Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Sensor placement study for estimation of Reynolds number in flow past a circular cylinder using an Inverse Physics-Informed Neural Network

Dinesh Sai Palli

palli_b231153me@nitc.ac.in

National Institute of Technology Calicut

Vardhan Nagulla

National Institute of Technology Calicut

Sai Saketha Chandra Athkuri

saisakethachandra@nitc.ac.in

National Institute of Technology Calicut

Research Article

Keywords: Inverse Physics-Informed Neural Networks, Sensor Placement, Reynolds Number Estimation, Flow Past a Circular Cylinder, Physics-Informed Neural Networks, Computational Fluid Dynamics

Posted Date: November 3rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11590329/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Sensor placement study for estimation of Reynolds number in flow past a circular cylinder using an Inverse Physics-Informed Neural Network

Dinesh Sai Palli*, Vardhan Nagulla†, and Sai Saketha Chandra Athkuri‡
National Institute of Technology Calicut, Kozhikode, Kerala, 673601, India

Determining an unknown flow parameter from a limited number of experimental measurement data is an essential requirement for many engineering problems. Inverse physics-informed neural network (I-PINN) provide a mesh-free way unlike the usual adjoint- or conjugate-gradient-based methods by adding the unknown parameter as a trainable one along with other weights of the neural network during the training phase. In this paper, we study the effective sensor placement for a flow past a circular cylinder to recover the Reynolds number (Re) of the flow. We consider five zones in the flow independently and try to recover the Re from synthetic data obtained from the validated forward PINN. In the first of three sweeps, the near-, mid-, and far-wake zones recover Re to within 9.9–21.6% error whereas the shear-layer zone stays in a tighter 28.2–31.7% error band. The upstream zone which samples the near uniform free stream values comes out to be the ineffective almost every time at 46.3–54.1%. The two repeat sweeps shift these errors, but the wake zones stay the most accurate in all three. We also see that more sensors do not translate into better accuracy in any zone, which implies that sensor location matters more than sensor count for recovering Re.

Nomenclature

D	=	cylinder diameter
F	=	viscous fraction, size of the viscous term relative to the convective term
G	=	gradient energy, $\|\nabla u\|^2 + \|\nabla v\|^2 + \|\nabla p\|^2$
Re	=	Reynolds number, $U_\infty D/\nu$
u, v	=	streamwise and transverse velocity components
p	=	pressure
R	=	cylinder radius
r	=	distance from the cylinder center
w_d, w_p, w_b	=	weights of the data, PDE and boundary-condition losses
x, y	=	streamwise and transverse coordinates
θ	=	angular position on cylinder surface, measured from the front stagnation point
C_p	=	surface pressure coefficient
U_∞	=	free-stream velocity
ν	=	kinematic viscosity, $1/Re$ (non-dimensional)
ρ	=	Spearman rank correlation coefficient
$\mathcal{N}_\phi$	=	neural network with trainable weights ϕ
$\mathcal{L}$	=	total training loss
N	=	number of sensor points supplied per zone
Re_{rec}	=	recovered Reynolds number

*Undergraduate Student, Department of Mechanical Engineering, National Institute of Technology Calicut.
†Undergraduate Student, Department of Mechanical Engineering, National Institute of Technology Calicut.
‡Assistant Professor, Department of Mechanical Engineering, National Institute of Technology Calicut.

I. Introduction

Designing engineering systems often requires engineers to work backward instead of predicting a flow from known conditions, the goal is to recover an unknown flow parameter from a limited set of measurements. The Reynolds number is a particularly important quantity to recover, because it controls the qualitative character of the flow itself [1]. For instance, the Reynolds number of the ocean current flowing past an offshore riser is rarely known, so it must be estimated from a few sensors placed in the flow around the riser, since it governs the drag and vortex shedding that drive fatigue-inducing vibrations. Estimating this parameter accurately from a small number of sensors is therefore useful both for validating lab experiment and numerical experiment for building systems that replicate the real ones.

Inverse problems of this kind are usually solved by embedding a computational fluid dynamics (CFD) solver in an optimization loop, in which the unknown parameter is adjusted iteratively until the computed solution matches the measurements. Each update requires the gradient of the loss function with respect to the parameter, which is generally obtained by solving an adjoint problem [2, 3]. The adjoint method yields accurate gradients, but at a price. It requires a fully differentiable solver and a separate adjoint derivation for every set of governing equations [3], and each optimization step involves one forward solve and one adjoint solve [3]. Over many inversions, this cost becomes substantial.

The physics-informed neural network (PINN) of Raissi et al. [4] offers a more direct alternative. A PINN is trained so that its output satisfies the governing partial differential equations and boundary conditions, with the residuals of both added to its loss function. Because the network is fully differentiable, an unknown physical parameter can simply be added to the list of trainable quantities and updated alongside the network weights, using a small data-loss term computed from whatever sparse measurements in an inverse fashion. No additional adjoint solver or new numerical code is required for each choice of design variables. This has been demonstrated for many fluid problems [4–9]. Raissi et al. [5] used a closely related framework to infer entire velocity and pressure fields directly from passive-scalar flow visualizations. Related physics-constrained formulations have gone a step further and shown that even labeled simulation data can be avoided entirely during training [10]. A recent review by Cai et al. [11] summarizes the large and fast-growing body of work applying PINNs across fluid mechanics.

The same inverse-PINN idea has been extended well beyond fluid flow. Jagtap et al. [12] developed conservative PINNs for forward and inverse problems on discrete domains, and Ngo and Lim [13] used an inverse PINN to jointly recover internal states and unknown kinetic parameters in a catalytic reactor model. Together with the broader outlook on physics-informed machine learning given by Karniadakis et al. [14], these studies show that treating an unknown parameter as one more trainable network weight is now a fairly mature and general strategy across engineering disciplines. It is not tied to a single equation or field.

PINNs also have training difficulties that bear directly on this study. Chuang and Barba [15] trained a PINN on the unsteady Navier–Stokes equations for flow past a cylinder without any labelled data. The network did not produce vortex shedding and converged to a steady state solution instead. Krishnapriyan et al. [16] found that PINNs can fail even on moderately difficult problems. They attributed these to a poorly conditioned optimization landscape, not to limited expressiveness of the network. Wang et al. [17] similarly traced similar failures to an imbalance between the gradients of the individual loss terms. For this reason, an inverse PINN should first be tested on a flow that is known to train well. Otherwise, a poor estimate of the parameter could simply be a training failure and say nothing about how recoverable the parameter is. Once training is reliable, two questions remain: how much sensor data is needed, and where should it be collected?

Sensor location has also been studied outside the PINN literature, under the heading of sparse sensor placement [18, 19]. Manohar et al. [20] applied sparse sensor placement to flow reconstruction, using features learned from training data to select the sensor locations. They also point out that careful placement pays off most when sensors are expensive. In all of these studies, though, the goal is to reconstruct a whole field. We are not aware of any systematic test of whether location matters as much when an inverse PINN recovers a single global parameter such as the Reynolds number.

We address this gap with steady, two-dimensional, incompressible flow past a circular cylinder at $Re = 40$, a canonical case on which PINNs behave well. It makes a sensible first test. The wake is steady and symmetric with no vortex shedding [1, 21], which avoids the training problems discussed above, and classical experimental data exist against which a forward PINN can be validated [22]. The forward PINN is trained first and checked against these data. The validated flow field then supplies synthetic sensor readings for an inverse PINN. The inverse PINN is trained while sweeping both the number of sensors ($N = 10, 20, 30, 40, 50$) and their location across five physically distinct zones of the flow - the near wake, mid wake, far wake, upstream region, and shear layers - with the goal of turning the resulting trends into practical, physically grounded guidance for placing sensors in sparse data flow parameter estimation experiments.

II. Governing Equations

The non-dimensional, incompressible Navier-Stokes equations are solved to obtain the steady state solution for flow past circular cylinder. The Reynolds number considered here is 40.

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{1}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right), \tag{2}$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right). \tag{3}$$

The computational domain spans $[-3, 7] \times [-2, 2]$ with the cylinder of radius $R = 0.5$ as shown in Fig. 1. In the forward problem, Re is prescribed and Eqs. (1)–(3) are solved for the field (u, v, p). In the inverse problem, $\nu = 1/Re$ is instead treated as an unknown, trainable scalar recovered jointly with the field from sparse observations.

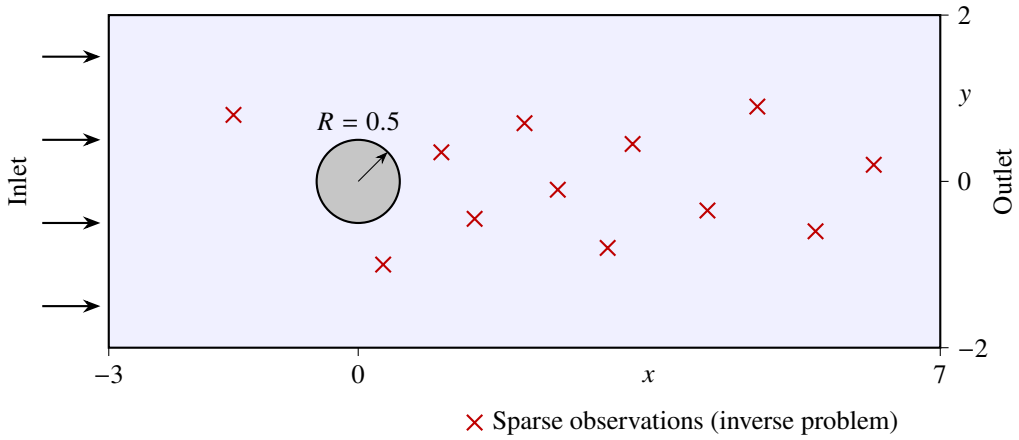


Fig. 1 Computational domain $[-3, 7] \times [-2, 2]$ with the cylinder of radius $R = 0.5$ excised.

III. Methodology

A. Forward PINN

The forward model is a fully connected network $\mathcal{N}_\phi : (x, y) \mapsto (u, v, p)$ with 8 hidden layers of 20 neurons each, tanh activations, and Glorot initialization (Fig. 3). Spatial derivatives needed for the PDE residuals come from automatic differentiation. We train the network on 10,000 interior collocation points and 300 boundary points sampled with Latin hypercube sampling, with extra refinement near the cylinder surface. Boundary conditions consist of a uniform inlet ($u = 1$, $v = 0$), Neumann outflow, free-stream conditions on the top and bottom boundaries, and no-slip on the cylinder surface. The loss function is

$$\mathcal{L} = w_{PDE}\mathcal{L}_{\text{PDE}} + w_{BC}\mathcal{L}_{\text{BC}}, \tag{4}$$

with $w_{PDE} = w_{BC} = 1$, which we minimize with Adam at a fixed learning rate of 10^{-3} for 20,000 iterations. The forward model was implemented in DeepXDE [23] with a PyTorch backend.

B. Forward PINN Validation

We check the trained forward solution against the classical experimental surface pressure-coefficient distribution of Apelt [22] for $Re = 40$. The PINN-predicted $C_p(\theta)$ tracks the experimental data closely over the front face and separation region, with only a small offset near the base ($\theta > 120°$) – typical of steady laminar closure (Fig. 4b). The predicted velocity and pressure fields also show the expected symmetric recirculation bubbles, a steady attached wake

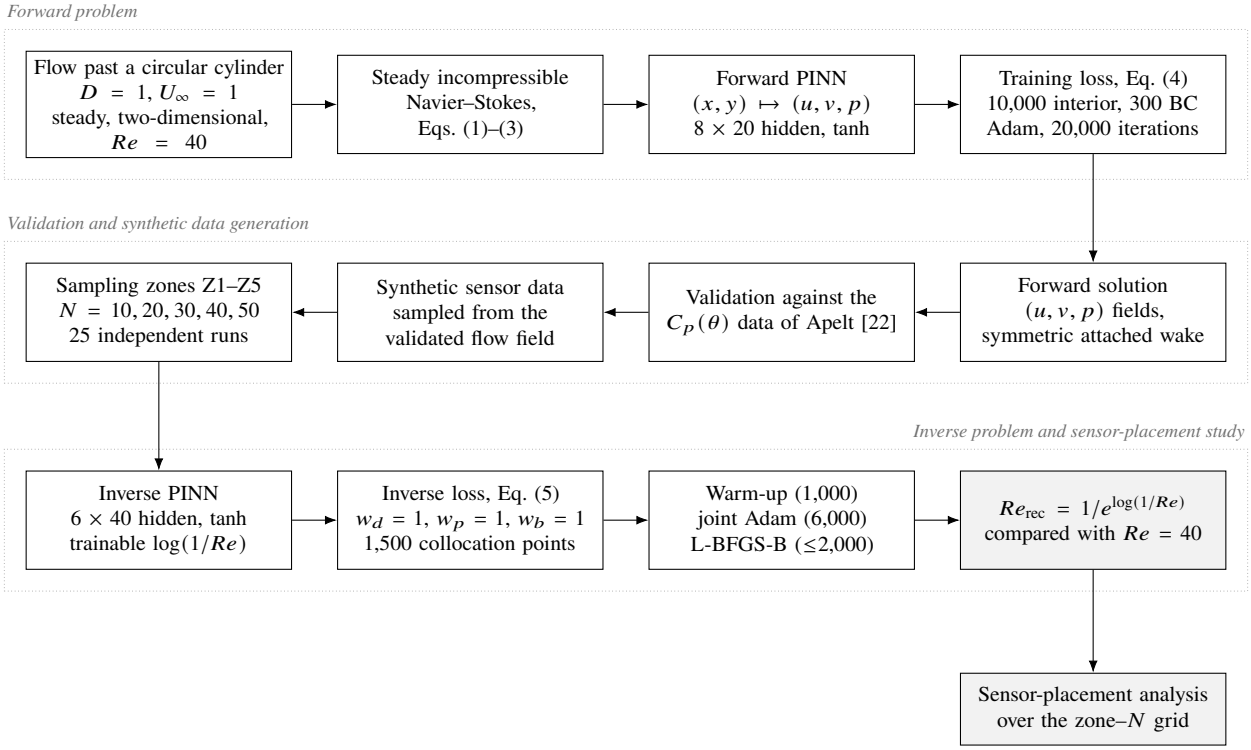


Fig. 2 Overall methodology. A forward PINN is trained and validated, the validated field is sampled to produce sparse synthetic sensor data in five spatial zones, and an inverse PINN recovers Re independently for each zone–N combination.

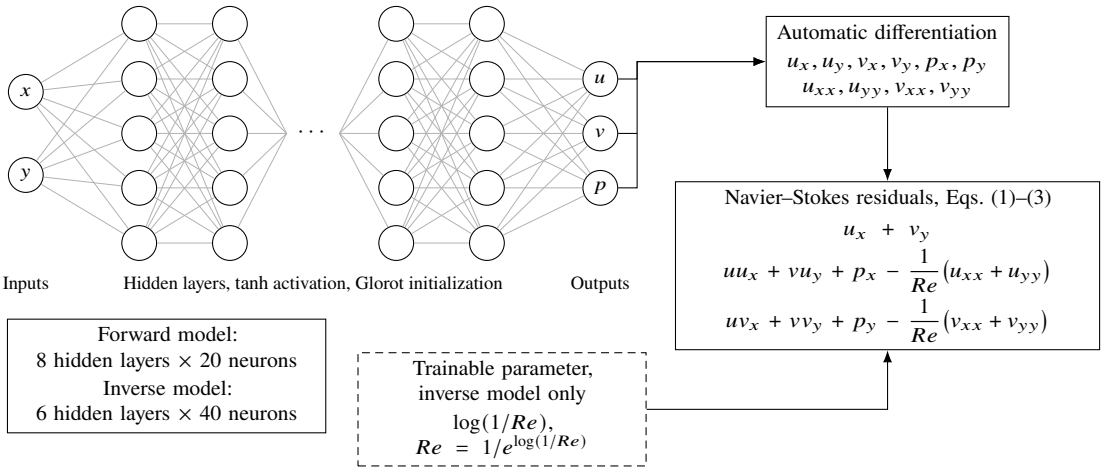
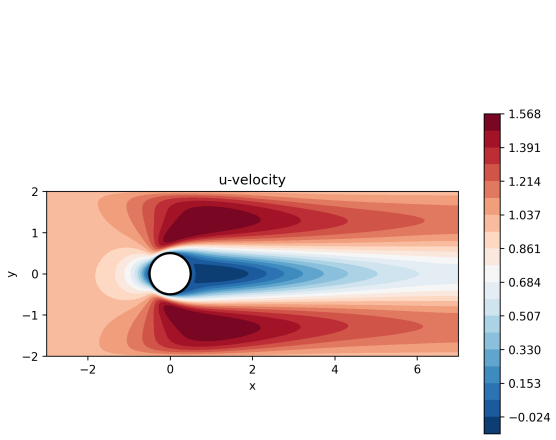
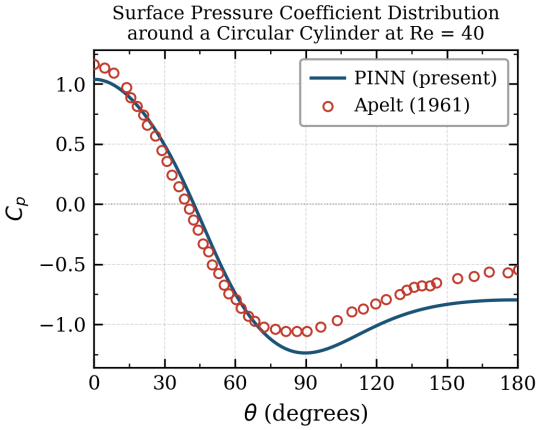


Fig. 3 Fully connected PINN architecture used for both the forward and the inverse model: inputs (x, y), $\tanh$-activated hidden layers, and outputs (u, v, p), with the derivatives entering the Navier–Stokes residuals obtained by automatic differentiation. The inverse model carries the additional trainable scalar $\log(1/Re)$, which enters the momentum residuals through the current estimate of Re.

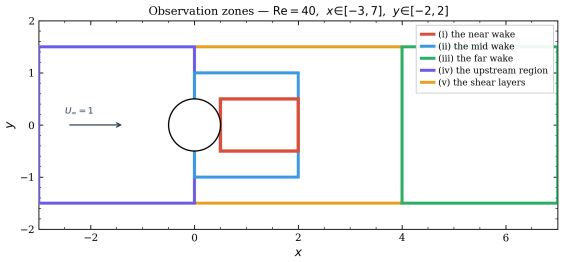
of physically consistent length, and a correctly located front stagnation point (Fig. 4a). Taken together, this agreement confirms that the forward PINN reproduces the correct physics and can be trusted as a source of synthetic data for the inverse study; Fig. 4c shows the five zones subsequently used to sample that data.



(a) u-velocity field, $Re = 40$



(b) C_p vs. Apelt [22]



(c) Sensor observation zones

Fig. 4 Forward PINN solution and validation for flow past a cylinder ($D = 1$, $Re = 40$): (a) predicted streamwise velocity field, (b) surface pressure coefficient compared against experimental data, and (c) the five spatial zones used for sparse sensor sampling in the inverse study.

C. Inverse PINN Formulation

The inverse PINN is also a fully connected tanh network, here with 6 hidden layers of 40 neurons each (the forward model has 8 layers of 20). The parameter $\log(1/Re)$ is added as an extra trainable variable, parameterized in log-space so gradient updates stay stable. The recovered value is kept within $5 \le Re \le 200$. A data-loss term is added to the total loss:

$$\mathcal{L} = w_d\,\mathcal{L}_{\text{data}} + w_p\,\mathcal{L}_{\text{PDE}} + w_b\,\mathcal{L}_{\text{BC}}, \qquad Re = 1/e^{\log(1/Re)}, \tag{5}$$

with weights $w_d = 1$, $w_p = 1$, $w_b = 1$. $\mathcal{L}_{\text{data}}$ is the mean-squared error between the network's predicted (u, v, p) and the sparse synthetic sensor values at N sampled locations within a given zone; $\mathcal{L}_{\text{PDE}}$ is evaluated on 1,500 collocation points using the current estimate of Re. The boundary loss combines four terms with different weights:

$$\mathcal{L}_{\text{BC}} = \mathcal{L}_{\text{in}} + \mathcal{L}_{\text{out}} + 2\,\mathcal{L}_{\text{wall}} + 5\,\mathcal{L}_{\text{cyl}}, \tag{6}$$

where $\mathcal{L}_{\text{in}}$ sets $u = 1$, $v = 0$ at 50 inlet points and $\mathcal{L}_{\text{out}}$ sets $p = 0$ at 50 outlet points. On the top and bottom boundaries (60 points each), $\mathcal{L}_{\text{wall}}$ enforces the free-stream values $u = 1$, $v = 0$. No-slip ($u = v = 0$) on the cylinder is enforced by $\mathcal{L}_{\text{cyl}}$ at 100 surface points.

Training runs in three phases. For the first 1,000 epochs (the warm-up), Re is fixed at the initial guess $Re_0 = 20$ and the network learns to fit the sensor data and the physics. We do this so that an unconverged field cannot drag the parameter estimate around early on. The joint-training phase follows and lasts for 6,000 epochs. Here the network weights and Re are trained together, each with its own Adam optimizer ($\text{lr}_\phi = 10^{-3}$ and $\text{lr}_{Re} = 10^{-4}$). Late in this phase

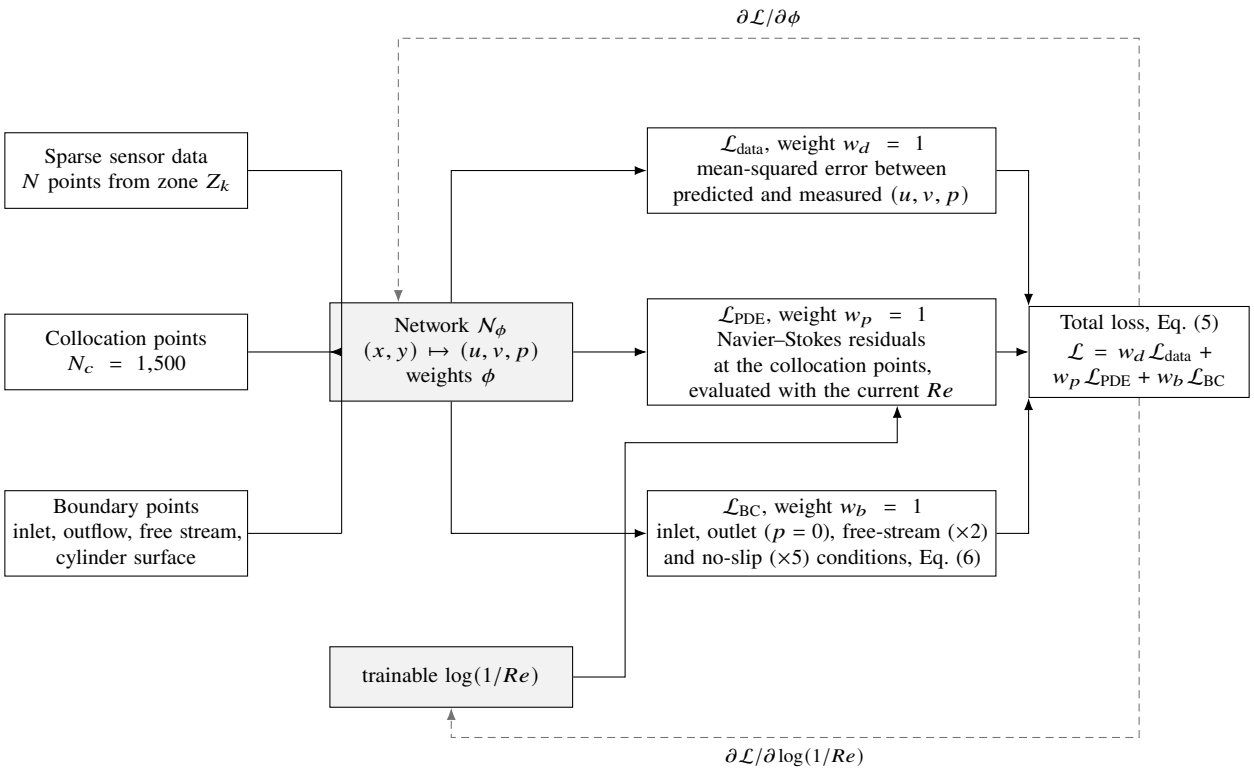


Fig. 5 Composition of the inverse-PINN loss, Eq. (5)**. The data term is the only component computed from the zone-specific sensor samples; the PDE and boundary terms are evaluated on the same domain-wide points in every run. Dashed lines denote the gradients that update the network weights and the trainable parameter** $\log(1/Re)$**.**

the learning rate is reduced by a factor of 10. Then we do L-BFGS-B fine-tuning, for at most 2,000 iterations. All the trainable parameters, the network weights and $\log(1/Re)$, are put into one flattened vector, and L-BFGS-B updates that vector using second-order line-search steps. Adam only uses first-order steps. This full three-phase schedule (warm-up → joint Adam → L-BFGS-B) is applied to every zone and sensor count reported in Sec. IV.

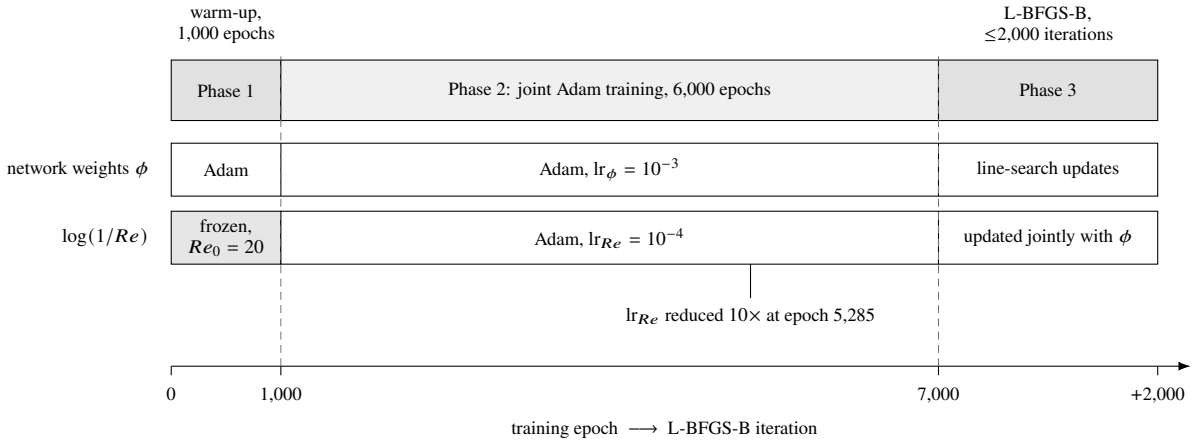


Fig. 6 Three-phase optimization schedule. All three phases minimize the same total loss, Eq. (5). Re is frozen at its initial guess during the warm-up phase, updated by a separate Adam optimizer during joint training, and finally updated together with the network weights as part of a single flattened parameter vector during L-BFGS-B fine-tuning.

D. Sensor Placement Study Design

We define five zones in the validated forward field : (Z1) the near wake, right behind the cylinder; (Z2) the mid wake; (Z3) the far wake; (Z4) the upstream region, in the nearly uniform approach flow; and (Z5) the shear layers around the recirculation bubbles. For each zone, the inverse PINN is trained separately with $N = 10, 20, 30, 40$ and 50 sensor points, all drawn from that zone alone. Zone Z1 required special treatment. As the smallest zone, it contains only 30 of the synthetic points, which is too few for $N = 40$ and 50. Its box was therefore enlarged in steps of 0.02 on every side until it enclosed 20 additional points, which required a margin of 0.16. In the sampling order, these additional points follow the original 30. Each run yields a recovered value Re_{rec}, and we report its percentage error relative to the true value $Re = 40$, against a target of less than 5%. Sensor location and sensor count are varied one at a time so that their effects can be separated.

IV. Results and Discussion

A. Effect of Sensor Location and Count

Table 1 lists the recovered Re and the corresponding percentage error for each zone and for $N = 10$ to 50. The near wake (Z1) gives the smallest error in the study, 9.9% at $N = 10$, and its error remains below 16% at all other values of N. The far wake (Z3) follows, with errors of 12.9–14.3%. The mid wake (Z2) reaches comparable accuracy, but its error varies more widely, from 13.2% to 21.6%. In the shear layers (Z5), the error stays within 28.2–31.7% for all N. The upstream zone (Z4) is the worst at every N, from 46.3% at $N = 10$ to 54.1% at $N = 20$. Adding sensors does not necessarily reduce the error. For the near wake, the error goes 9.9%, 13.3%, 11.6%, 14.3% and 15.4% as N goes from 10 to 50. No zone and N combination reaches the 5% error target. The closest is Z1 at $N = 10$ with 9.9%, about double the target.

We repeated the full sweep twice more, as Run 2 and Run 3 (Table 3). The errors changed noticeably between runs. The best error over all zones was 9.9% in Run 1, 14.2% in Run 2 and 10.8% in Run 3. So a single estimate depends on the network initialization and on which sensor points are sampled. In all three runs, the best errors of the three wake zones are lower than that of the shear layers. In Run 1 and Run 3 the upstream zone is the worst. Table 2 gives the full sweep for Run 3. Its best result is 10.8% at Z2 with $N = 10$, and the far wake stays between 10.9% and 13.8%, close to

the wake accuracy in Run 1. The upstream zone is again the worst in Run 3, with errors from 43.9% to 62.3%.

Table 1 Recovered Reynolds number and percentage error from the inverse PINN for Run 1 (full warm-up + joint Adam + L-BFGS-B pipeline) across all five sampling zones and the full sensor-count sweep (true $Re = 40$).

Zone	$N = 10$		$N = 20$		$N = 30$		$N = 40$		$N = 50$	
	Re_{rec}	Err.(%)	Re_{rec}	Err.(%)	Re_{rec}	Err.(%)	Re_{rec}	Err.(%)	Re_{rec}	Err.(%)
Z1 – Near wake	**36.04**	**9.9**	34.68	13.3	35.34	11.6	34.27	14.3	33.82	15.4
Z2 – Mid wake	34.64	13.4	31.41	21.5	34.70	13.2	33.67	15.8	31.38	21.6
Z3 – Far wake	34.67	13.3	34.28	14.3	34.86	12.9	34.84	12.9	34.77	13.1
Z4 – Upstream	21.49	46.3	18.38	54.1	19.03	52.4	20.92	47.7	21.10	47.3
Z5 – Shear layers	28.73	28.2	28.56	28.6	28.12	29.7	27.33	31.7	28.00	30.0

Table 2 Recovered Reynolds number and percentage error for Run 3 (independent replicate, $Re_0 = 20$) across all five sampling zones and the full sensor-count sweep (true $Re = 40$).

Zone	$N = 10$		$N = 20$		$N = 30$		$N = 40$		$N = 50$	
	Re_{rec}	Err.(%)	Re_{rec}	Err.(%)	Re_{rec}	Err.(%)	Re_{rec}	Err.(%)	Re_{rec}	Err.(%)
Z1 – Near wake	34.51	13.7	34.15	14.6	**34.76**	**13.1**	34.44	13.9	31.75	20.6
Z2 – Mid wake	**35.68**	**10.8**	34.54	13.7	31.71	20.7	31.95	20.1	33.81	15.5
Z3 – Far wake	34.48	13.8	34.96	12.6	35.06	12.4	**35.63**	**10.9**	35.45	11.4
Z4 – Upstream	19.61	51.0	**22.45**	**43.9**	15.09	62.3	15.18	62.1	19.03	52.4
Z5 – Shear layers	29.57	26.1	28.34	29.2	**29.99**	**25.0**	27.94	30.2	28.30	29.3

Table 3 Best recovered error per zone across three independent replicates: the original sweep (Run 1), a repeat sweep with different initialization (Run 2), and a third independent replicate (Run 3, Table 2). True $Re = 40$.

Zone	Run 1 (%)	Run 2 (%)	Run 3 (%)
Z1 – Near wake	9.9	15.7	13.1
Z2 – Mid wake	13.2	14.2	10.8
Z3 – Far wake	12.9	16.7	10.9
Z4 – Upstream	46.3	24.2	43.9
Z5 – Shear layers	28.2	25.6	25.0

B. Effect of the Initial Guess Re_0 on Recovery

All results reported above use a warm-up value of $Re_0 = 20$ roughly half the true $Re = 40$ and comfortably inside the $[5, 200]$ clip range placed on the recovered parameter. We tested the sensitivity to this choice by repeating the Z1 (near-wake) sweep with two other starting values and nothing else changed. The first one started from $Re_0 = 1$ which required lowering the clip bound to $[1, 200]$, since the starting value would otherwise have been clipped straight away. The second started from $Re_0 = 100$. Sensor locations, sensor counts, network architecture and the three-phase schedule were the same as in the $Re_0 = 20$ case (Run 3, Table 2). Table 4 gives the results.

Neither mis-initialized run works, and each fails in its own way. Starting from $Re_0 = 1$, the estimate barely moves: Re_{rec} remains between 1.25 and 1.27 for every sensor count, corresponding to an error of 96.8–96.9% that does not vary with N. The optimizer has stalled close to its starting point. At $Re = 1$, the flow lies deep in the Stokes-drag regime,

Table 4 Recovered Reynolds number and percentage error in Z1 (near wake) for three initial guesses Re_0, over all sensor counts. True $Re = 40$. The $Re_0 = 20$ column is the Z1 row of Table 2.

N	$Re_0 = 1$ Re_{rec} (Err.%)	$Re_0 = 20$ Re_{rec} (Err.%)	$Re_0 = 100$ Re_{rec} (Err.%)
10	1.27 (96.8)	34.51 (13.7)	139.59 (249.0)
20	1.26 (96.8)	34.15 (14.6)	102.91 (157.3)
30	1.26 (96.8)	34.76 (13.1)	91.57 (128.9)
40	1.25 (96.9)	34.44 (13.9)	154.10 (285.2)
50	1.25 (96.9)	31.75 (20.6)	140.86 (252.2)

where viscous terms dominate the momentum balance for any nearby value of Re. The network can then fit the sensor data, the PDE residuals and the boundary conditions reasonably well with little change in Re, and the residual barely responds to the parameter. This creates a shallow basin around the initial value from which the solution never escapes. Inside it $\partial\mathcal{L}/\partial Re$ is small, so neither the joint Adam phase nor the L-BFGS-B phase gets a strong enough signal to push the estimate out. The mechanism is the same weak gradient identified below (Sec. IV.C) as the cause of poor recovery in the upstream zone, but here it is caused by a poor *initial* guess and not by a poor *sensor* location.

Starting at $Re_0 = 100$ goes wrong in the other direction. Instead of settling near its starting point, the recovered Re swings unpredictably across the sweep, from 91.6 up to 154.1, with errors of 128.9–285.2%. All of these are worse than the $Re_0 = 1$ case, despite starting only 2.5× further from the truth (as opposed to 40×). Here the gradient signal is too strong and too poorly conditioned. Because Re is trained in log-space through $\log(1/Re)$, a fixed Adam step changes Re by a fixed fraction, so the absolute swings in Re grow with Re itself. From $Re_0 = 100$, each step moves the estimate about 2.5 times as far as the same step would near $Re = 40$, and the optimizer overshoots the true value and never settles near it. Unlike the $Re_0 = 1$ case, there is no stable attracting band at all: the estimates do not even agree with each other across N, let alone with $Re = 40$.

With $Re_0 = 20$, by contrast, every N gives a recovered value within a factor of 1.3 of the truth (13.1–20.6% error), an order of magnitude better than either mis-initialized case. The practical takeaway is that the joint optimizer's basin of attraction around the true Re is narrow relative to the full clipped search range: an initial guess needs to be within roughly the same order of magnitude as the (unknown) true value for the wake-zone accuracy levels in Table 2 to be reachable at all. In a genuine parameter-identification setting where Re is not known in advance, this argues for a coarse bracketing step – for example a short, cheap low-fidelity sweep, or a physically informed prior from the qualitative flow regime (attached vs. separated wake, presence of vortex shedding, etc.) – before committing to the full warm-up → joint Adam → L-BFGS-B pipeline reported elsewhere in this paper.

C. Physical Interpretation

The gap between wake or shear layer performance and upstream performance has a fairly direct physical explanation. In the upstream region the flow is nearly uniform ($u \approx 1$, $v \approx 0$), so viscous stresses and, with them, sensitivity to Re are small and the gradient of the loss with respect to Re, which reaches Re only through the viscous term of the PDE residual, ends up weak and noisy, and the parameter estimate is slow to move away from its initial guess $Re_0 = 20$, staying the farthest from the true value throughout the sweep. In the wake regions, by contrast, the viscous term is a sizeable part of the momentum balance, both in the recirculation bubble and in the recovering wake deficit. A change in Re therefore shows up as a measurable change in the local velocity and pressure field, which gives the optimizer a stronger signal to work with. The shear layers fall in between. Their gradients are large, but most of the zone is in the outer flow next to the recirculation bubbles, where pressure and convection carry most of the momentum balance. The accuracy in this zone is intermediate and almost insensitive to the number of sensors. Section IV.F puts numbers on this explanation.

Figure 8, which tracks Re during training, is consistent with this picture and reveals a further trend. In every zone, the estimate moves away from the initial guess and levels off well before the end of the joint Adam phase. It then remains almost constant throughout L-BFGS-B fine-tuning, even though the total loss decreases by up to an order of magnitude over the same interval. All zones show this behavior, and it is most evident upstream, where the curve is flattest and the estimate never approaches $Re = 40$. Our interpretation is that once the network has fitted the sensor and

physics data, the loss is nearly flat along the Re direction, so further descent refines the fit to the data while leaving Re largely unchanged. This matters for interpreting Table 1: the accuracy limits seen there are set by the information contained in the sensor data and by the conditioning of the inverse problem. An insufficient number of Adam iterations can be ruled out as the cause, because L-BFGS-B, which uses its own line search, ends at nearly the same Re. Across the entire N sweep, the wake zones rank first, and in Runs 1 and 3 the upstream zone ranks last, an ordering consistent with the physical explanation above.

The configuration of L-BFGS-B may also play a role. In our implementation, it operates on a single flattened vector containing several thousand network weights and one entry for $\log(1/Re)$. One step size is selected per iteration by line search, and the search direction is dominated by the weights because there is so various and several of them. Consequently, most of each step is spent refining the network's fit to the data and the physics residuals, and only a small fraction of it acts on Re. With the newest data we can not tell which of two things causes the plateau in Re. The gradient with respect to Re may be genuinely close to zero (a local optimum), or it may still carry information that is overwhelmed by the much larger block of weight gradients sharing the same step (Fig. 7). Updating Re and the weights in alternating blocks would distinguish between these two cases, and we leave this for future work.

The warm-up phase plays a smaller role. Holding Re at $Re_0 = 20$ for the first 1,000 epochs lets the network establish a reasonable velocity and pressure field before the estimate is allowed to change, which guards against gradient instability early in training. It does not, however, determine the final accuracy. In runs without any warm-up, where Re and the weights were initialized and trained together from the first epoch, the errors in every zone were very close to those obtained with the staged schedule, and the warm-up gave only slightly better absolute values. We retained it because it adds little cost and removes one possible source of early training noise.

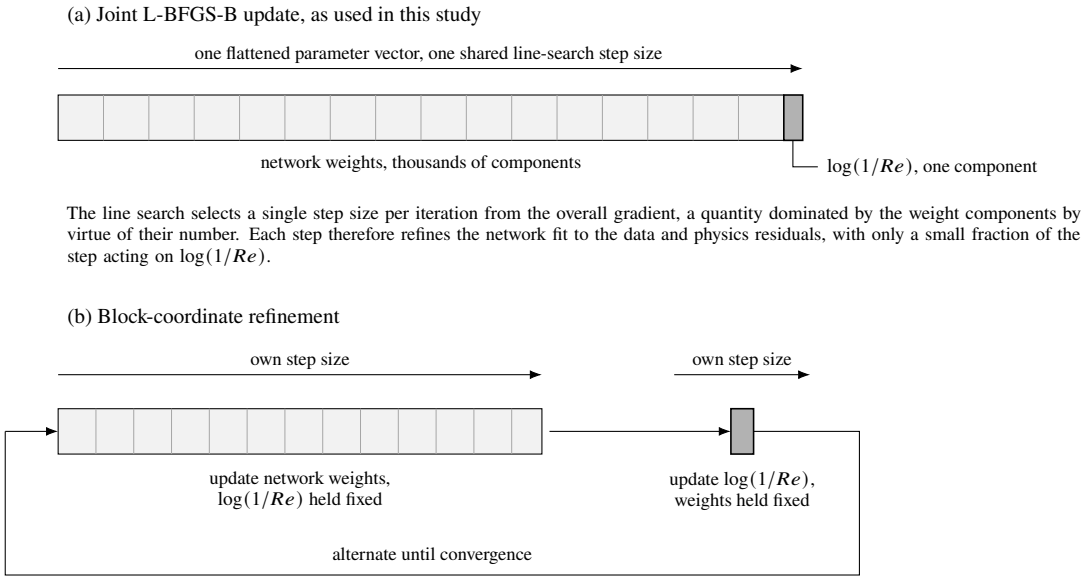


Fig. 7 Structure of the L-BFGS-B fine-tuning phase. (a) The optimizer operates on a single flattened vector containing the network weights alongside the scalar $\log(1/Re)$, so one shared step size governs both. (b) A block-coordinate scheme that updates $\log(1/Re)$ independently of the network weights would distinguish a true local optimum from gradient starvation of the parameter within the joint step.

D. Training Loss Diagnostics: Best vs. Worst Zone

To complement the Re training history comparison with a more mechanistic view of training itself, Fig. 9 breaks the total loss down into its data, PDE, and boundary condition components for the best performing zone (Z1, near wake) and the worst performing zone (Z4, upstream), across their respective sensor count sweeps and through both the Adam and L-BFGS-B phases. The PDE and boundary-condition losses are evaluated on the same domain-wide collocation and boundary points in every run, regardless of where the sensor zone is located, so their decay behavior should be broadly similar across zones, as the figure confirms. The data loss is the only term that changes with the sensor zone. It is built entirely from the sensor samples taken in that zone. Re appears only in the PDE term, so the gradient that updates it

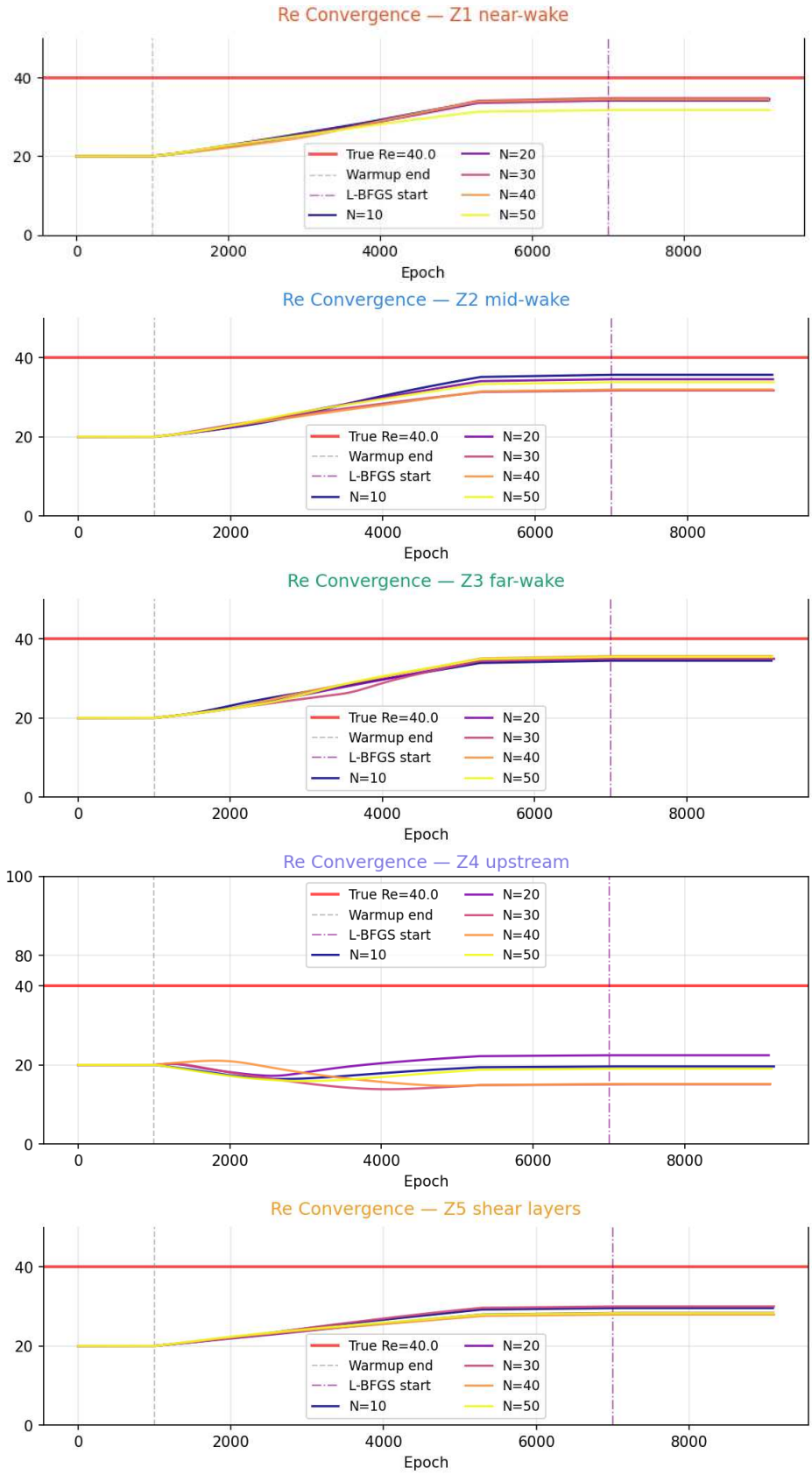


Fig. 8 Recovered Reynolds number vs. training epoch for all five zones (Run 3). The dash-dot line marks the start of L-BFGS-B fine-tuning; axes are truncated above the data range. Convergence is strongest in the near/far wake, moderate in the shear layers, and weakest upstream — a ranking consistent across all N.

comes from the PDE residual. The data term affects Re indirectly. It constrains the network's velocity and pressure field within the zone, and the PDE residual of that field then determines which value of Re fits best. The data loss therefore provides the most direct link between sensor location and recovery quality. Comparing the data loss curves of different zones side by side (Fig. 9) gives a training level view to go with the physical argument in Sec. IV.C, beyond what the final recovered Re alone shows. In both zones, the total loss drops sharply again at the start of L-BFGS-B, which appears as a step at the dash-dot line. The fine tuning phase is clearly improving the network fit, and Fig. 8 shows that this improvement barely moves Re.

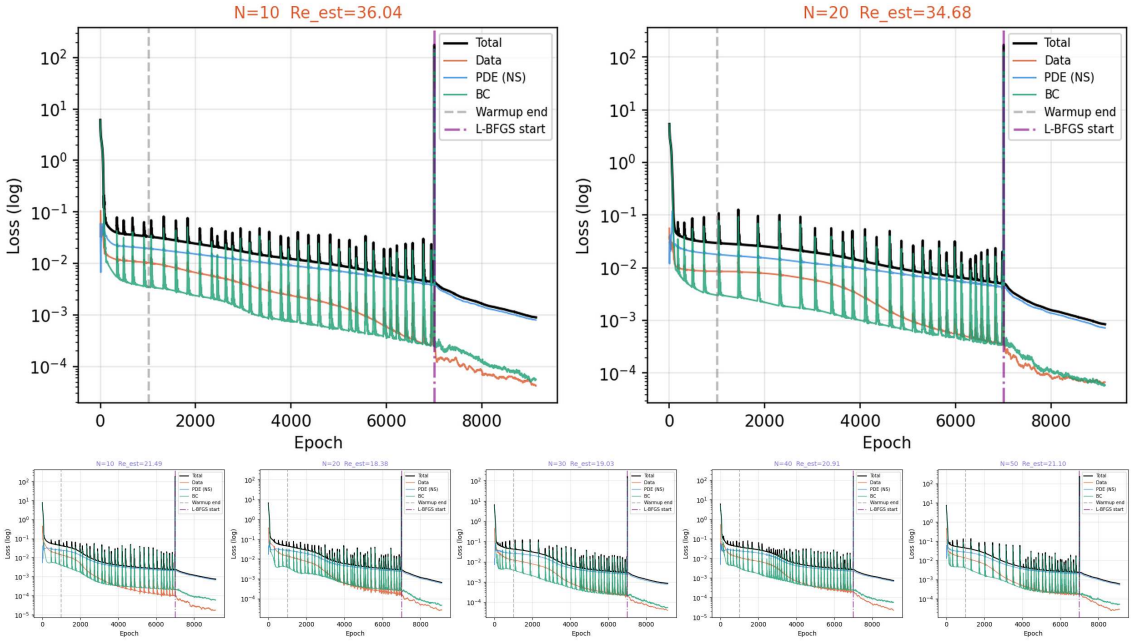


Fig. 9 Training loss decomposition (total, data, PDE, and boundary-condition components, log scale) versus epoch in Run 1 for the best-performing zone (top row, Z1 near wake, $N = 10$, 20) and the worst-performing zone (bottom row, Z4 upstream, $N = 10$–50), across the full warm-up + joint Adam + L-BFGS-B pipeline. The data-loss curve is the only component computed from zone-specific sensor samples, and is therefore the direct training-level link to the Re-recovery differences shown in Fig. 8.

E. Summary of Trends

Ranking the five zones by their best achieved error across the full Run 1 sweep gives: near wake (9.9% at $N = 10$), far wake (12.9% at $N = 30$), mid wake (13.2% at $N = 30$), shear layers (28.2% at $N = 10$), and upstream (46.3% at $N = 10$). In both Runs 1 and 3, the ranking by region holds at every sensor count: wake-region sensors outperform shear-layer sensors, which in turn outperform upstream sensors. For Run 2, Table 3 reports only the best errors. There, the wake zones still lead, but the upstream zone's best error (24.2%) is slightly lower than that of the shear layers (25.6%). Figure 10(a) shows this as a heatmap across the full zone–N grid, and Fig. 10(b) overlays the error-vs-N curves for all five zones; the error surface is structured far more strongly by *row* (zone) than by *column* (N), and the five curves in panel (b) stay cleanly separated and non-crossing between the wake/shear group and the upstream zone across the whole sweep – reinforcing that sensor location, not sensor count, is the dominant factor governing recovery accuracy in this problem.

F. What Makes a Zone Informative: Gradient Energy and Viscous Fraction

Section IV.A ranks the zones but does not say why one zone's sensor data tell the optimizer more about Re than another's. An obvious candidate is the size of the local velocity and pressure gradients. We compare it with a second quantity that comes straight from the governing equations.

Re enters Eqs. (2)–(3) only through the viscous term. The derivative of the momentum residual with respect to

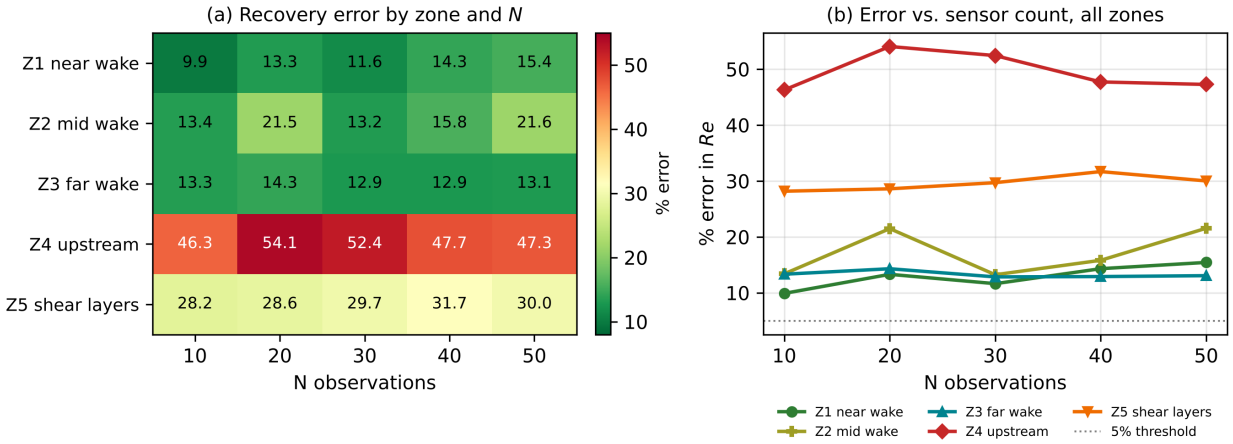


Fig. 10 Run 1 recovery errors: (a) heatmap of the error (%) for each zone and sensor count, $N = 10$–50, and (b) the same errors plotted against N, one curve per zone. The upstream zone (Z4) remains a clear outlier at every N, the near- and far-wake zones (Z1, Z3) are consistently strongest, and the shear-layer zone (Z5) sits at a stable intermediate level throughout.

$\nu = 1/Re$ is $-\nabla^2\mathbf{u}$, so a change in Re is visible only where the viscous term makes up a sizeable part of the local momentum balance. Elsewhere, the sensor data fit almost equally well over a wide range of Re, and the optimizer gets little signal to move the estimate. Steep gradients do not guarantee a large viscous term. Where the flow accelerates around the cylinder, for instance, the gradients are large, yet pressure and convection set almost all of the momentum balance. The two point-wise quantities we compare are the gradient energy

$$G = |\nabla u|^2 + |\nabla v|^2 + |\nabla p|^2, \tag{7}$$

and the viscous fraction

$$F = \frac{\nu\,|\nabla^2\mathbf{u}|}{|(\mathbf{u}\cdot\nabla)\mathbf{u}|}, \tag{8}$$

which measures the viscous term against the convective term. Equation (8) uses the true $\nu = 1/40$, so F depends only on the flow field and is independent of each model's recovered Re. Any constant ν would leave the rankings unchanged.

We computed G and F from the 25 trained Run 3 models (Table 2), differentiating each network exactly to get first and second derivatives. A model reproduces the synthetic data well only within its own zone, so it is evaluated only there. For the zone values in Table 5 we take medians over a uniform grid of spacing 0.02, excluding points with $r < 0.75$ (within 0.25 of the cylinder surface), where very large second derivatives would otherwise dominate. In Fig. 11, each model is represented by the median over its own N sensor points.

Table 5 Zone-level gradient energy G (Eq. 7) and viscous fraction F (Eq. 8), median over each zone, excluding $r < 0.75$. Errors are the Run 3 means over $N = 10$–50 (Table 2).

Zone	G	F	Mean Err. (%)
Z1 – Near wake	2.03	5.16	15.2
Z2 – Mid wake	0.88	0.66	16.2
Z3 – Far wake	0.20	1.04	12.2
Z4 – Upstream	0.025	0.019	54.3
Z5 – Shear layers	0.62	0.36	27.9

Gradient energy only separates the upstream zone. The gradients in Z4 are one to two orders of magnitude smaller than in any other zone (Table 5), so G does pick it out. Beyond that, it fails. The shear layers (Z5) have three

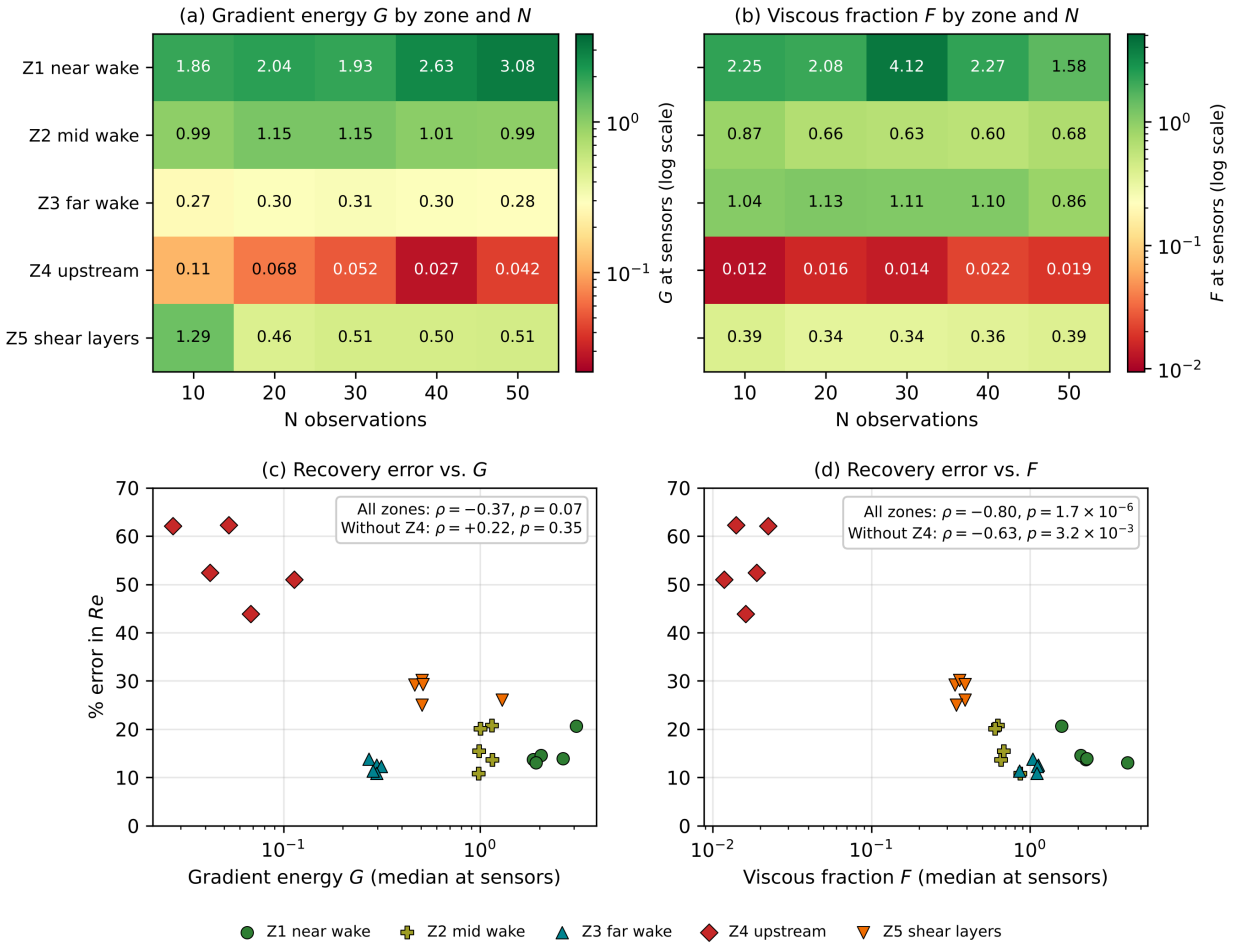


Fig. 11 Gradient energy G and viscous fraction F at the sensor points of all 25 Run 3 models (median over each model's N sensors). (a, b) Zone–N heatmaps in the same layout as Fig. 10(a), colored so that a quantity that explains recovery would show the zone pattern of the Run 3 errors (Table 2). (c, d) Recovery error against each quantity, with Spearman rank correlations for all zones and with the upstream zone (Z4) left out. G separates the upstream zone from the others but does not order the rest. Viscous fraction broadly follows the error ranking, placing the wake zones ahead of the shear layers and the shear layers ahead of the upstream zone.

times the gradient energy of the far wake (Z3) and more than twice its error. In Fig. 11(a), the far wake, which has the lowest mean error in Run 3 (Table 2), has some of the lowest values of G. Across the 25 models, the rank correlation between G at the sensors and the recovery error is weak (Spearman $\rho = -0.37$, $p = 0.07$; Fig. 11c), and it disappears once Z4 is left out ($\rho = +0.22$, $p = 0.35$).

Viscous fraction follows the ranking much more closely. F is consistent with the recovery results. The three wake zones have $F \geq 0.6$, the shear layers $F \approx 0.36$ and the upstream zone $F \approx 0.02$ (Table 5), which puts the three groups in the same order as their errors, and the heatmap of F in Fig. 11(b) broadly follows the zone pattern of the Run 3 errors in Table 2. Inside the wake group the order is not exact. F places Z1 above Z3, while Z3 has the lower mean error (12.2% against 15.2%). Across the 25 models, F at the sensors correlates strongly with the recovery error (Spearman $\rho = -0.80$, $p = 1.7 \times 10^{-6}$; Fig. 11d), and the correlation stays significant with Z4 left out ($\rho = -0.63$, $p = 3.2 \times 10^{-3}$).

The physical picture behind this is consistent across the zones. Inside the recirculation bubble behind the cylinder the flow is slow and strongly sheared, and the viscous term exceeds the convective term ($F > 1$). In the far wake, the gradients are gentle, but viscous diffusion drives the recovery of the wake deficit. The viscous and convective terms are of similar size there ($F \approx 1$), and the velocity data still carry information about ν. The shear-layer zone lies mostly in the outer flow alongside the recirculation bubbles. There, the gradients are steep, yet the momentum balance is dominated by pressure and convection ($F \approx 0.36$), and the sensors reveal correspondingly less about ν. In the upstream zone, the viscous term is significant only in a thin layer at the front of the cylinder, which explains why even a close fit to the Z4 data leaves Re poorly determined. For sensor placement, the deciding factor is how much of the local momentum balance the viscous term carries, and that the share can be large even though the flow changes slowly.

A few limits apply. The analysis uses a single replicate (Run 3), and it accounts only for differences between zones; within a zone, the change in error with N does not follow F consistently. The models also recovered Re below 40, so their second derivatives carry some bias, although the differences in F between zones span about two orders of magnitude. Finally, the correlations are associations across 25 trained models, and we offer them as supporting evidence for the explanation above.

V. Conclusions

We have trained a forward PINN for flow past a circular cylinder at $Re = 40$. The surface-pressure distribution was validated with the experimental data of Apelt [22]. We then used the validated field to generate a synthetic sensor data for training inverse PINN for recovering the Re. We tested five sensor zones with $N = 10$ to 50 sensors. It was found that sensor locations clearly affected the accuracy more than sensor count. In the first sweep (Run 1), the errors in Re in the near and far wake were between 9.9% and 15.4%. The mid wake gave similar errors, but they varied more (13.2–21.6%). The shear layers stayed in a narrow band of 28.2–31.7%. As expected, the upstream zone gave largest error consistently, with errors ranging from 46.3–54.1% at every sensor count. In Run 3 the upstream errors were 43.9–62.3%, and in Run 2 the upstream zone's best error (24.2%) was slightly below that of the shear layers (25.6%). More sensors did not monotonically lower the error in any zone. No combination of zone and N reached the 5% target. The L-BFGS-B optimizer although lowered the total loss by an order of magnitude Re hardly changed in any zone predictions. This indicates that the Re recovery depends more on how much information the loss landscape holds rather than just optimizer.

The zone ranking is consistent with the viscous fraction, the size of the viscous term relative to the convective term. It broadly follows the recovery errors, ranking the wake zones above the shear layers and the shear layers above the upstream zone, and it correlates strongly with the error across the 25 Run 3 models (Spearman $\rho = -0.80$). Gradient energy picks out only the upstream zone. The initial guess matters as well. Started from $Re_0 = 1$ or $Re_0 = 100$, the Z1 sweep failed at every N, with errors of about 97% and 129–285%.

To summarize, for a practical scenario sensors should be placed where the viscous term carries a sizeable share of the local momentum balance, such as the recirculation region behind the cylinder or the recovering far wake, and the initial guess for Re should be of the right order of magnitude. Even so, good placement alone is unlikely to give high absolute accuracy.

References

[1] Williamson, C. H. K., "Vortex Dynamics in the Cylinder Wake," *Annual Review of Fluid Mechanics*, Vol. 28, 1996, pp. 477–539. doi: 10.1146/annurev.fl.28.010196.002401.

[2] Jameson, A., "Aerodynamic Design via Control Theory," *Journal of Scientific Computing*, Vol. 3, No. 3, 1988, pp. 233–260. doi: 10.1007/BF01061285.

[3] Giles, M. B., and Pierce, N. A., "An introduction to the adjoint approach to design," *Flow, turbulence and combustion*, Vol. 65, No. 3, 2000, pp. 393–415.

[4] Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations," *Journal of Computational Physics*, Vol. 378, 2019, pp. 686–707. doi: 10.1016/j.jcp.2018.10.045.

[5] Raissi, M., Yazdani, A., and Karniadakis, G. E., "Hidden Fluid Mechanics: Learning Velocity and Pressure Fields from Flow Visualizations," *Science*, Vol. 367, No. 6481, 2020, pp. 1026–1030. doi: 10.1126/science.aaw4741.

[6] Rao, C., Sun, H., and Liu, Y., "Physics-Informed Deep Learning for Incompressible Laminar Flows," *Theoretical and Applied Mechanics Letters*, Vol. 10, No. 3, 2020, pp. 207–212. doi: 10.1016/j.taml.2020.01.039.

[7] Jin, X., Cai, S., Li, H., and Karniadakis, G. E., "NSFnets (Navier–Stokes Flow Nets): Physics-Informed Neural Networks for the Incompressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 426, 2021, p. 109951. doi: 10.1016/j.jcp.2020.109951.

[8] Kissas, G., Yang, Y., Hwuang, E., Witschey, W. R., Detre, J. A., and Perdikaris, P., "Machine Learning in Cardiovascular Flows Modeling: Predicting Arterial Blood Pressure from Non-Invasive 4D Flow MRI Data Using Physics-Informed Neural Networks," *Computer Methods in Applied Mechanics and Engineering*, Vol. 358, 2020, p. 112623. doi: 10.1016/j.cma.2019.112623.

[9] Cai, S., Wang, Z., Fuest, F., Jeon, Y. J., Gray, C., and Karniadakis, G. E., "Flow over an Espresso Cup: Inferring 3-D Velocity and Pressure Fields from Tomographic Background Oriented Schlieren via Physics-Informed Neural Networks," *Journal of Fluid Mechanics*, Vol. 915, 2021, p. A102. doi: 10.1017/jfm.2021.135.

[10] Sun, L., Gao, H., Pan, S., and Wang, J.-X., "Surrogate Modeling for Fluid Flows Based on Physics-Constrained Deep Learning Without Simulation Data," *Computer Methods in Applied Mechanics and Engineering*, Vol. 361, 2020, p. 112732. doi: 10.1016/j.cma.2019.112732.

[11] Cai, S., Mao, Z., Wang, Z., Yin, M., and Karniadakis, G. E., "Physics-Informed Neural Networks (PINNs) for Fluid Mechanics: A Review," *Acta Mechanica Sinica*, Vol. 37, No. 12, 2021, pp. 1727–1738. doi: 10.1007/s10409-021-01148-1.

[12] Jagtap, A. D., Kharazmi, E., and Karniadakis, G. E., "Conservative Physics-Informed Neural Networks on Discrete Domains for Conservation Laws: Applications to Forward and Inverse Problems," *Computer Methods in Applied Mechanics and Engineering*, Vol. 365, 2020, p. 113028. doi: 10.1016/j.cma.2020.113028.

[13] Ngo, S. I., and Lim, Y. I., "Solution and Parameter Identification of a Fixed-Bed Reactor Model for Catalytic CO2 Methanation Using Physics-Informed Neural Networks," *Catalysts*, Vol. 11, No. 11, 2021, p. 1304. doi: 10.3390/catal11111304.

[14] Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L., "Physics-Informed Machine Learning," *Nature Reviews Physics*, Vol. 3, No. 6, 2021, pp. 422–440. doi: 10.1038/s42254-021-00314-5.

[15] Chuang, P.-Y., and Barba, L. A., "Predictive Limitations of Physics-Informed Neural Networks in Vortex Shedding," *arXiv preprint arXiv:2306.00230*, 2023.

[16] Krishnapriyan, A. S., Gholami, A., Zhe, S., Kirby, R. M., and Mahoney, M. W., "Characterizing Possible Failure Modes in Physics-Informed Neural Networks," *Advances in Neural Information Processing Systems*, Vol. 34, 2021.

[17] Wang, S., Teng, Y., and Perdikaris, P., "Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks," *SIAM Journal on Scientific Computing*, Vol. 43, No. 5, 2021, pp. A3055–A3081.

[18] Joshi, S., and Boyd, S., "Sensor Selection via Convex Optimization," *IEEE Transactions on Signal Processing*, Vol. 57, No. 2, 2009, pp. 451–462.

[19] Krause, A., Singh, A., and Guestrin, C., "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies," *Journal of Machine Learning Research*, Vol. 9, 2008, pp. 235–284.

[20] Manohar, K., Brunton, B. W., Kutz, J. N., and Brunton, S. L., "Data-Driven Sparse Sensor Placement for Reconstruction: Demonstrating the Benefits of Exploiting Known Patterns," *IEEE Control Systems Magazine*, Vol. 38, No. 3, 2018, pp. 63–86. doi: 10.1109/MCS.2018.2810460.

[21] Dennis, S. C. R., and Chang, G.-Z., "Numerical Solutions for Steady Flow Past a Circular Cylinder at Reynolds Numbers up to 100," *Journal of Fluid Mechanics*, Vol. 42, No. 3, 1970, pp. 471–489.

[22] Apelt, C. J., "The Steady Flow of a Viscous Fluid Past a Circular Cylinder at Reynolds Numbers 40 and 44," Reports and Memoranda 3175, Aeronautical Research Council, London, 1961.

[23] Lu, L., Meng, X., Mao, Z., and Karniadakis, G. E., "DeepXDE: A Deep Learning Library for Solving Differential Equations," *SIAM Review*, Vol. 63, No. 1, 2021, pp. 208–228. doi: 10.1137/19M1274067.