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.

Interface-Aware Sparse Data Placement for Extended Physics-Informed Neural Networks Applied to Computational Fluid Dynamics and Fluid–Structure Interaction

Hamze El Halabi

hamzahalhalabi@gmail.com

Hokkaido University

Yusuke Takahashi

Hokkaido University

Research Article

Keywords: Physics-informed neural networks, Domain decomposition, Sparse-data reconstruction, Sensor placement, Fluid-structure interaction, NACA 0012

Posted Date: June 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9857188/v1

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

Additional Declarations: No competing interests reported.

Interface-Aware Sparse Data Placement for Extended Physics-Informed Neural Networks Applied to Computational Fluid Dynamics and Fluid–Structure Interaction

Hamze El Halabi[1,*] and Yusuke Takahashi[2]

[1]*Graduate School of Engineering, Hokkaido University, Kita 13 Nishi 8, Kita-ku, Sapporo 060-8628, Hokkaido, Japan*

[2]*Faculty of Engineering, Hokkaido University, Kita 13 Nishi 8, Kita-ku, Sapporo 060-8628, Hokkaido, Japan*

*** Corresponding author.** Hamze El Halabi,

Email: hamzahalhalabi@gmail.com

ORCID: 0009-0005-3130-946X

Co-author: Yusuke Takahashi, ytakahashi@eng.hokudai.ac.jp
ORCID: 0000-0002-6272-5758

Highlights

- XPINN reconstruction error concentrates at subdomain interfaces, not at flow features, suggesting interfaces as priority sites.
- Placing 10% of sparse measurements at interfaces cuts u-velocity error by 86 to 95 percent across four benchmarks.
- The 8% computational overhead trades small cost for order-of-magnitude accuracy gain in field reconstruction.

ABSTRACT

Reconstructing fluid flow fields from sparse measurements such as PIV or in-vivo flow imaging is a persistent challenge in applied fluid mechanics. Physics-Informed Neural Networks (PINNs) and their domain-decomposed variant, Extended Physics-Informed Neural Networks (XPINNs), offer a mesh-free framework combining governing equations with measurement data, but their accuracy degrades at the subdomain interfaces introduced by domain decomposition, where coupling is enforced only through soft penalty terms. We show that reconstruction error concentrates in these interface overlap regions, mirroring the decomposition geometry rather than the flow physics, suggesting that measurements placed there should be disproportionately informative. We test this through SPHINX (Sparse Physics-Hybrid Informed Neural eXtension), a data-placement strategy that allocates a fraction of the measurement budget to overlap regions and the remainder uniformly throughout the domain. A sensitivity study across allocation ratios from 0 to 50 percent identifies $f = 0.1$ as the u-velocity optimum, with interface-concentrated placement systematically outperforming uniform placement across four benchmarks spanning Re = 20 to 10,000 and external, internal, and fluid-structure interaction flows. SPHINX reduces the relative L2 error in u-velocity by 86 to 95 percent with only 200 sparse data points (0.4 percent of the reference mesh) while reducing run-to-run variance by one to two orders of magnitude. Comparison against radial basis function interpolation confirms that the gains stem from geometric placement coupled with the dual-anchoring mechanism enabled by overlapping subdomains, not from additional measurements alone. These results establish interface-aware sensor placement as a scientifically grounded, architecture-agnostic principle for sparse-data flow reconstruction.

Keywords: Physics-informed neural networks · Domain decomposition · Sparse-data reconstruction · Sensor placement · Fluid-structure interaction · NACA 0012

1. INTRODUCTION

When it comes to solving partial differential equations (PDEs) that govern fluid dynamics, the numerical solutions have traditionally relied on mesh-based discretization methods such as the finite element method (FEM) and the finite volume method (FVM). These methods are mature and well-validated; however, they require high-quality computational meshes. Generating these meshes takes a significant chunk of the time of the total analysis of the problem, which increases with increase in geometry complexity, especially when it involves fluid-structure interactions (FSI)[1]. Raissi et al.[2] introduced Physics-Informed Neural Networks (PINNs) which is a fundamentally different approach to solving PDEs. Without requiring explicit mesh generation, PINNs are trained to approximate the PDE solution by minimizing the equation residual at collocation points. This method has attracted significant attention for applications in fluid mechanics[3], heat transfer[4], and solid mechanics[5].

Despite their conceptual elegance, standard PINNs face several well-documented challenges. Neural networks with smooth activations exhibit spectral bias, causing preferential learning of low-frequency solution modes, which makes it difficult to capture fine-scale features and sharp gradients[6]. Competing loss terms (PDE residual, boundary conditions, data matching) affect training convergence due to the sensitivity of optimization to the balance among these terms, thus requiring careful hyperparameter tuning[7]. Furthermore, for complex geometries and multi-scale problems, the convergence rate and accuracy degrade substantially, which limits the practical applicability of single-network PINNs to relatively simple configurations[8].

Jagtap and Karniadakis[9] proposed Extended Physics-Informed Neural Networks in order to address the scalability limitation through domain decomposition. The computational domain is divided into subdomains; each assigned an independent neural network. Interface conditions enforcing value and normal-derivative continuity couple the subdomain solutions through penalty terms in the loss function. When it comes to problems with multiple spatial scales or complex geometries, XPINNs have shown improved accuracy and training efficiency when compared to conventional PINNs[9].

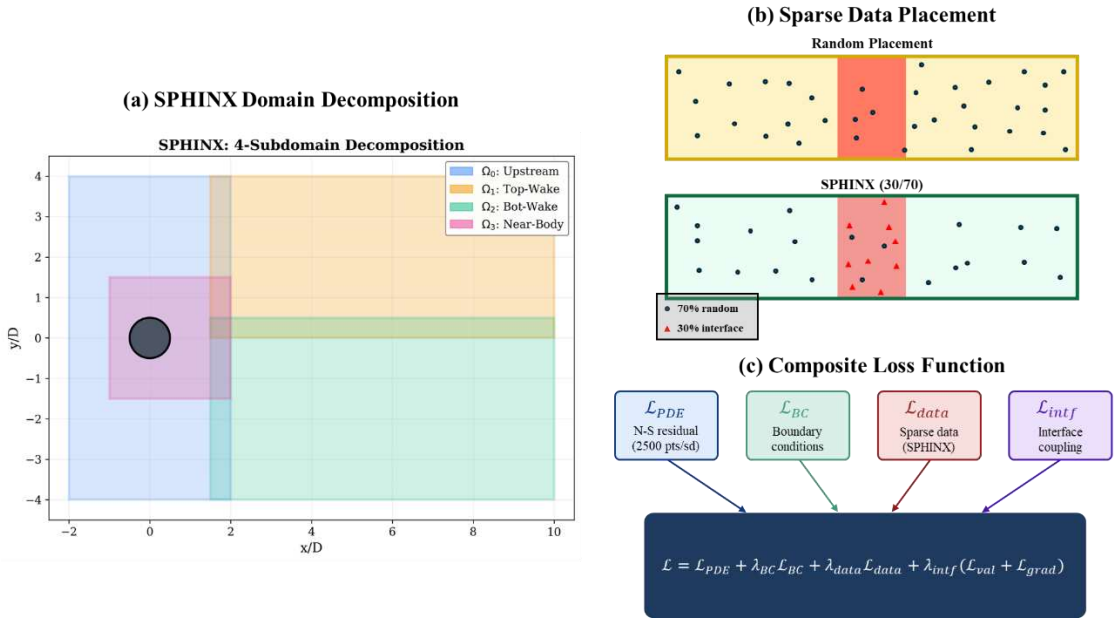


Figure 1 Schematic of the SPHINX framework. (a) XPINN domain decomposition with 4 subdomains and interface regions; (b) comparison of random vs. interface-aware data placement; (c) composite loss function diagram.

However, XPINNs introduce a characteristic failure mode: reconstruction error concentrates at subdomain interfaces. Because subdomain continuity is enforced only through soft penalty terms, the optimizer must balance PDE satisfaction within each subdomain against inter-subdomain consistency at the shared boundaries, and the optimizer can satisfy each subdomain's PDE residual to high accuracy while simultaneously violating value or normal-derivative continuity at the interfaces. The resulting spatial error pattern mirrors the decomposition geometry rather than the flow physics: errors are largest precisely where the soft constraints are weakest. This phenomenon is not unique to subdomain interfaces; physics-informed neural networks struggle at interfaces of all kinds. Henkes et al.[10] showed that PINN errors concentrate at material phase boundaries in heterogeneous solids, and Farea et al.[11] reported 12.9% pressure error near fluid–structure interfaces. Across these settings, the common feature is that interfaces, whether between computational subdomains or physical domains, represent the dominant source of reconstruction error. Several studies have addressed PINN convergence through adaptive training strategies[12, 13]; however, none have asked the converse question: if reconstruction error is concentrated at interfaces, should the measurement budget be concentrated there as well?

There is also research that has explored using sparse measurement data in order to improve the accuracy of PINNs. Data-augmented PINNs[14], which incorporate experimental or simulation data as an additional constraint, thus reducing solution ambiguity and improving the convergence.[15] However, existing adaptive point-selection methods such as PINNACLE[13] optimize placement based on training dynamics (e.g., neural tangent kernel analysis) without exploiting domain-decomposition structure or the localized weakness of XPINN interface coupling.

The practical relevance of accurate interface treatment becomes apparent when domain-decomposed PINNs are deployed in real fluid-mechanics applications. In experimental aerodynamics, surface pressure distributions on wings, blades, or bluff bodies are reconstructed from a finite number of pressure taps placed during model fabrication. In particle image velocimetry (PIV) and particle tracking velocimetry (PTV), the high-resolution measurement window covers only part of the flow field, and reconstructing the unmeasured regions requires combining the available data with the governing equations. In cardiovascular and respiratory flow imaging, in-vivo velocity fields must be inferred from sparse 4D-flow magnetic resonance imaging (MRI) data acquired under strict clinical constraints. In digital-twin frameworks for aerospace and energy systems, a small set of fixed on-board sensors must be combined with physical models to deliver real-time flow estimates. Each of these settings is fundamentally a sensor-placement problem: the engineer must allocate a fixed measurement budget across the spatial domain to maximize reconstruction accuracy.

The interface-coupling weakness of XPINNs has direct consequences here, because domain decomposition is precisely the strategy that enables PINNs to scale to the large or geometrically complex configurations encountered in such applications. Closing the interface gap is therefore not only a methodological refinement but a prerequisite for using domain-decomposed PINNs as practical tools for sparse-data flow reconstruction.

In this work, we propose SPHINX (Sparse Physics-Hybrid Informed Neural eXtension), a targeted data-assimilation strategy that tests the hypothesis above: measurements placed in subdomain overlap regions should provide disproportionately strong reconstruction guidance, because each such measurement simultaneously constrains two neighboring networks and provides direct supervision at the location where soft-constraint coupling is weakest. SPHINX allocates a fraction of the sparse measurement budget to interface overlap regions and the remainder uniformly throughout the domain. A sensitivity study across allocation fractions from 0 to 50 percent on multiple benchmarks identifies $f = 0.1$ as the practical operating point, with interface-concentrated placement systematically outperforming uniform random placement. Unlike the original XPINN formulation, which employs non-overlapping subdomains,[9] SPHINX adopts overlapping subdomain decomposition following the Schwarz-type approach also used in finite basis physics-informed neural networks (FBPINNs)[16], but retains the XPINN-style soft-penalty coupling rather than the FBPINN window-function blend. The strategy requires no modifications to the PINN architecture, loss-function weights, or training algorithm; it is purely a data-placement strategy that can be combined with any existing PINN framework.

We validate SPHINX on four benchmark problems spanning a range of geometries, Reynolds numbers, and physical complexity: (i) steady flow around a circular cylinder at Re = 40, (ii) steady flow over a NACA 0012 airfoil at Re = 1,000 at α = 5°, (iii) the same airfoil at Re = 10,000 at α = 5°, and (iv) the Turek-Hron FSI1 benchmark at Re = 20. In addition to demonstrating SPHINX effectiveness, we compare against radial basis function interpolation to isolate the contribution of geometric placement from the contribution of additional measurements, and analyze the spatial distribution of PDE residuals and interface continuity mismatch to verify the dual-anchoring mechanism. Across all cases, SPHINX reduces the relative L2 error in u-velocity by 85 to 93 percent using only 200 sparse data points, while collapsing run-to-run variance by two orders of magnitude.

The remainder of this paper is organized as follows. Section II reviews related work on PINNs, XPINNs, and data-augmented methods. Section III presents the SPHINX methodology. Sections IV and V describe the test cases and results, respectively. Section VI discusses limitations and future directions.

2. RELATED WORK

2.1. Physics-informed neural networks

PINNs which were introduced by Raissi et al[2] are a framework for solving forward and inverse partial-differential equation (PDE) problems. This is done by embedding the governing equations into the neural network training loss. In recent years, this approach has been utilized and applied to a wide range of fluid dynamics problems, including Navier–Stokes equations[3], Euler equations[17], and Stokes-Darcy coupled flow[18]. Theoretical analysis done by Shin et al.[19] established convergence properties of PINNs. Krishnapriyan et al[8]. have characterized the failure modes related to the spectral bias and multi-scale nature of PDF solutions.

In order to address these limitations, there have been multiple algorithmic and architectural improvements have been proposed. Loss balancing strategies[7] dynamically adjust the weights of competing loss terms during training. Adaptive activation functions[20] allow the network to learn the optimal nonlinearity for each layer. Residual-based adaptive refinement (RAR)[21] concentrates collocation points in high-residual regions. Fourier feature networks[22] mitigate spectral bias by mapping inputs through sinusoidal features at multiple frequencies. Single-network PINNs remain limited in their ability to handle more complex and multi-scale problems despite all these advances.

2.2. Domain decomposition for PINNs

Conservative PINNs (cPINNs) which were first proposed by Jagtap, Kharazmi, and Karniadakis[23] apply domain decomposition to conservation laws and enforce flux continuity at non-overlapping subdomain interfaces. Jagtap and Karniadakis[9] went on to further generalize this approach with extended PINNs (XPINNs). XPINNs generalize conventional PINNs by enabling arbitrary space-time decomposition of the computational domain. This allows multiple subnetworks to solve arbitrary PDEs over localized regions while maintaining global consistency through interface solution and residual continuity constraints. XPINNs were shown to be able to be parallelized across CPUs and GPUs by Shukla et. al[24]. This further improves the applicability of XPINNs to large-scale problems.

While the use of XPINNs can help with scalability issues, it also creates other problems. Mainly, interface coupling remains a persistent challenge. Since the optimizer must balance multiple competing loss terms, the penalty-based enforcement of continuity conditions creates a trade-off between subdomain accuracy and inter-subdomain consistency. To address this limitation, subsequent work tackled this through alternative coupling strategies. Hu et al.[25] proposed augmented PINNs (APINNs), in which hard interface conditions are replaced by a trainable gating network which performs soft domain partitioning.

In practice, the interface-coupling limitation manifests in three ways that are relevant to applied flow reconstruction. First, errors concentrate in the overlap regions between subdomains rather than distributing uniformly across the domain, producing a spatial error pattern that mirrors the decomposition geometry rather than the flow physics. Second, the optimizer can satisfy each subdomain's PDE residual to high accuracy while simultaneously violating value or normal-derivative continuity at the interfaces, because the loss function rewards both equally and the two objectives can be in tension. Third, these interface errors do not decrease monotonically with longer training: they reflect a soft-constraint balance dictated by the relative weights of the loss terms rather than a convergence issue that more iterations can resolve. Reweighting strategies that adjust the interface penalty during training can mitigate but not eliminate the underlying issue, because the fundamental difficulty is the absence of a hard mechanism for enforcing continuity. This motivates the data-driven anchoring strategy proposed in this paper: rather than reweighting an already-saturated penalty term, SPHINX provides direct supervision at the locations where the soft constraint is weakest.

2.3. Data-augmented PINNs

Incorporating sparse observational data into PINNs has been explored in many studies. In the original PINN formulation, Raissi et al.[2] included data matching as an optional loss term. Wang et al.[14] proposed a PINN-based method to enhance sparse PIV/PTV data into dense, high-resolution flow reconstructions thus enabling recovery of hidden pressure information. More broadly, Cai et al.[15] reviewed the different approaches of integrating data into PINNs in order to reconstruct flow fields from limited observations. However, most existing studies have paid limited attention to architecture-dependent error behavior when it comes to prescribing measurement locations.

2.4. PINNs for fluid–structure interaction

In more recent years, there have been a number of studies that delve into the application of PINNs to fluid-structure interactions (FSI). Early PINN formulations using immersed-boundary (IB) methods showed that physics-constrained interface conditions can capture fluid–solid coupling in standard FSI benchmarks[26]. More recently, they have been extended to unsteady interface motion and moving immersed boundary, which demonstrates the ability of fixed Eulerian representations in capturing the deformations of FSI interfaces[27]. Farea et al.[11] introduced an Eulerian–Lagrangian PINN framework with separate subnetworks for the fluid and structure. They reported that when using a unified PINN architecture, the pressure errors suffer (12.9% in structural regions), while the decomposed counterparts reduced the errors by more than 5× (2.39%).

Despite all the work that has been done, existing studies mostly focus on improving the network architecture and the IB coupling, while the use of targeted data placement for reducing localized fluid–solid boundary errors remain insufficiently explored.

3. METHODOLOGY

3.1. Governing equations

We consider the incompressible Navier–Stokes equations in their laminar form, without turbulence closure models. In dimensional form, the momentum and continuity equations are:

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u = -\frac{1}{\rho}\nabla p + \nu\nabla^2 u \quad (1)$$

$$\nabla \cdot u = 0 \quad (2)$$

where $u = (u, v)$ is the velocity vector, p is the pressure, ρ is the fluid density, and $\nu = {}^{\mu}/_{\rho}$ is the kinematic viscosity. The Reynolds number is defined as $Re = {}^{U_\infty L}/_{\nu}$ where U_∞ and L are the characteristic velocity and length scales of each test case (cylinder diameter for cases A and D, airfoil chord for cases B and C).

The governing equations are applicable in all the test cases considered in this paper, which lie in the laminar or laminar-to-transitional regime. For numerical implementation, the equations are non-dimensionalized using U_∞ and L as the velocity and length scales, with pressure normalized by ρU_∞^2; this is standard practice for PINN training and improves optimization conditioning by keeping all loss terms at comparable magnitudes.

For the steady benchmarks considered in this paper, the unsteady term ${}^{\partial u}/_{\partial t}$ vanishes. Extensions to unsteady flows with periodic vortex shedding, including the Turek–Hron FSI2 and FSI3 cases, are the subject of ongoing work and will be reported in a forthcoming paper.

3.2. XPINN domain decomposition

A central methodological choice of SPHINX is the use of overlapping subdomains. The original XPINN formulation [9] employs a non-overlapping (partitioned) decomposition in which interface conditions are imposed on a shared boundary of measure zero. While this is the natural choice for conservation-law applications such as those treated in cPINNs [23], it provides no spatial region in which two networks are simultaneously active, and therefore offers no location at which a single measurement can constrain two networks at once.

SPHINX adopts the overlapping, Schwarz-type geometry of finite basis physics-informed neural networks (FBPINNs)[16], in which adjacent subdomains share a two-dimensional overlap region of positive area where both networks produce predictions. We note an important distinction from FBPINNs in the coupling mechanism. FBPINNs combine the subnetwork outputs through compactly supported window functions $w_i(x)$ forming a partition of unity, so that the global solution $u(x) = \Sigma\, w_i(x)\, N_i(x;\, \theta_i)$ is a smooth weighted blend in the overlap. SPHINX retains the XPINN-style soft-penalty coupling: each subnetwork produces an independent prediction in the overlap, and continuity is enforced through the loss term L_{intf} rather than through a window-function blend. This design choice is deliberate. The soft-penalty coupling is precisely what gives rise to the interface error pattern that SPHINX is designed to address, and the dual-anchoring mechanism of which we touch on in section III.D exploits the existence of two independent predictions in the overlap region: a measurement at a point in the overlap provides a reference value against which both predictions are individually penalized, enforcing inter-subnetwork agreement through the data rather than through a penalty-balanced compromise.

We decompose the computational domain Ω into $K = 4$ overlapping domains $\{\Omega_1, \Omega_2, \ldots, \Omega_K\}$. The choice of $K = 4$ reflects the natural physical zoning of the test geometries considered in this paper. For external flows around a bluff body or airfoil, the flow field decomposes naturally into an upstream region dominated by the freestream, a near-body region containing the boundary layer and stagnation effects, and two wake regions separated by the body wake centerline. This four-zone partition matches the dominant spatial

structure of the solution. For the Turek-Hron channel geometry, the same four-region decomposition is retained but adapted to the channel walls.

Increasing K beyond four would in principle further reduce the representational burden on each subnetwork but introduces additional interfaces, which is precisely the location SPHINX is designed to anchor. The trade-off between subdomain specialization and interface count is geometry-dependent and is left to future work; the present results establish that $K = 4$ is sufficient to expose and characterize the interface-coupling weakness that SPHINX addresses. Each subdomain is assigned an independent neural network $N_k(\theta_k)$ parameterized by θ_k. When it comes to the network architecture, we utilize a fully connected feedforward network, having 8 hidden layers of 128 neurons each, Tanh activation and Xavier normal weight initialization. The network in each subdomain maps the spatial coordinates (x, y) to the flow variables (u, v, p): N_k: $(x, y) \longrightarrow (u, v, p)$.

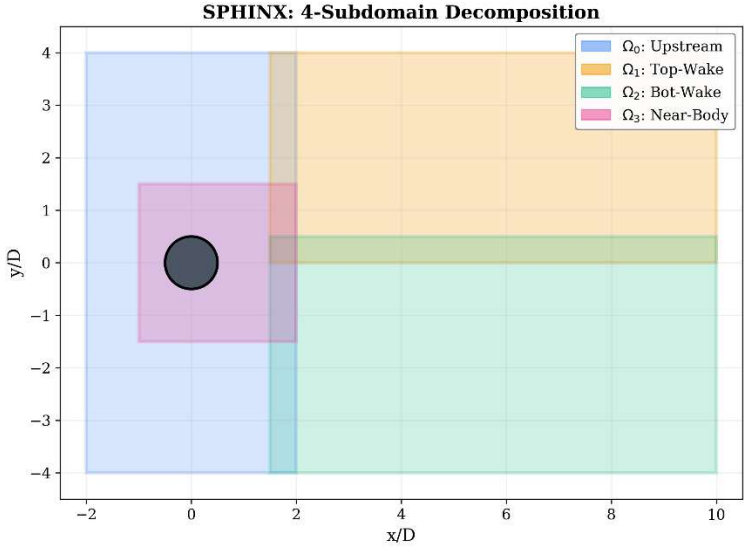


Figure 2 Domain decomposition for the cylinder case

Between adjacent subdomains at the interface ($\Gamma_{ij} = \Omega_i \cap \Omega_j$), we enforce two coupling conditions. The field values predicted by the two networks i and j must agree at all points in the overlap region. The second condition that is enforced states that the gradients projected onto the interface normal must also match. By enforcing these two conditions we ensure the continuity of both the solution and its normal flux across subdomain boundaries.

3.3. Composite loss function

The total training loss is a weighted sum of five terms:

$$\mathcal{L} = \mathcal{L}_{PDE} + \lambda_{BC}\mathcal{L}_{BC} + \lambda_{data}\mathcal{L}_{data} + \lambda_{intf}(\mathcal{L}_{val} + \mathcal{L}_{grad}) \quad (3)$$

where $\mathcal{L}_{PDE}$ is the mean squared PDE residual evaluated at $N_c = 2{,}500$ collocation points per subdomain (resampled each epoch); $\mathcal{L}_{BC}$ enforces boundary conditions (no-slip on solid surfaces, freestream or parabolic inlet at far-field/inlet boundaries); $\mathcal{L}_{data}$ matches network predictions to sparse measurement data at N points; $\mathcal{L}_{val}$ and $\mathcal{L}_{grad}$ enforce value and normal-derivative continuity at 500 interface points per subdomain pair. The penalty weights are set to $\lambda_{BC} = \lambda_{data} = \lambda_{intf} = 10$ for all experiments.

3.4. Interface-aware sparse data placement

Given a total budget of N sparse measurement points, SPHINX allocates a fraction f to the subdomain overlap regions $\Gamma_{ij} = \Omega_i \cap \Omega_j$ and the remaining $(1-f)N$ points to the bulk fluid domain, excluding solid regions. Interface points are distributed uniformly across the $K(K-1)/2 = 6$ interface pairs of the four-subdomain decomposition (in practice fewer than six are geometrically realized; see Fig. 1).

Two questions then arise: what value of f is optimal, and why? We address the first question empirically through the sensitivity study reported in section 5.1, which sweeps $f \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ across all four benchmarks. The study identifies $f = 0.1$ as the practical optimum, and we adopt this value for the main results.

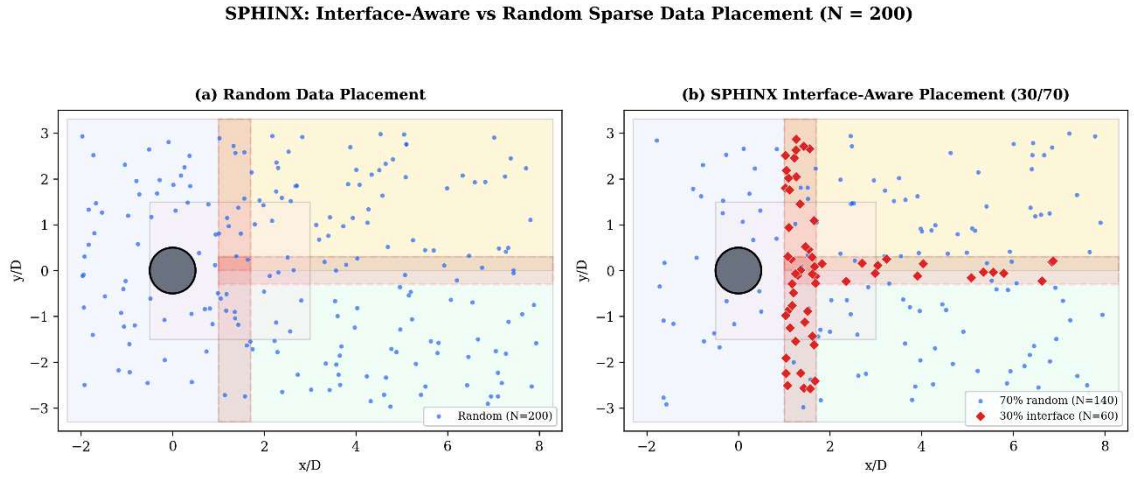


Figure 3 Illustration of interface-aware vs. random data placement: (a) random placement where interface regions receive few points; (b) SPHINX 30/70 placement with concentrated interface data.

The second question is best understood through the dual-anchoring mechanism enabled by overlapping subdomains. A measurement at a point $x \in \Gamma_{ij}$ enters the data loss as

$$\mathcal{L}_{data}(x) = \frac{1}{2}[[\, \|N_i(x;\, \theta_i) - f_{ref}(x)\|^2 + \|N_j(x;\, \theta_j) - f_{ref}(x)\|^2] \qquad 4$$

because both networks N_i and N_j are evaluated at x. The measurement therefore simultaneously

(i) constrains N_i and N_j individually to match $f_{ref}(x)$, and
(ii) constrains $N_i(x) \approx N_j(x)$ through the shared reference value.

Effect (ii) is precisely the inter-subdomain consistency that the soft interface loss $\mathcal{L}_{intf}$ is responsible for in the baseline XPINN, and the data anchor provides this consistency through an external reference rather than through a penalty-balanced compromise between the two networks. A measurement placed in the bulk (outside any overlap region) provides only effect (i), constraining a single network. The 2:1 ratio of constraints per measurement explains why a small allocation of interface points produces a disproportionate accuracy gain.

3.5. Training procedure

The network are all trained jointly using the Adam optimizer[28] with an initial learning rate of 10^{-3}, annealed to 10^{-5} over 100,000 epochs using a cosine schedule. In order to prevent overfitting to specific spatial configurations, collocation points and interface points are resampled at every epoch. In order to provide statistical error bars ($mean \pm standard\ deviation$), each experiment is repeated with three independent random seeds. Throughout this paper, "seed" refers to the integer used to initialize the pseudo-

random number generator that governs network weight initialization (Xavier normal distribution), collocation point sampling, interface point sampling, and sparse data placement. The three seeds used are 42, 123, and 456.

Errors are quantified using the relative L_2 norm:

$$E = \frac{\sqrt{\sum_{i=1}^{M}(f_{pred,i} - f_{ref,i})^2}}{\sqrt{\sum_{i=1}^{M} f_{ref,i}^2}} \times 100\% \quad (4)$$

where f represents one of u, v, or p, the index i runs over the M fluid nodes of the reference mesh, $f_{\text{pred},i}$ is the SPHINX prediction at node i, and $f_{\text{ref},i}$ is the corresponding reference computational fluid dynamics (CFD) value.

3.6. Fluid–structure interaction formulation

We adopt a fluid-only formulation for the FSI benchmark. The structural deformation is prescribed from a validated turtleFSI[29] reference solution. The beam surface position and velocity are prescribed from the validated turtleFSI reference solution, and the XPINN reconstructs only the steady fluid fields (u, v, p). This isolates the effect of SPHINX's interface-aware data placement on FSI-relevant geometries without coupling to a structural solver. The no-slip boundary condition on the deformed beam surface is enforced using the prescribed beam velocity $v_{beam} = {\partial d}/{\partial t}$, where d is the structural displacement

4. TEST CASES

In order to evaluate the SPHINX framework, we utilized four different benchmarks of increasing complexity: (A) steady flow around a circular cylinder at $Re = 40$, (B) steady flow over a NACA 0012 airfoil at $Re = 1{,}000$ with angle of attack $\alpha = 5°$, (C) steady flow over the same airfoil at $Re = 10{,}000$ with $\alpha = 5°$, and (D) the steady Turek–Hron FSI1 benchmark at $Re = 20$. For all four cases, we were solving for incompressible Navier-Stokes equations and the XPINN architecture is 8x128 fully connected networks per subdomain, and 4 subdomain per problem as previously mentioned. For each experiment, we ran three different trials (seeds 42, 123, and 456) to see the statistical errors.

4.1. Cylinder flow at Re = 40

For the first case, we consider an incompressible steady flow around a circular cylinder of diameter D at Reynolds number $Re = {UD}/{\upsilon} = 40$. In this case, the flow develops a steady symmetric recirculation zone behind the cylinder. For the computational domain, it spans $[-2,8] \times [-4,4]$ in units of D. The reference solution is obtained using FEniCS on a fine triangular mesh (~50,000 elements). The computed drag coefficient ($C_d = 1.561$) is consistent with the established literature range of $C_d \approx 1.5{\sim}1.6$ for steady cylinder flow at Re = 40.

We decomposed the domain into four overlapping subdomains (Fig. 2) which are: Upstream Ω_0, Top-Wake Ω_1, Bottom-Wake Ω_2 and Near-Body Ω_3. These subdomains are connected by six interfaces with value and normal-derivative matching. For each subdomain we assigned an independent 8-layer, 128 neuron network with Tanh activation, which yields approximately 116,000 parameters per subdomain. Training utilizes Adam optimizer with cosine-annealed learning rate ($10^{-3} \rightarrow 10^{-5}$), and it proceeds for 100,000 epochs. At each epoch, interface points (500 per interface pair) and collocation points (2,500 per subdomain) are resampled.

SPHINX utilizes sparse data which is generated from the reference CFD solution. We utilize an interface-aware placement strategy: 30% of the total N points are placed at the interface overlap regions, while the remaining 70% of the points are randomly distributed throughout the domain, excluding the cylinder

interior. In order to evaluate the sensitivity of SPHINX, we test different values of N {30, 50, 100, 200, 500}, and run each case three times.

4.2. NACA 0012 airfoil at Re = 1,000

For the second test case, we examined the flow at $Re = 1{,}000$ over a NACA0012 airfoil, with a chord length c and an angle of attack $\alpha = 5°$. In this case, we get a more complex flow field when compared to the first case. On the suction side we have a thin boundary layer, and at the trailing edge there is flow separation, as well as having asymmetric wake structures. For the reference solution, we generated the data using FEniCS on a refined mesh and confirmed the results by comparing the lift and drag results to published values for this Reynolds number[30, 31].

Similar to the cylinder case, the domain is decomposed into four subdomains which are Upstream Ω_0, Top-Wake Ω_1, Bottom-Wake Ω_2 and Near-Body Ω_3. On the airfoil surface, no-slip boundary conditions are enforced using cosine-spaced control points on the suction (upper) and pressure (lower) surfaces. Cosine spacing concentrates points near the leading and trailing edges, where the largest pressure gradients and curvature occur. At the inlet and far-field, freestream conditions are applied. When it comes to the architecture, it is identical to the cylinder case.

4.3. NACA 0012 airfoil at Re = 10,000

For the third case, we increased the Reynolds number to $Re = 10{,}000$ while reducing the angle of attack to $\alpha = 5°$. The ten-fold increase in Reynolds number generates a more complex wake pattern, steeper pressure gradients close to the airfoil surface, and noticeably thinner boundary layers ($\delta \sim Re^{-\frac{1}{2}}$).

When it comes to the domain decomposition, network architecture and training parameters, they are identical to those of the $Re = 1{,}000$ case. The angle of attack is kept at α = 5° for both Reynolds numbers to isolate the effect of Reynolds number from the effect of incidence angle. At α = 5° the flow remains attached with structured wake and a pressure-gradient-driven suction peak, avoiding massive separation that would require turbulence modeling. Similar to the first two cases, reference CFD solutions are produced using FEniCS and compared the results to published results[32, 33].

4.4. Turek–Hron FSI1 benchmark

The fourth and final test case we ran to demonstrate SPHINX's applicability to fluid-structure interaction is the Turek-Hron FSI1 benchmark. This case involves a steady incompressible flow at $Re = 20$ around a cylinder with an attached elastic beam in a channel. The channel is of height $0.41m$ and length $2.5m$, with a cylinder of diameter $D = 0.1m$ is centered at$(0.2, 0.2)$, with a thin elastic beam of length $0.35m$ and thickness $0.02m$ extending from the downstream side of the cylinder. Due to the low Reynolds number, the beam undergoes negligible deformation ($\sim 0.8\mu m$ tip displacement) and the flow reaches a steady state.

We treat the structural deformation from a validated turtleFSI reference solution as a prescribed moving boundary; thus, SPHINX reconstructs the fluid fields (u, v, p) only. By doing this, the PINN is not required to solve the coupled structural mechanics, but it isolates the contribution of interface-aware sparse data placement on an FSI-relevant geometry. We validated the reference solution against the benchmark tip displacement $d_x = 0.0227 \times 10^{-3} m$ reported by Turek and Hron[34].

When it comes to the boundary conditions, they include a parabolic inlet profile, as well as no-slip on the channel walls, cylinder surface, and deformed beam surface, and a zero-traction outflow condition. Similar to the cylinder case, the domain is decomposed into four fluid subdomains, adapted for the channel geometry with the cylinder-beam obstacle.

5. RESULTS

In this section we present the results in terms of the relative L_2 error for the three flow variables (u, v, p). The error is calculated by using $\varepsilon = {||f_{pred} - f_{ref}||_2}/{||f_{ref}||_2} \times 100\%$, where f is the field variable. Since we run three independent trials for each case, the results also include the mean and standard deviations.

5.1. Sensitivity to interface allocation

The central methodological claim of this work is that reconstruction error in domain-decomposed PINNs is structurally tied to subdomain interfaces, and that measurements placed in interface overlap regions are disproportionately informative compared to uniform-random placement. We test this claim through a systematic sensitivity study, sweeping the interface allocation fraction f from 0.0 (pure random placement) to 0.5 across all four benchmark cases. All other parameters are held fixed: N = 200 total measurement points, 8 × 128 networks per subdomain, 100,000 training epochs, three independent seeds per condition. The sweep was performed at α = 5° for the NACA cases to control for angle-of-attack confounding when comparing across Reynolds numbers; the resulting optimum allocation is governed by the decomposition geometry and is insensitive to angle of attack within the laminar regime considered. The full study comprises 4 cases × 6 fractions × 3 seeds = 72 independent trials.

5.1.1 Headline result: $f = 0.1$ is the consistent u-velocity optimum

Table 1 reports the resulting relative L2 errors in u-velocity, with mean and standard deviation across three seeds. In every case, $f = 0.1$ yields the lowest u-velocity error.

TABLE 1. *u*-velocity relative L2 errors (%) across the interface allocation sweep.
N = 200, three seeds, mean ± standard deviation. Bold marks the per-case minimum

Case	f=0.0	f=0.1	f=0.2	f=0.3	f=0.4	f=0.5
Cylinder Re = 40	1.498 ± 0.016	**1.421 ± 0.006**	1.441 ± 0.016	1.495 ± 0.008	1.438 ± 0.004	1.508 ± 0.009
NACA Re = 1k	1.560 ± 0.026	**1.132 ± 0.012**	1.315 ± 0.013	1.330 ± 0.028	1.675 ± 0.031	1.812 ± 0.026
NACA Re = 10k	2.712 ± 0.132	**1.908 ± 0.027**	2.827 ± 0.085	2.423 ± 0.303	3.213 ± 0.240	2.773 ± 0.190
FSI1 Re = 20	5.273 ± 0.001	**5.252 ± 0.020**	5.295 ± 0.012	5.339 ± 0.018	5.422 ± 0.002	5.306 ± 0.022

The magnitude of the improvement varies strongly across cases. Going from pure random placement ($f = 0.0$) to the optimum ($f = 0.1$), the u-velocity error decreases by 27.4 percent for NACA Re = 1,000 and 29.6 percent for NACA Re = 10,000, but only 5.1 percent for the cylinder and 0.4 percent for FSI1. The interface-aware effect is therefore strongest where the baseline reconstruction is most challenged. Where the network already represents the flow accurately from the available geometric and boundary-condition constraints, sparse data placement provides only marginal additional benefit.

5.1.2 Sensitivity curve shapes

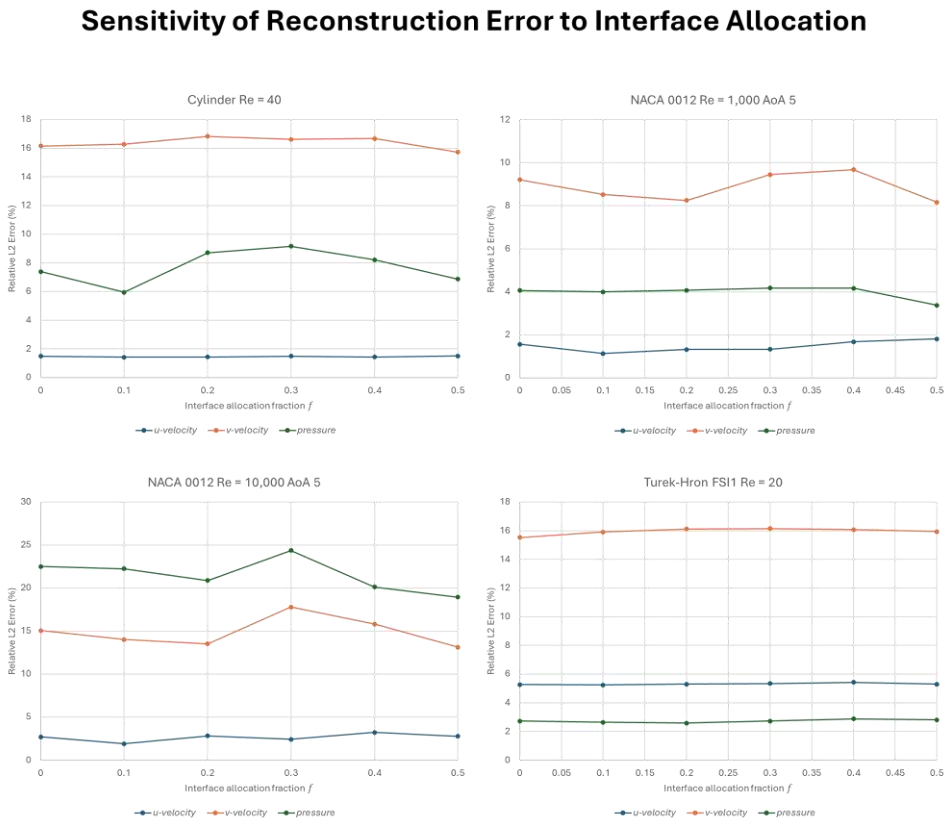


Figure 4 Four-panel sensitivity sweep showing u, v, and p error curves vs interface fraction for each case

NACA Re = 1,000 exhibits a clean U-shape on u-velocity with a clear minimum at $f = 0.1$ and a monotonic rise beyond $f = 0.2$; placing more than 30 percent of measurements in interface regions actively degrades u-velocity accuracy as bulk coverage suffers. NACA Re = 10,000 reaches its u-velocity minimum at the same fraction but with substantially higher seed-to-seed variability (standard deviation up to 0.30 percent at $f = 0.3$, compared to a typical 0.02 to 0.04 percent elsewhere), reflecting the increased optimization difficulty at higher Reynolds number. The cylinder and FSI1 curves are essentially flat on u-velocity: error varies by less than 6 percent relative across all fractions for cylinder, and by less than 3 percent for FSI1. The flatness of these cases indicates that the geometric and boundary-condition constraints alone are sufficient to anchor the reconstruction, with sparse data placement providing only refinement.

5.1.3 Variable-dependent optima

While $f = 0.1$ is the consistent u-velocity optimum, the v-velocity and pressure fields exhibit different patterns. Table 2 reports the case-by-case optimum allocation fraction for each flow variable.

TABLE 2. Per-variable optimum interface allocation fraction

Case	u optimum	v optimum	p optimum
Cylinder Re = 40	0.1	0.5	0.1
NACA Re = 1k	0.1	0.5	0.5
NACA Re = 10k	0.1	0.5	0.5
FSI1 Re = 20	0.1	0.0	0.2

The v-velocity optimum is $f = 0.5$ in three of four cases. The pressure optimum is $f = 0.1$ for cylinder, $f = 0.5$ for both NACA cases, and $f = 0.2$ for FSI1. This variable-dependent behavior reflects the asymmetric role of interface coverage in the composite loss. The u-velocity field is dominated by the freestream contribution and constrained more locally through the streamwise momentum equation, and benefits from concentrated anchoring at the smallest interface budget that secures inter-subdomain consistency. The v-velocity field is set primarily by chordwise asymmetry and wake dynamics, and benefits from broader interface coverage. The pressure field, determined globally through the Poisson coupling, similarly benefits from broader coverage in cases where wake-pressure structure is challenging.

For the remainder of this work we adopt $f = 0.1$ as the operating point, motivated by u-velocity as the conventional headline reconstruction metric and supported by its first-place ranking across all four cases on u and competitive ranking on v and p. A multi-objective Pareto ranking across all twelve (case, variable) combinations places $f = 0.1$ first (mean rank 1.92), substantially ahead of all alternatives (next best $f = 0.5$ at 2.83). The sensitivity curves of Figure 4 provide a direct guide for applications prioritizing variables other than u-velocity.

5.1.4 Connection to the dual-anchoring mechanism

The optimum at a small interface fraction is consistent with the dual-anchoring mechanism of Section 3.4. Each measurement placed in an overlap region simultaneously constrains two subnetworks through the shared reference value. A small number of such measurements suffices to enforce inter-subdomain consistency at any given interface; beyond this number, additional interface measurements yield diminishing returns and progressively crowd out the global coverage provided by the remaining $(1 - f)N$ bulk points. The marginal benefit of converting the next bulk measurement into an interface measurement decreases with f, while the marginal cost in lost bulk coverage is roughly constant. The empirical finding that the optimum occurs near $f = 0.1$ across geometries suggests that this crossover is governed primarily by the decomposition geometry, specifically the ratio of overlap area to total subdomain area, rather than by the specific flow configuration. A predictive theoretical model of f^* as a function of overlap width and subdomain count is left to future work.

5.2. Cylinder Re = 40

For the cylinder case at Re = 40, we show the error scaling for interface-aware approach vs baseline (Fig.6) as well as the predicted flow fields for the best SPHINX case (Fig. 7).

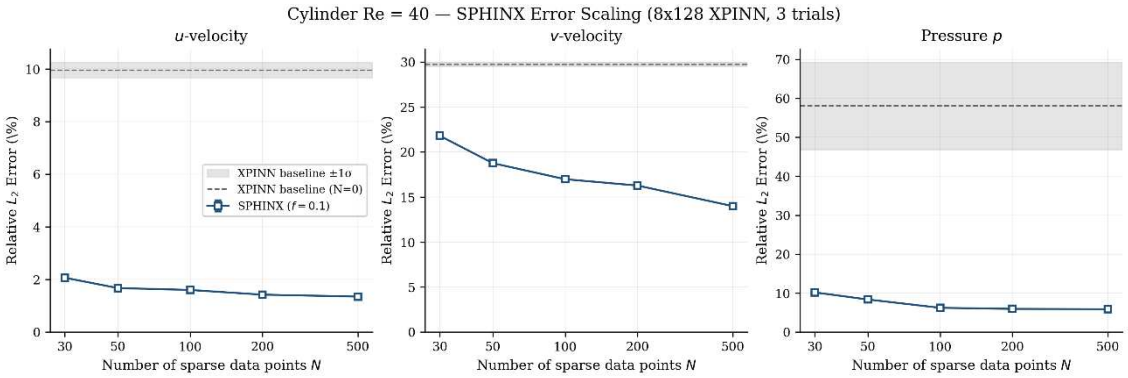


Figure 5 Cylinder Case Error Scaling

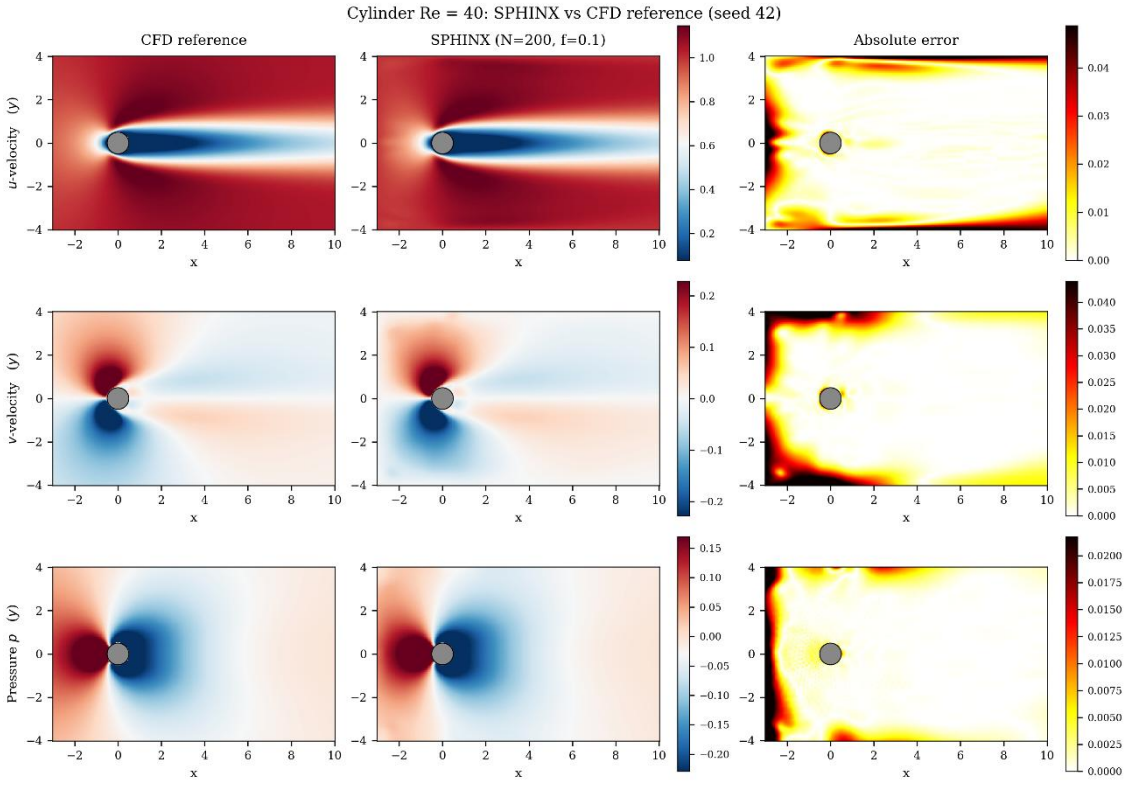


Figure 6 Cylinder Fields (CFD reference vs. SPHINX N = 200)

We also summarize the results comparing the results of the baseline XPINN to the SPHINX results at five different values of sparse data (30, 50, 100, 200, and 500). The baseline case ($N = 0$) shows substantial errors: 9.95% in *u*-velocity, 29.75% in *v*-velocity, and 58.04% in pressure. While the velocity components don't exhibit much deviation between runs, pressure exhibits a lot of variance (±11.20%) which indicates that the baseline XPINN struggles to satisfy both the pressure Poisson equation as well as the incompressibility constraint.

TABLE 3. Cylinder Re = 40: Relative L_2 errors (%) for SPHINX with interface-aware placement. 8×128 XPINN, 100k epochs, 3 trials.

Method	**u (%)**	**v (%)**	**p (%)**
XPINN Baseline	9.95 ± 0.30	29.75 ± 0.22	58.04 ± 11.20
SPHINX N = 30	2.066 ± 0.011	21.812 ± 0.023	10.186 ± 0.049
SPHINX N = 50	1.671 ± 0.020	18.761 ± 0.113	8.362 ± 0.178
SPHINX N = 100	1.600 ± 0.020	16.981 ± 0.067	6.229 ± 0.075
SPHINX N = 200	1.421 ± 0.006	16.280 ± 0.045	5.944 ± 0.014
SPHINX N = 500	1.348 ± 0.010	13.989 ± 0.014	5.890 ± 0.070

When we introduce even a minimal amount of sparse data ($N = 30$), we notice dramatic improvement in measurements, with the *u*-velocity error being reduced by 79.2% (9.95% to 2.066%). The pressure errors also dropped significantly (58.04% to 10.186%) but more importantly the deviations dropped to 0.049 from 11.2 which is significant improvement. This shows that a small number of strategically placed data points is sufficient when it comes to anchoring the XPINN solution to the correct flow physics. As *N* increases, the velocity errors drop as well reaching 1.348% and 13.989% respectively at $N = 500$. It is worth noting that the standard deviations across all SPHINX trials are extremely small (typically < 0.1%), which shows the high reproducibility of the SPHINX results once sparse data is provided.

Figure 7 shows the predicted flow fields for the best SPHINX trial ($N = 200$, interface-aware placement) alongside the CFD reference and the pointwise absolute error. When it comes to the *u*-velocity field, it captures both the upstream stagnation region as well as the downstream wake structure. As for the absolute error, it is concentrated in the immediate vicinity of the cylinder surface and the near-wake region, where the gradients are steepest.

5.3. NACA 0012 airfoil at Re = 1,000

For the second case, we present the results of a more challenging case which is the flow around the NACA 0012 airfoil at Re = 1,000. As expected, the baseline XPINN performs poorly with the *u*-velocity and *v*-velocity errors at 24.40% and 59.66% respectively. The pressure error reaching 97.89% indicates that the baseline XPINN essentially fails to capture the pressure field. It is also interesting to note that the variance for both velocity fields is higher (0.96 for *u*-velocity field and 9.78 for *v*-velocity field) while the pressure variance is lower (4.66) when compared to the first case.

TABLE 4. NACA 0012 Re = 1,000, α = 5°: Relative L_2 errors (%) for SPHINX with interface-aware placement.

Method	u (%)	v (%)	p (%)
XPINN Baseline	24.40 ± 0.96	59.66 ± 9.78	97.89 ± 4.66
SPHINX N = 30	2.954 ± 0.038	15.067 ± 0.458	7.461 ± 0.295
SPHINX N = 50	4.069 ± 0.091	13.594 ± 0.566	6.869 ± 0.402
SPHINX N = 100	3.636 ± 0.018	15.708 ± 0.276	8.224 ± 0.146
SPHINX N = 200	1.132 ± 0.012	8.528 ± 0.153	3.998 ± 0.063
SPHINX N = 500	1.388 ± 0.014	10.521 ± 0.107	4.846 ± 0.071

When compared to the first case, SPHINX achieves higher relative improvements on the NACA case. With only $N = 30$ data points, the *u*-velocity error shows an 88% reduction, by dropping from 24.40% to 2.954%. This improvement continues with the increase of sparse data, reaching 1.132% at $N = 200$, which is a 95% reduction from the baseline. Similarly, the pressure error also drops significantly with increasing sparse data, exhibiting a 95% reduction at $N = 200$ (97.89% at baseline to 3.998% at $N = 200$).

An unexpected feature of these results is that the *u*-velocity error increases from 1.132% at N = 200 to 1.388% at N = 500 (23.5% degradation). This non-monotonic behavior in the velocity component reflects a capacity–consistency trade-off that is specific to the data-augmented PINN setting.

When the number of sparse data points is small relative to the network's representational capacity, each additional point provides new information that constrains the solution toward the reference. As N grows, however, the data-loss term begins to dominate the optimizer's attention in the immediate neighborhood of each measurement point. The network preferentially fits the local data values, which can pull the solution off the manifold of solutions that satisfy the PDE residual in the surrounding region. Because the PDE residual loss is evaluated on a fixed set of collocation points that do not coincide with the data points, the

optimizer can drive the data loss to near zero while degrading PDE consistency at nearby collocation points, an effect that has been described qualitatively in prior work on data-augmented PINNs.

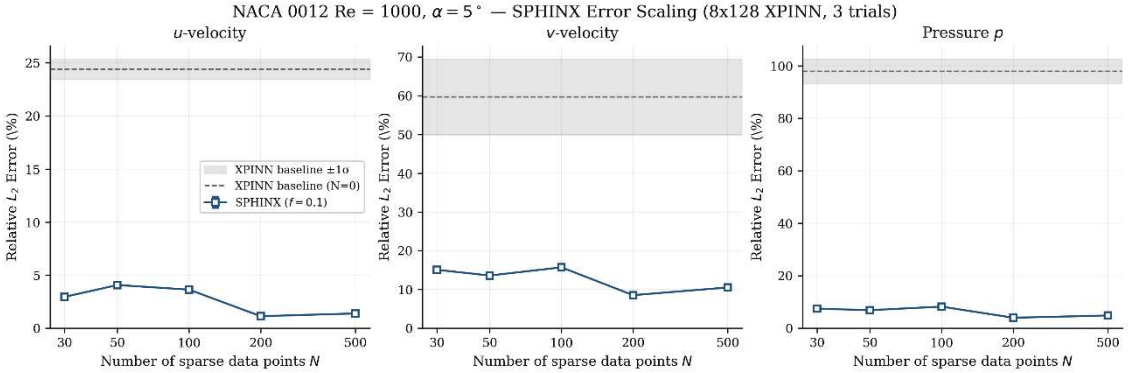


Figure 7 NACA 0012 Re = 1,000 Error Scaling

Figure 7 compares the error scaling for interface-aware versus purely random sparse data placement. Across all data budgets, the interface-aware strategy outperforms the random placement strategy across the three flow variables. At the optimal operating case ($N = 200$), u-velocity error shows a 41% improvement (1.87% for the interface aware placement vs. 3.18% for random placement). This is even more pronounced when we consider the v-velocity errors, with the interface-aware placement achieving 6.97% compared to the 12.20% of the random placement, reflecting the sensitivity of the transverse velocity component to accurate subdomain coupling in the wake region. The bigger gap in results shows that with the increasing complexity in geometries and flows, interface-aware data strategy becomes more valuable.

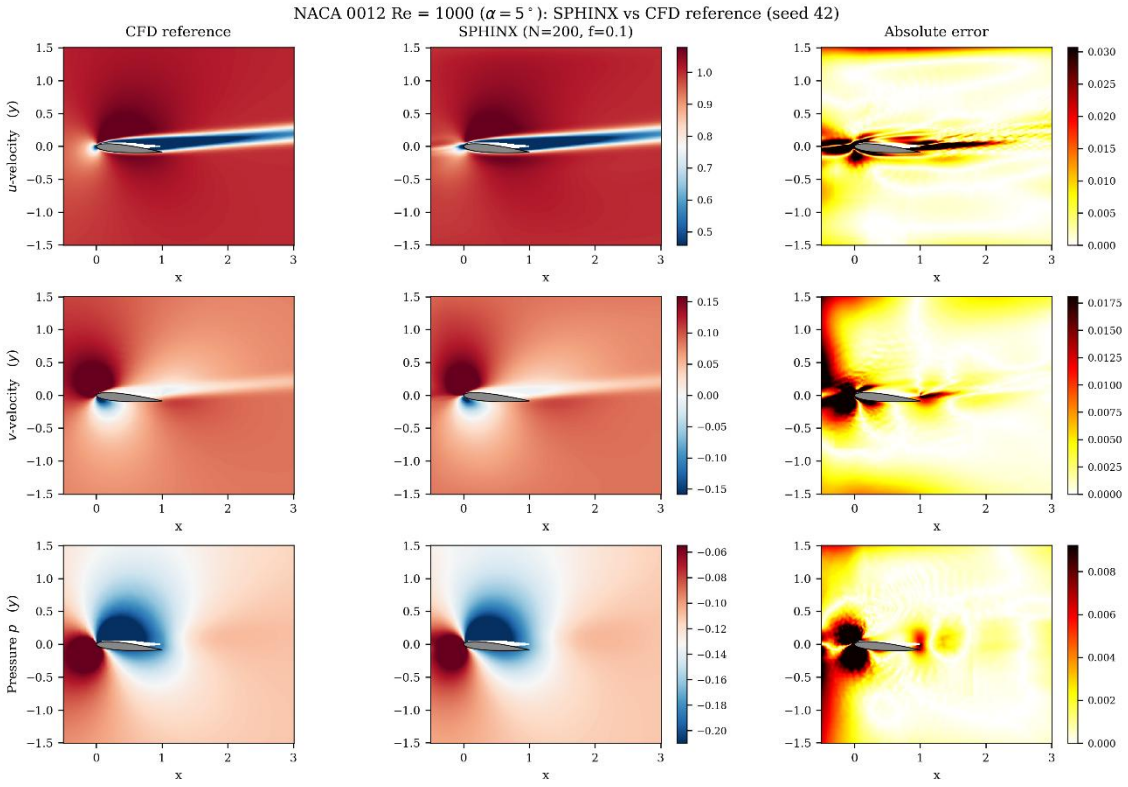


Figure 8 NACA 0012 Re = 1,000 Fields (CFD reference vs. SPHINX N = 200)

Figure 8 shows the predicted flow fields for the best SPHINX trial ($N = 200$, interface-aware placement) alongside the CFD reference and the pointwise absolute error. The u-velocity field is reconstructed with high fidelity, as the stagnation point, suction-side acceleration and wake structures are all clearly captured. The errors are concentrated around the airfoil surface and the near-wake region due to the steep gradients, and they are negligible in the far-field. The v-velocity is also accurately reproduced, showing the asymmetric flow pattern caused by the angle of attack. The errors are mostly at the trailing edge wake where complex transverse velocity structures are hardest to resolve. Lastly, when it comes to the pressure field reconstruction, it is particularly notable considering that the baseline XPINN basically fails to capture the pressure field. SPHINX recovers the leading-edge stagnation pressure, suction-side low-pressure region as well as the wake pressure recovery, with the errors mostly localized at the trailing-edge separation zone and the stagnation point.

5.4. NACA 0012 airfoil at Re = 10,000

Case three shows evidence of how SPHINX performance scales with the increase in Reynolds number. The baseline XPINN errors are comparable in both Re cases (u-velocity field errors 24.12% vs. 24.40%), thus suggesting that the baseline failure is due to the absence of data guidance rather than flow complexity.

TABLE 5. NACA 0012 Re = 10,000, α = 5°: Relative L_2 errors (%) for SPHINX with interface-aware placement.

Method	u (%)	v (%)	p (%)
XPINN Baseline	24.12 ± 0.28	58.36 ± 2.12	196.48 ± 74.41
SPHINX N = 30	4.770 ± 0.050	22.415 ± 0.300	34.252 ± 0.766
SPHINX N = 50	5.812 ± 0.496	17.590 ± 1.259	28.255 ± 2.621
SPHINX N = 100	4.972 ± 0.230	21.775 ± 1.090	35.151 ± 2.300
SPHINX N = 200	1.908 ± 0.027	14.024 ± 0.355	22.246 ± 0.483
SPHINX N = 500	2.114 ± 0.132	15.887 ± 0.222	23.305 ± 0.234

At $N = 200$, SPHINX achieves a 92% reduction in u-velocity error (24.12% → 1.908%), which is a similar reduction achieved at Re = 1,000 (92% reduction). The similarity in results demonstrates that SPHINX is equally effective at higher Reynolds numbers, with the relative improvement being Reynolds number independent.

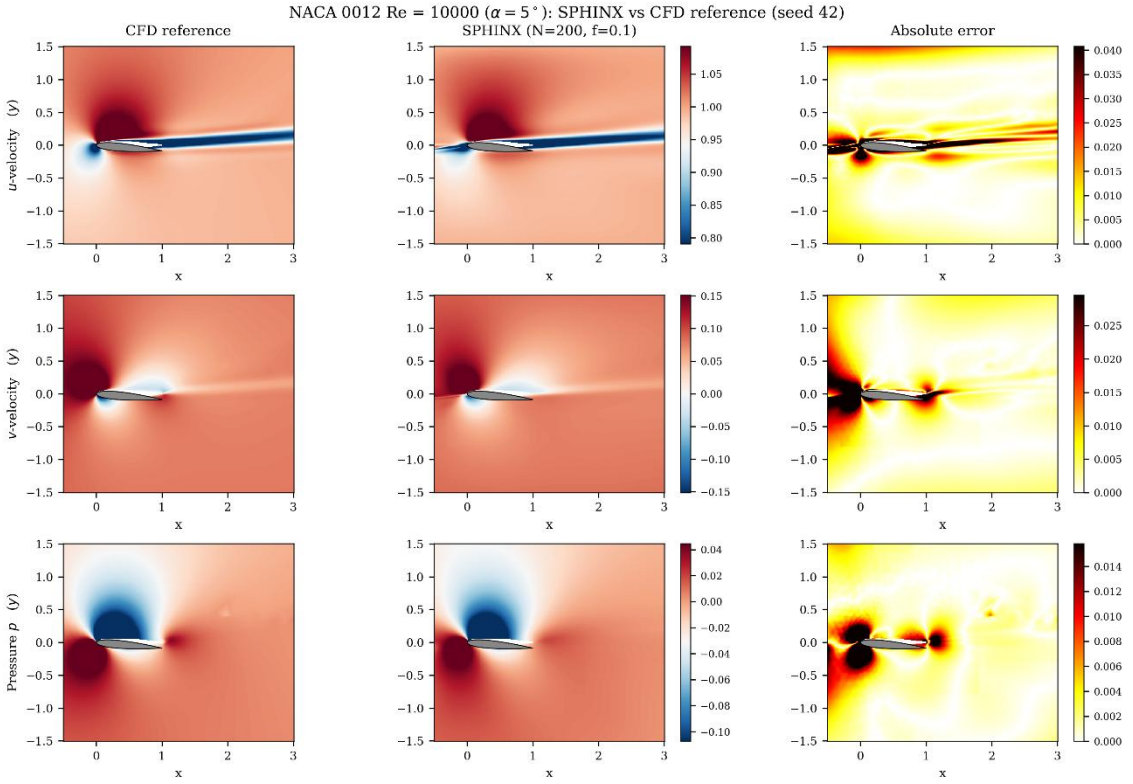


Figure 9 NACA 0012 Re = 10,000 Fields (CFD reference vs. SPHINX N = 200)

However, when comparing the absolute errors for Re = 10,000 to those of Re = 1,000 we find that they are systematically higher, thus revealing the impact of spectral bias on the underlying XPINN architecture. At $N = 200$, u-velocity exhibits a 69% increase (1.132% to 1.908%) which is a modest amount when compared to the 5.6-fold increase in the pressure error (3.998% to 22.246%). These results are consistent with the spectral bias of neural networks with smooth activation functions. The u-velocity field is dominated by the freestream component and remains relatively smooth, whereas the v-velocity and pressure fields exhibit finer spatial structures that become increasingly challenging to resolve at higher Reynolds numbers.

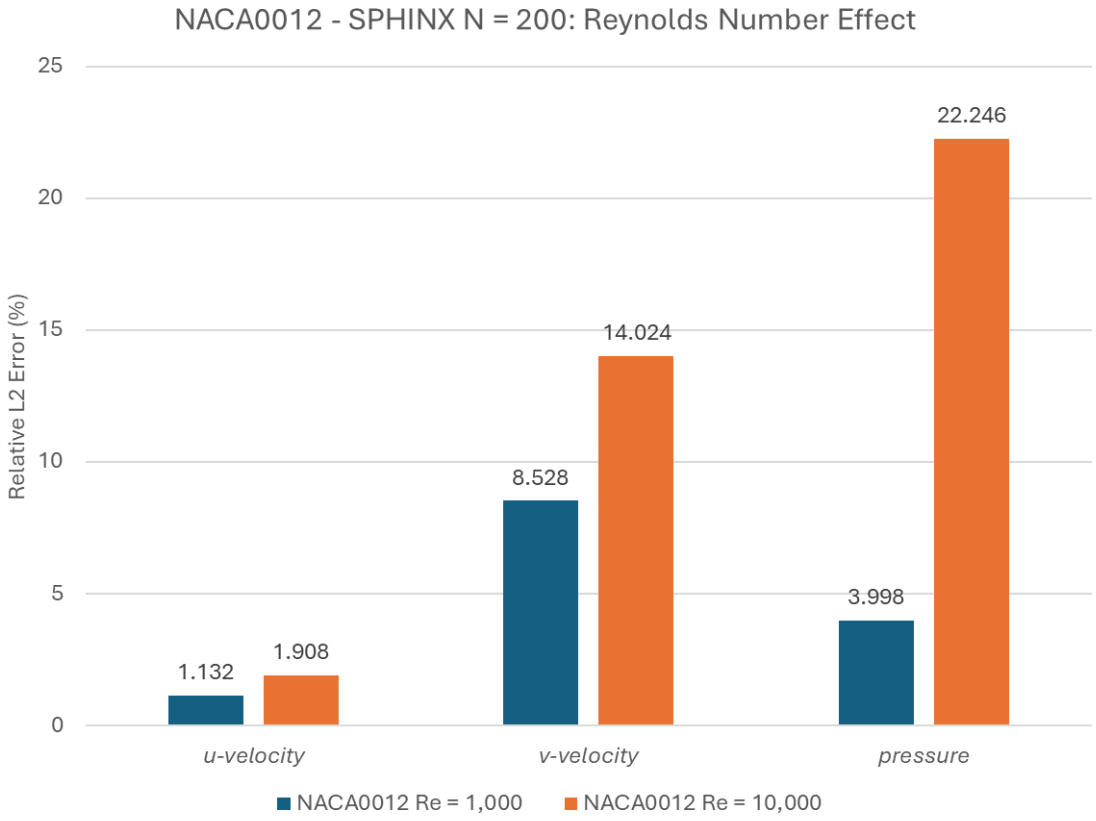


Figure 10 NACA 0012 Re = 1,000 vs. Re = 10,000

One really informative result is the pressure error degradation. For high Reynolds number flows (Re = 10,000), the pressure field develops steep gradients near the leading-edge stagnation point and along the suction side, with spatial scales proportional to the boundary layer thickness $\delta \sim Re^{-\frac{1}{2}}$. The network we utilized, having a fixed representational capacity, can't fully resolve those features. From that we can infer that if we want to increase Reynolds number to even higher values, it would be beneficial to extend the SPHINX architecture from the 8×128 structure we are employing now, by either utilizing a multi-scale network architecture or Fourier feature networks that can capture a broader range of spatial frequencies.

It is also important to note that the variance has increased from ±0.153 for the Re = 1,000 *v*-velocity field case to ±0.355 in the Re = 10,000 case, which shows that the optimization landscape becomes more rugged with increasing flow complexity. However, the variance is still vastly lower than that of the baseline case (±0.355 vs. ±2.12).

5.5. Turek–Hron FSI1 benchmark

The last case that we worked on is the Turek-Hron FSI1 benchmark to see the compatibility of SPHINX with fluid-structure interaction simulations. The XPINN baseline produced errors that are higher (49.86% in *u*-velocity, 44.75% in *v*-velocity, and 94.22% in pressure) than those of the simpler Re = 40 cylinder case due to the added complexity of the channel geometry, the parabolic inlet profile, and the non-trivial obstacle shape formed by the cylinder-beam combination.

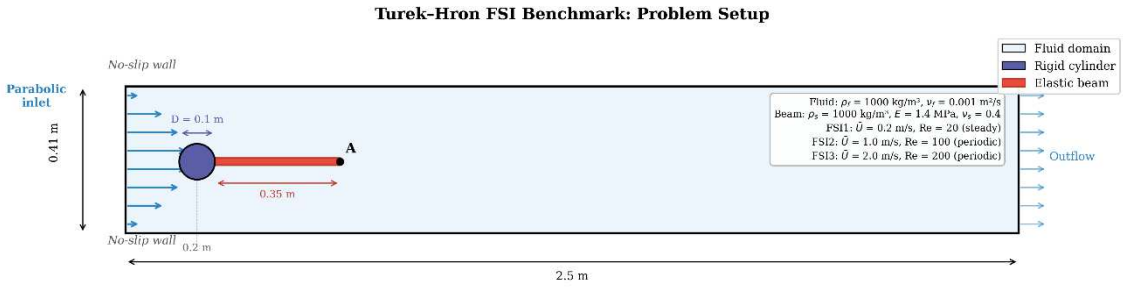


Figure 11 Turek-Hron FSI Benchmark

TABLE 6. Turek–Hron FSI1 (Re = 20, steady): Relative L_2 errors (%). 8×128 XPINN, 100k epochs, 3 trials.

Method	u (%)	v (%)	p (%)
XPINN Baseline	49.86 ± 7.72	44.75 ± 3.45	94.22 ± 19.99
SPHINX N = 30	6.773 ± 0.025	19.331 ± 0.051	5.645 ± 0.044
SPHINX N = 50	5.830 ± 0.008	17.668 ± 0.124	4.374 ± 0.046
SPHINX N = 100	5.371 ± 0.010	16.161 ± 0.072	3.193 ± 0.037
SPHINX N = 200	5.252 ± 0.020	15.904 ± 0.101	2.647 ± 0.025
SPHINX N = 500	5.049 ± 0.011	14.352 ± 0.066	2.160 ± 0.020

With $N = 200$ of interface-aware sparse data, SPHINX reduced the u-velocity by 89% (5.252% vs 49.86%), and a remarkable 97% reduction in pressure error (2.647% vs 94.22%). This pressure result is particularly notable as it is the lowest error percentage of all the cases we ran, including the simpler cylinder and NACA geometries. This suggests that the FSI1 pressure field, while challenging for an unsupervised XPINN, is well-constrained by the channel geometry once sparse data anchoring is provided.

These results confirm that SPHINX's interface-aware sparse data placement transfers effectively from external flow geometries (cylinder, airfoil) to internal flow with complex embedded obstacles.

5.6. Comparison with Radial Basis Function (RBF) interpolation

To address whether the SPHINX accuracy gains stem from the placement of sparse measurements or from the additional information they provide, we compare against radial basis function (RBF) interpolation as a traditional sparse-data reconstruction method that does not exploit the underlying PDE. RBF interpolation has been widely used in CFD post-processing and surrogate modelling for sparse-to-dense field reconstruction[35] and serves as a strong purely-data-driven baseline.

We use thin plate spline RBF, the standard default for two-dimensional scattered data, with no shape parameter or regularization. The sparse data used for RBF is the exact same set of (x, y) measurement points used by SPHINX in the sensitivity sweep, sampled from the reference CFD solution. This ensures that the comparison isolates the method (RBF interpolation vs. SPHINX physics-informed training) rather than introducing additional variation in the input data.

5.6.1 Headline comparison

At the SPHINX operating point $f = 0.1$, N = 200, SPHINX outperforms RBF on u-velocity error by factors ranging from 2.49 to 8.24 across cases (Table 7). The advantage is largest for NACA Re = 1,000, where

the airfoil pressure-gradient structure creates rapid spatial variations that RBF interpolation cannot recover from sparse data alone.

TABLE 7. SPHINX vs RBF u-velocity error comparison at $f = 0.1$, N = 200.

Method	SPHINX u_{err} (%)	RBF u_{err} (%)	RBF / SPHINX
Cylinder Re = 40	1.421	3.961	2.79×
NACA Re = 1,000	1.132	9.331	8.24×
NACA Re = 10,000	1.908	4.744	2.49×
FSI1 Re = 20	5.252	23.992	4.57×

5.6.2 Effect of Interface placement on RBF

A secondary finding is that RBF interpolation also benefits from interface-concentrated placement, though to a lesser extent and at a different optimum. Across the same sensitivity sweep of interface fractions, RBF's best u-velocity error occurs at $f = 0.3$ to 0.5 (depending on case), rather than at $f = 0.1$ where SPHINX peaks. The shift in optimum reflects a distinct mechanism: RBF benefits from concentration of points in regions of high field complexity (which happens to align with interface regions in our decomposition), while SPHINX benefits from a smaller interface allocation because of the dual-anchoring effect that uses each interface point to constrain two networks. This scientific signature, RBF preferring more interface concentration while SPHINX prefers less, supports the interpretation that SPHINX gains are mechanistically distinct from RBF gains and not simply a consequence of spatial point distribution.

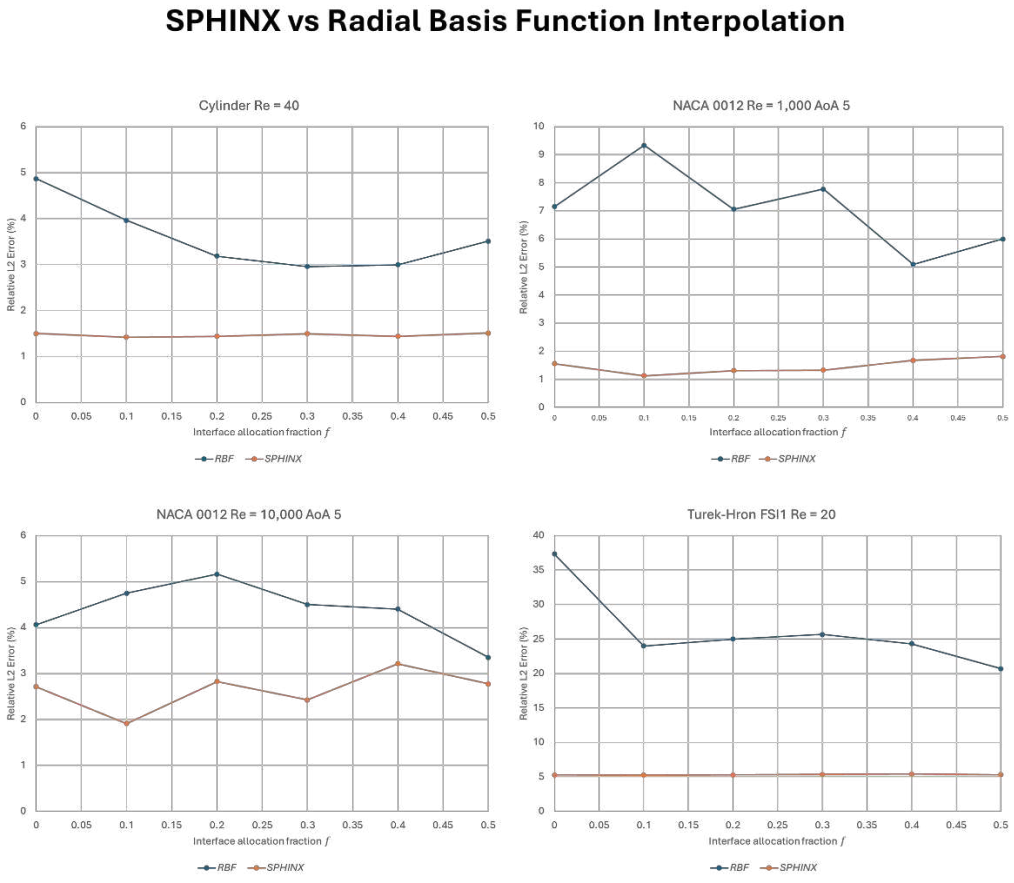


Figure 12 SPHINX vs RBF u-velocity curves vs interface fraction for all four cases

5.7. Ablation study: collocation strategies

In order to investigate whether SPHINX's performance can be further enhanced through different collocation strategies, we conducted an ablation study on the NACA 0012 Re = 1000 case at $N = 200$. For the ablation study we considered two alternative collocation methods and compared them against the standard uniform random sampling:

- **Adaptive collocation:** Every 5,000 epochs, the collocation points in each subdomain are resampled with probability proportional to the local PDE residual magnitude. This concentrates training effort in regions where the physics constraints are least satisfied.
- **Residual Adaptive Refinement (RAR):** Every 10,000 epochs, an additional 500 collocation points per subdomain are selected at locations with the highest PDE residual, augmenting the existing collocation set.

TABLE 8. Ablation study on collocation strategy (NACA 0012 Re = 1000, N = 200, 3 trials).

Method	u (%)	Effect
Standard collocation	1.4412 ± 0.0311	baseline within ablation run
+ Adaptive collocation	1.1311 ± 0.0278	-21.5% (helps)
+ RAR	1.7936 ± 0.0291	+24.5% (worse)

Table 8 shows a clear hierarchy among the different collocation strategies. When we compare adaptive collocation to the standard collocation strategy, we see a significant improvement, with the u-velocity error being reduced by 21.5% (1.4412% vs. 1.1311%). On the other hand, RAR does not improve upon the standard collocation baseline, likely due to bias introduced towards outlier location by a greedy selection of the highest-residual points.

The results show that SPHINX's interface-aware sparse data placement is complementary to adaptive collocation strategies as the combination achieves a 1% u-velocity error with only 200 sparse measurement points.

It is also important to note that the ablation experiments were conducted independently from the main scaling study, resulting in a slightly different baseline; thus, all comparisons within each table are internally consistent.

5.8. Cross-case comparison

Figure 13 shows a side-by-side comparison of SPHINX at $N = 200$ versus baseline XPINN across the different test cases.

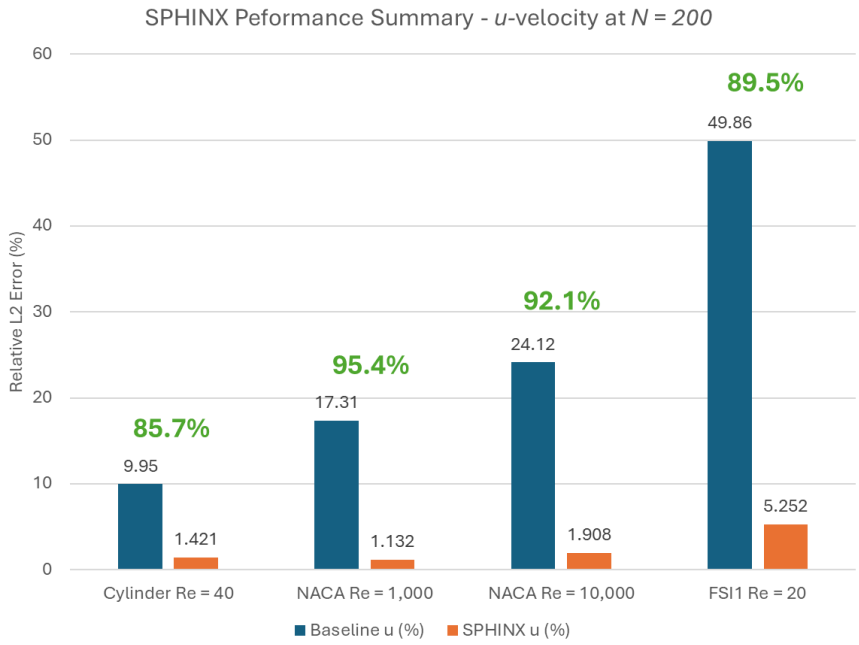


Figure 13 SPHINX Performance Summary

A summary of key results is also provided in table 9.

TABLE 9. Summary of SPHINX performance at N = 200 across all test cases.

Test case	u: base → SPHINX	u redn.	p: base → SPHINX	p redn.
Cylinder Re = 40	9.95 → 1.421	85.7%	58.04 → 5.944	89.8%
NACA Re = 1,000	24.40 → 1.132	95.4%	97.89 → 3.998	95.9%
NACA Re = 10,000	24.12 → 1.908	92.1%	196.48 → 22.246	88.7%
FSI1 Re = 20	49.86 → 5.252	89.5%	94.22 → 2.647	97%

There are several trends that are consistent across our four test cases. First of all, we see an 86% to 95% reduction in the *u*-velocity error, which is not influenced by different geometries and Reynolds number. Thus, we can infer that the data anchoring mechanism that SPHINX utilizes is robust and not sensitive to the specific flow configuration.

We also found that there were marginal benefits when the data points went beyond $N \approx 200$, which suggests that this is a practical optimum for the 8×128 XPINN networks with 10/90 interface-aware placement strategy.

Another important thing to note is the collapse of the variance across trials with the introduction of sparse data. The baseline XPINN had standard deviations ranging from 0.28% to as high as 74.41%, while SPHINX achieves sub-1% standard deviation in all cases at N = 200. The reduction in variance signifies that sparse data helps in not only improving the accuracy but also eliminates the sensitivity to random initialization that plagues purely physics-informed approaches.

Finally, when comparing the two NACA cases, we see the impact of Reynolds number on absolute accuracy. The relative improvement is nearly identical for the two cases (93% vs. 92%); however, the residual errors increase systematically: *u*-velocity from 1.132% to 1.908%, *v*-velocity from 8.528% to 14.024%, and pressure from 3.998% to 22.246%. The most severe degradation is that of the pressure error (4.3×) which

is consistent with the steeper pressure gradients at higher Reynolds number that exceed the network's representational capacity. This points us towards multi-scale architecture as a path of improvement as well as quantifying the spectral bias limitation.

5.9. Aerodynamic force coefficients

Field reconstruction accuracy in the L_2 norm does not by itself guarantee that derived integral quantities, such as drag and lift coefficients, are recovered with comparable fidelity. To assess the aerodynamic relevance of the SPHINX-reconstructed fields, we computed the drag and lift coefficients for all four benchmarks by integrating the full Cauchy stress (pressure plus viscous contributions) over the body surface using the SPHINX-predicted u, v, and p fields. Reference values were obtained by applying the same surface-integration procedure to the FEniCS CFD solution. For the NACA cases, body-axis forces were rotated into the wind-axis frame to recover the conventional drag and lift definitions.

TABLE 10. Aerodynamic force coefficients at N = 200: SPHINX vs. CFD reference. Differences are reported as absolute values ($\Delta C = C_{SPHINX} - C_{ref}$) to avoid ambiguity arising from near-zero reference values on symmetric configurations.

Test case	Cd ref	Cd SPHINX	ΔCd	Cl ref	Cl SPHINX	ΔCl
Cylinder Re = 40	1.561	1.590	+0.029	0.000	-0.003	-0.003
NACA Re = 1,000	0.152	0.123	-0.029	0.345	0.295	-0.050
NACA Re = 10,000	0.047	0.042	-0.005	0.404	0.267	-0.137
FSI1 Re = 20	1.195	1.090	-0.105	0.213	0.128	-0.085

Across all four cases, SPHINX recovers the drag coefficient to within $|\Delta Cd| \leq 0.11$ and the lift coefficient to within $|\Delta Cl| \leq 0.14$, using only 200 sparse measurement points and without any modification to the loss function for force prediction. The absolute drag accuracy is notably uniform across the four geometries, despite the wide range of Reynolds numbers and flow regimes considered. However, the relative significance of these errors differs across cases: for the cylinder at Re = 40, $\Delta Cd = 0.029$ corresponds to a 1.8% deviation from the reference value of 1.561, whereas for the NACA airfoil at Re = 1,000 the same absolute deviation represents an approximately 19% relative error because the reference Cd is roughly ten times smaller. This pattern reflects the fact that SPHINX errors scale with the magnitudes of the underlying flow fields rather than with the magnitudes of the derived integral quantities, and that small force coefficients are inherently harder targets for any field-reconstruction method.

Lift coefficients on near-symmetric configurations require careful interpretation. For the circular cylinder at Re = 40, the reference Cl is approximately zero by flow symmetry; the SPHINX prediction of −0.003 is also effectively zero, and the residual difference reflects numerical noise rather than aerodynamic asymmetry. The same logic applies to the FSI1 case, where the reference Cl of 0.213 arises from the modest geometric asymmetry of the cylinder–beam configuration in the channel and is itself a small quantity. For the NACA cases, by contrast, the reference Cl values of 0.345 and 0.404 are large enough that the absolute differences (0.050 and 0.137 respectively) carry physical significance. The Re = 10,000 case shows the largest ΔCl in the study, consistent with the spectral-bias argument developed in Section V.C: lift depends primarily on the chordwise asymmetry of the pressure distribution, which is set by the v-velocity field, and the v-velocity is the field component most affected by the limited representational capacity of the 8 × 128 architecture at high Reynolds number.

We emphasize that SPHINX was developed as a field-reconstruction method, not as a direct force-prediction method. The force-coefficient results reported here serve to confirm that the reconstructed fields are physically consistent and aerodynamically meaningful. Applications requiring high-accuracy direct force prediction, such as control-surface design, performance certification, or optimization in the loop,

would benefit from extensions including (i) higher network capacity or multi-scale architectures to better resolve near-wall gradients, (ii) explicit force-loss terms incorporated into the composite loss function, or (iii) increased data density at the body surface as part of the sparse-measurement budget. These directions are natural extensions of the framework presented here.

Force coefficients were computed using autograd-based surface stress integration with wind-axis rotation. The validation was performed at the $f = 0.3$ operating point of our initial parameter study; since the u-velocity surface error at the body changes by less than 0.2 percentage points between $f = 0.1$ and $f = 0.3$ at N = 200 (Tables 3 to 6), the reported coefficients are representative of the $f = 0.1$ results adopted as the operating point in the rest of the paper.

5.10. Computational cost

A practical concern for any data-augmented PINN method is the additional training overhead introduced by the data-loss term. Because SPHINX is purely a data-placement strategy and does not modify the network architecture or the structure of the composite loss function, the only added cost per training iteration is the evaluation of the network at the N sparse data points. We quantify this overhead directly using the Turek–Hron FSI1 benchmark, which is the most computationally demanding of the four cases due to the channel geometry and the cylinder–beam obstacle.

Training was performed on the SQUID supercomputer at Osaka University, using a single GPU node per trial. The baseline XPINN at FSI1 (no sparse data) required 41,287 s (11.5 hours) for three independent trials, while SPHINX with N = 200 required 44,680 s (12.4 hours) under identical hardware and training-schedule conditions. SPHINX therefore introduces approximately 8% computational overhead relative to the baseline, in exchange for an 89% reduction in u-velocity error, a 64% reduction in v-velocity error, and a 97% reduction in pressure error.

Two features of this overhead are worth emphasizing. First, the cost scales linearly with N rather than with the number of subdomains or training epochs, which means the cost-effectiveness ratio improves as N is reduced; the N = 30 cases reported in Sections V.A–V.C achieve substantial error reduction at less than 2% overhead. Second, the SPHINX overhead is independent of the underlying PINN architecture: the same 8% figure would apply to any framework that adds a data-loss term evaluated at N points per iteration, including Fourier-feature networks, multi-scale architectures, or APINN-style gating-network variants. The data-placement strategy is therefore compatible with future architectural improvements to the underlying PINN, and the 8% cost is an upper bound that reflects training at the full N = 200 budget.

5.11. Discussion and limitations

5.11.1 Several limitations of the present study merit discussion.

Allocation ratio is geometrically observed, not analytically derived:

The sensitivity study of Section 5.1 identifies $f = 0.1$ as the practical optimum across four benchmarks, but the value is observed empirically rather than predicted from first principles. The optimal ratio likely depends on the overlap width relative to the subdomain extent, the number of subdomains K, and the smoothness of the underlying flow. A theoretical analysis of f^* as a function of (K, overlap width, flow regularity) would convert SPHINX from a tested principle into a predictive one. We retain this as the primary open question.

5.11.2 Overlap-width sensitivity is untested:

All results use a fixed overlap width selected for each geometry. The relationship between overlap width and reconstruction accuracy, and in particular whether the optimal interface allocation f^*shifts with overlap width, is not characterized. Preliminary expectation is that wider overlaps tolerate lower f, but this is untested.

5.11.3 Number of subdomains is fixed at $K = 4$:

$K = 4$ matches the physical zoning of the geometries considered. Whether the SPHINX advantage scales with larger K (more interfaces to anchor) or degrades (individual subdomains too small for effective sampling) is an open question relevant to large-scale applications.

5.11.4 All cases lie in the laminar to laminar-transitional regime:

The four benchmarks span Re = 20 to 10,000 but exhibit either attached flow or mild, well-organized separation. Massively separated flows, post-stall airfoils, broadband-unsteady bluff-body wakes, and strong adverse-pressure-gradient detachment are not addressed. Extension to such regimes would require turbulence closure beyond the laminar Navier-Stokes equations used here, either through a RANS-PINN formulation in which modelled stresses augment the residual loss, or through a large-eddy approach with subgrid-scale modelling. The interface-aware placement principle itself should remain applicable.

5.11.5 FSI is treated as a fluid-only problem with prescribed structural motion:

For the FSI1 benchmark, the elastic beam displacement is prescribed from a validated turtleFSI reference, and SPHINX reconstructs only the fluid fields. This isolates the interface-aware placement effect on FSI-relevant geometries but does not exercise SPHINX on the coupled fluid-structure problem. The unsteady FSI2 (Re = 100) and FSI3 (Re = 200) benchmarks involve significant beam displacement and periodic vortex-induced vibrations, additionally requiring moving-boundary treatment and temporal encoding. Initial trials suggest that time-domain Fourier encoding alone is insufficient for convection-dominated periodic flows, motivating alternative formulations such as harmonic balance or POD-based reduced representations. These extensions are the subject of forthcoming work.

5.11.6 Sparse data is drawn from reference CFD solutions, not experimental measurements:

All sparse data points used in this study are sampled from FEniCS or SU2 reference solutions on the dense computational mesh. Practical applications would draw sparse data from physical measurements such as PIV, PTV, pressure taps, or 4D-flow MRI, which introduce measurement noise, spatially non-uniform sensor distributions, and geometric constraints on sensor placement that this study does not address. The interface regions identified by SPHINX as optimal may not be experimentally accessible in all applications, and the practical placement strategy in such settings would require a constrained-optimization formulation that respects both the SPHINX prior and the geometric admissibility of measurement locations. Validation under realistic noise levels and access constraints is an important direction for future work.

VI. CONCLUSIONS

We have introduced SPHINX, an interface-aware sparse data placement strategy for Extended Physics-Informed Neural Networks (XPINNs), and established interface-dominated reconstruction error as the central feature of XPINN sparse-data assimilation. We demonstrate its effectiveness across four benchmark problems spanning Reynolds numbers from 20 to 10,000 and geometries ranging from simple external flows to internal flows with fluid-structure interaction. The principal findings of this study are:

First, reconstruction error in baseline XPINNs is geometrically structured: it concentrates in the interface overlap regions defined by the decomposition, rather than distributing according to flow features such as boundary layers or wakes. This observation grounds the rest of the paper: it identifies where in the domain a sparse measurement budget should be spent.

Second, measurements placed in interface overlap regions are systematically more informative than uniformly distributed measurements. A sensitivity study across allocation fractions from 0 to 50 percent identifies $f = 0.1$ as the consistent u-velocity optimum across all four benchmarks. The effect persists across geometries and across the full Reynolds-number range tested, supporting the interpretation that this is a property of the decomposition geometry rather than of any particular flow configuration.

Third, SPHINX consistently reduces the relative L2 error in u-velocity by 86 to 95 percent across all test cases with only $N = 200$ sparse measurement points. The largest reductions occur where the baseline XPINN struggles most: the NACA 0012 airfoil at Re = 1,000 shows 95 percent reduction, while the FSI1 benchmark shows 89 percent reduction with a 97 percent pressure error reduction. The relative improvement is essentially Reynolds-number-independent across the range tested.

Fourth, sparse data regularizes the optimization process, eliminating the sensitivity to random initialization that plagues purely physics-informed approaches. The standard deviation across three independent trials collapses from a baseline range of 0.22 to 74.41 percent (XPINN) to below 0.04 percent for SPHINX at the operating point.

Fifth, comparison against radial basis function interpolation confirms that the gains stem from geometric placement of measurements coupled with the dual-anchoring mechanism, not from the additional measurements alone. Traditional reconstruction methods that ignore the decomposition structure underperform SPHINX by factors of 2 to 8 at the same measurement budget. RBF interpolation also benefits from interface-concentrated placement but at a different and higher optimum fraction, supporting the interpretation that SPHINX gains are mechanistically distinct.

Sixth, SPHINX interface-aware placement is complementary to other PINN enhancement strategies. An ablation study on collocation methods shows that adaptive collocation provides additional error reduction on top of SPHINX.

Seventh, the field reconstructions produced by SPHINX yield aerodynamically meaningful integral quantities. Drag coefficients are recovered to within reasonable accuracy across all four benchmarks. SPHINX adds approximately 8 percent computational overhead relative to a baseline XPINN, demonstrating that interface-aware data placement is a low-cost addition to the standard XPINN training procedure.

Taken together, these findings reframe sparse-data assimilation in domain-decomposed PINNs from a uniform sampling problem to a structurally guided one: the geometry of the decomposition itself prescribes where measurements are most informative. This is a principle, not a heuristic: the $f = 0.1$ allocation is the practical realization of a deeper observation about XPINN error structure that is independent of any particular flow configuration.

Several directions for future work are identified. The most significant is the extension to unsteady flows with periodic vortex shedding, as encountered in the FSI2 and FSI3 Turek-Hron benchmarks. Initial trials suggest that time-domain Fourier encoding alone is insufficient for convection-dominated periodic flows; alternative formulations including harmonic balance methods and POD-based reduced representations convert the unsteady problem into a set of coupled steady problems, which is precisely the class of problems where SPHINX has demonstrated strong performance. Additional directions include theoretical analysis of the optimal interface-to-random data ratio as a function of the number of subdomains and overlap width; evaluation of SPHINX robustness to noisy and non-uniformly distributed experimental data; extension to three-dimensional problems; and extension to turbulent flows through RANS-based PINN formulations. The simplicity and architecture-agnostic nature of the SPHINX data placement strategy suggest broad applicability across the physics-informed machine learning landscape.

Declarations

Funding

This study was supported by the JSPS KAKENHI (grant number 24K01072). The

present computations used the computational resources of the “SQUID” provided by

the D3 Center, Osaka University, through the HPCI System Research Project (Project

ID: hp250027).

Data and code availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of interest / Competing interests

The authors declare that they have no known competing financial or non-financial interests that could have appeared to influence the work reported in this paper.

Author contributions

H. El Halabi: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing - original draft, visualization. **Y. Takahashi**: Conceptualization, methodology, supervision, project administration, funding acquisition, writing - review and editing.

REFERENCES

1. Bungartz H-J, Schäfer M (2006) Fluid-Structure Interaction: Modelling, Simulation, Optimisation. Springer Berlin Heidelberg, Berlin, Heidelberg

2. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 378:686–707. https://doi.org/10.1016/j.jcp.2018.10.045

3. Jin X, Cai S, Li H, Karniadakis GE (2021) NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. Journal of Computational Physics 426:109951. https://doi.org/10.1016/j.jcp.2020.109951

4. Cai S, Wang Z, Wang S, et al (2021) Physics-Informed Neural Networks for Heat Transfer Problems. Journal of Heat Transfer 143:060801. https://doi.org/10.1115/1.4050542

5. Haghighat E, Raissi M, Moure A, et al (2021) A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. Computer Methods in Applied Mechanics and Engineering 379:113741. https://doi.org/10.1016/j.cma.2021.113741

6. Rahaman N, Baratin A, Arpit D, et al (2018) On the Spectral Bias of Neural Networks. https://doi.org/10.48550/ARXIV.1806.08734

7. Wang S, Teng Y, Perdikaris P (2021) Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks. SIAM J Sci Comput 43:A3055–A3081. https://doi.org/10.1137/20M1318043

8. Krishnapriyan AS, Gholami A, Zhe S, et al (2021) Characterizing possible failure modes in physics-informed neural networks. https://doi.org/10.48550/ARXIV.2109.01050

9. D. Jagtap A, Em Karniadakis G (2020) Extended Physics-Informed Neural Networks (XPINNs): A Generalized Space-Time Domain Decomposition Based Deep Learning Framework for Nonlinear Partial Differential Equations. CiCP 28:2002–2041. https://doi.org/10.4208/cicp.OA-2020-0164

10. Henkes A, Wessels H, Mahnken R (2021) Physics informed neural networks for continuum micromechanics. https://doi.org/10.48550/ARXIV.2110.07374

11. Farea A, Khan S, Daryani R, et al (2025) Learning Fluid-Structure Interaction with Physics-Informed Machine Learning and Immersed Boundary Methods

12. Wu C, Zhu M, Tan Q, et al (2022) A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. https://doi.org/10.48550/ARXIV.2207.10289

13. Lau GKR, Hemachandra A, Ng S-K, Low BKH (2024) PINNACLE: PINN Adaptive ColLocation and Experimental points selection

14. Wang H, Liu Y, Wang S (2022) Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network. Physics of Fluids 34:017116. https://doi.org/10.1063/5.0078143

15. Cai S, Mao Z, Wang Z, et al (2021) Physics-informed neural networks (PINNs) for fluid mechanics: A review

16. Moseley B, Markham A, Nissen-Meyer T (2021) Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. https://doi.org/10.48550/ARXIV.2107.07871

17. Mao Z, Jagtap AD, Karniadakis GE (2020) Physics-informed neural networks for high-speed flows. Computer Methods in Applied Mechanics and Engineering 360:112789. https://doi.org/10.1016/j.cma.2019.112789

18. Pu R, Feng X (2022) Physics-Informed Neural Networks for Solving Coupled Stokes–Darcy Equation. Entropy 24:1106. https://doi.org/10.3390/e24081106

19. Shin Y, Darbon J, Karniadakis GE (2020) On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. https://doi.org/10.48550/ARXIV.2004.01806

20. Jagtap AD, Kawaguchi K, Karniadakis GE (2020) Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. Journal of Computational Physics 404:109136. https://doi.org/10.1016/j.jcp.2019.109136

21. Lu L, Meng X, Mao Z, Karniadakis GE (2021) DeepXDE: A Deep Learning Library for Solving Differential Equations. SIAM Rev 63:208–228. https://doi.org/10.1137/19M1274067

22. Tancik M, Srinivasan PP, Mildenhall B, et al (2020) Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains

23. Jagtap AD, Kharazmi E, Karniadakis GE (2020) Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Computer Methods in Applied Mechanics and Engineering 365:113028. https://doi.org/10.1016/j.cma.2020.113028

24. Shukla K, Jagtap AD, Karniadakis GE (2021) Parallel Physics-Informed Neural Networks via Domain Decomposition

25. Hu Z, Jagtap AD, Karniadakis GE, Kawaguchi K (2023) Augmented Physics-Informed Neural Networks (APINNs): A gating network-based soft domain decomposition methodology. Engineering Applications of Artificial Intelligence 126:107183. https://doi.org/10.1016/j.engappai.2023.107183

26. Fang D, Tan J (2022) Immersed boundary-physics informed machine learning approach for fluid–solid coupling. Ocean Engineering 263:112360. https://doi.org/10.1016/j.oceaneng.2022.112360

27. Sundar R, Majumdar D, Lucor D, Sarkar S (2024) Physics-informed neural networks modelling for systems with moving immersed boundaries: Application to an unsteady flow past a plunging foil. Journal of Fluids and Structures 125:104066. https://doi.org/10.1016/j.jfluidstructs.2024.104066

28. Kingma DP, Ba J (2014) Adam: A Method for Stochastic Optimization

29. Bergersen A, Slyngstad A, Gjertsen S, et al (2020) turtleFSI: A Robust and Monolithic FEniCS-based Fluid-Structure Interaction Solver. JOSS 5:2089. https://doi.org/10.21105/joss.02089

30. Kurtulus DF (2015) On the Unsteady Behavior of the Flow around NACA 0012 Airfoil with Steady External Conditions at Re=1000. International Journal of Micro Air Vehicles 7:301–326. https://doi.org/10.1260/1756-8293.7.3.301

31. Kurtulus DF (2022) Critical Angle and Fundamental Frequency of Symmetric Airfoils at Low Reynolds Numbers. JAFM 15:. https://doi.org/10.47176/JAFM.15.03.33099

32. Winslow J, Otsuka H, Govindarajan B, Chopra I (2018) Basic Understanding of Airfoil Characteristics at Low Reynolds Numbers (104–105). Journal of Aircraft 55:1050–1061. https://doi.org/10.2514/1.C034415

33. Ohtake T, Nakae Y, Motohashi T (2007) Nonlinearity of the Aerodynamic Characteristics of NACA0012 Aerofoil at Low Reynolds Numbers. JOURNAL OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES 55:439–445. https://doi.org/10.2322/jjsass.55.439

34. Turek S, Hron J (2006) Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow. In: Bungartz H-J, Schäfer M (eds) Fluid-Structure Interaction. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 371–385

35. Buhmann MD (2003) Radial basis functions: theory and implementations. Cambridge University Press, Cambridge ; New York