improvements can be found of at least 20% for both cases with a maximum of $\approx 46\%$ for $\overline{v}$ in the wake case between the smallest and largest networks. While this may appear significant, it is worth noting that these improvements are calculated relative to the original (smallest) model, where the L_2 errors were already of the order $O(10^{-3})$ prior to increasing the network size. Whether improvements of this magnitude are detectable and useful in practice remains to be seen, since the PINN's accuracy in this application is currently limited by its passive scalar modeling abilities. Nevertheless, this outcome may indicate that a larger PINN network architecture would be more suitable in use cases where training data is sparse in order to make the most effective use of available data.

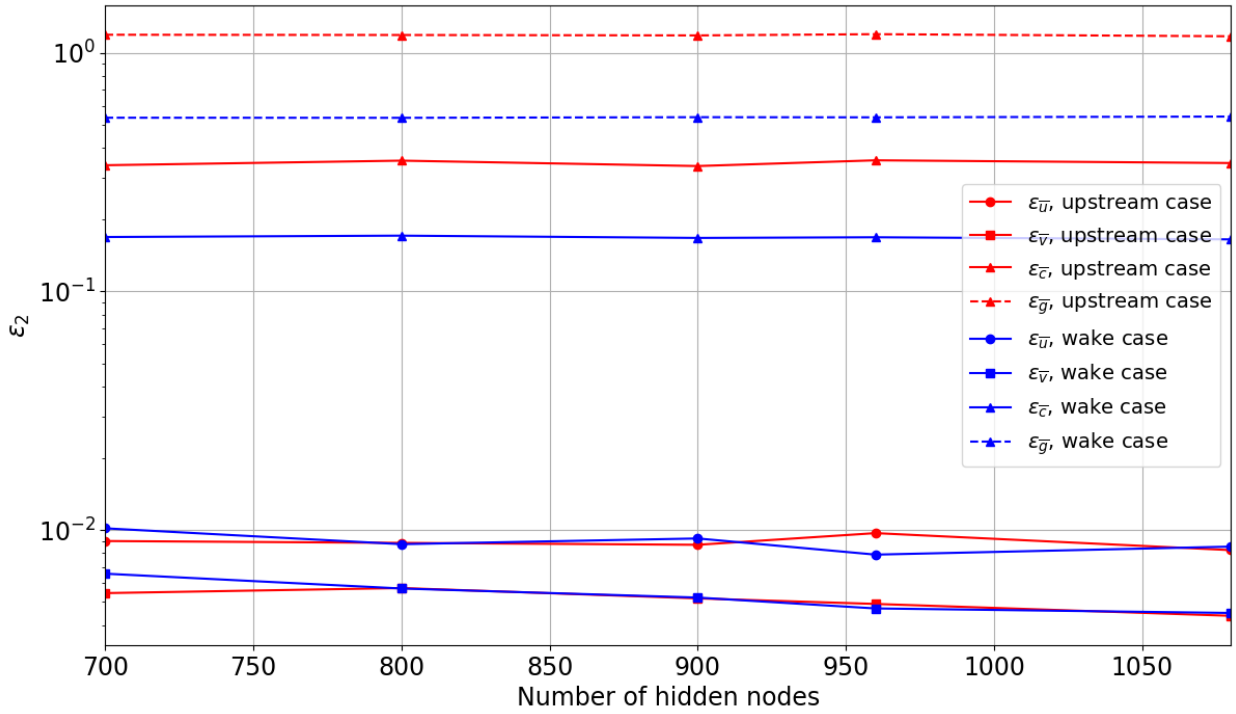


FIG. 13: Network size investigation.

ACKNOWLEDGMENTS

The authors would like to acknowledge the use of the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton, in the completion of this work.

J. I. Rawden gratefully acknowledges the financial support of the Antony Wright Scholarship, provided by the Department of Aeronautics and Astronautics at the University of Southampton, and the School of Engineering.

DATA AVAILABILITY

Data supporting the findings of this article are openly available at [44].

[1] WHO, Air pollution, `https://www.who.int/health-topics/air-pollution#tab=tab_1`, [Accessed 17-03-2025].

[2] V. Fuka, Z.-T. Xie, I. P. Castro, P. Hayden, M. Carpentieri, and A. G. Robins, Scalar fluxes near a tall building in an aligned array of rectangular buildings, Bound.-Layer Meteorol. **167**, 53 (2018).

[3] C. Gorlé, J. van Beeck, and P. Rambaud, Dispersion in the wake of a rectangular building: validation of two reynolds-averaged navier–stokes modelling approaches, Boundary-layer meteorology **137**, 115 (2010).

[4] I. P. Castro, I. R. Cowan, and A. G. Robins, Simulations of flow and dispersion around buildings, Journal of Aerospace Engineering **12**, 145 (1999).

[5] H. Lim, D. Hertwig, T. Grylls, H. Gough, M. v. Reeuwijk, S. Grimmond, and C. Vanderwel, Pollutant dispersion by tall buildings: laboratory experiments and large-eddy simulation, Exp. Fluids **63**, 92 (2022).

[6] T. Rich and C. Vanderwel, Pollutant dispersion around a single tall building, Bound.-Layer Meteorol. **190**, 34 (2024).

[7] P. Salizzoni, S. Fellini, H. Gamel, M. Marro, and L. Soulhac, Atmospheric dispersion downstream a two-dimensional obstacle: experimental evaluation of turbulence closure models, Boundary-Layer Meteorology **191**, 15 (2025).

[8] J. S. Apte and C. Manchanda, High-resolution urban air pollution mapping, Science **385**, 380 (2024).

[9] V. Mons, L. Margheri, J.-C. Chassaing, and P. Sagaut, Data assimilation-based reconstruction of urban pollutant release characteristics, J. Wind Eng. Ind. Aerodyn. **169**, 232 (2017).

[10] D. P. Foures, N. Dovetta, D. Sipp, and P. J. Schmid, A data-assimilation method for Reynolds-averaged Navier–Stokes-driven mean flow reconstruction, J. Fluid Mech. **759**, 404 (2014).

[11] S. Symon, N. Dovetta, B. J. McKeon, D. Sipp, and P. J. Schmid, Data assimilation of mean velocity from 2D PIV measurements of flow over an idealized airfoil, Exp. Fluids **58**, 1 (2017).

[12] D. Walters, A. J. Baran, I. Boutle, M. Brooks, P. Earnshaw, J. Edwards, K. Furtado, P. Hill, A. Lock, J. Manners, *et al.*, The met office unified model global atmosphere 7.0/7.1 and jules global land 7.0 configurations, Geoscientific Model Development **12**, 1909 (2019).

[13] A. Fraser, J. Abbott, and R. Rose, Application of data assimilation to the uk air quality forecast, Air Pollution Modeling and its Application XXIII , 617 (2014).

[14] U. Cadambi Padmanaban, B. Ganapathisubramani, C. Vanderwel, and S. Symon, Towards passive scalar reconstruction using data assimilation, 14th UK Conference on Wind Engineer-

ing (2024).

[15] P. Zille, T. Corpetti, L. Shao, and C. Xu, Super-resolution of turbulent passive scalar images using data assimilation, Exp. Fluids **57**, 21 (2016).

[16] S. Li, W. Zhou, H. J. Sung, and Y. Liu, On the origin of counter-gradient transport in turbulent scalar flux: physics interpretation and adjoint data assimilation, J. Fluid Mech. **999**, A81 (2024).

[17] R. Wang, B. Chen, S. Qiu, Z. Zhu, and X. Qiu, Data assimilation in air contaminant dispersion using a particle filter and expectation-maximization algorithm, Atmosphere **8**, 170 (2017).

[18] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. **378**, 686 (2019).

[19] L. Sliwinski and G. Rigas, Mean flow reconstruction of unsteady flows using physics-informed neural networks, Data-Centric Eng. **4**, e4 (2023).

[20] Y. Patel, V. Mons, O. Marquet, and G. Rigas, Turbulence model augmented physics-informed neural networks for mean-flow reconstruction, Phys. Rev. Fluids **9**, 034605 (2024).

[21] M.-C. Volk, A. Sergent, D. Lucor, M. Mommert, C. Bauer, and C. Wagner, A PINN methodology for temperature field reconstruction in the PIV measurement plane: Case of Rayleigh-Bènard convection, arXiv preprint arXiv:2503.23801 (2025).

[22] Y. Wang, B. Zhang, C. Hu, H. Jia, C. Wei, and H. Kikumoto, Flow field reconstruction and wind pressure estimation from sparse measurements using physics-informed neural networks: Application to two-dimensional street canyon flow, Building and Environment , 113616 (2025).

[23] C. Hu, Y. Cui, W. Zhang, F. Qian, H. Wang, Q. Wang, and C. Zhao, Solution of conservative-form transport equations with physics-informed neural network, Int. J. Heat Mass Trans. **216**, 124546 (2023).

[24] H. Eivazi, M. Tahani, P. Schlatter, and R. Vinuesa, Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations, Phys. Fluids **34** (2022).

[25] X. Jin, S. Cai, H. Li, and G. E. Karniadakis, NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations, J. Comput. Phys. **426**, 109951 (2021).

[26] M. Raissi, A. Yazdani, and G. E. Karniadakis, Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations, Science **367**, 1026 (2020).

[27] R. Vinuesa, S. L. Brunton, and B. J. McKeon, The transformative potential of machine learning for experiments in fluid mechanics, Nat. Rev. Phys. **5**, 536 (2023).

[28] J. G. von Saldern, J. M. Reumschüssel, T. L. Kaiser, M. Sieber, and K. Oberleithner, Mean flow data assimilation based on physics-informed neural networks, Phys. Fluids **34** (2022).

[29] A. Villié, S. Schmitter, J. G. von Saldern, S. Demange, and K. Oberleithner, Physics-informed neural networks for enhancing medical flow magnetic resonance imaging: Artifact correction and mean pressure and Reynolds stresses assimilation, Phys. Fluids **37** (2025).

[30] H. Wang, Y. Liu, and S. Wang, Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network, Phys. Fluids **34** (2022).

[31] A. Baklanov and Y. Zhang, Advances in air quality modeling and forecasting, Global Transitions **2**, 261 (2020).

[32] I. P. Castro and C. Vanderwel, *Turbulent flows: an introduction* (IOP Publishing, 2021).

[33] C. Keumnam, T. F. Irvine Jr, and J. Karni, Measurement of the diffusion coefficient of naphthalene into air, International journal of heat and mass transfer **35**, 957 (1992).

[34] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, DeepXDE: A deep learning library for solving differential equations, SIAM Rev. **63**, 208 (2021).

[35] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. **12**, 620 (1998).

[36] H. Jasak, OpenFOAM: Open source CFD in research and industry, Int. J. Nav. Archit. Ocean Eng. **1**, 89 (2009).

[37] OpenFOAM, Openfoam user guide, `https://www.openfoam.com/documentation/user-guide/a-reference/a.4-standard-boundary-conditions`, [Accessed 11/08/2025].

[38] C. Williamson, Three-dimensional wake transition, J. Fluid Mech. **328**, 345 (1996).

[39] D. P. Kingma and J. L. Ba, Adam: A method for stochastic gradient descent, in *Int. Conf. Learn. Represent.* (ICLR US., 2015) pp. 1–15.

[40] D. C. Liu and J. Nocedal, On the limited memory BFGS method for large scale optimization, Math. Program. **45**, 503 (1989).

[41] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, A limited memory algorithm for bound constrained optimization, SIAM J. Sci. Comput. **16**, 1190 (1995).

[42] G. Hasanuzzaman, H. Eivazi, S. Merbold, C. Egbers, and R. Vinuesa, Enhancement of PIV measurements via physics-informed neural networks, Meas. Sci. Technol. **34**, 044002 (2023).

[43] A. Solera-Rico, C. Sanmiguel Vila, M. Gómez-López, Y. Wang, A. Almashjary, S. T. Dawson, and R. Vinuesa, β-variational autoencoders and transformers for reduced-order modelling of fluid flows, Nat. Commun. **15**, 1361 (2024).

[44] `https://doi.org/10.5258/SOTON/D3810`.