RESEARCH ARTICLE | SEPTEMBER 10 2026

Physics-informed neural networks for two-way fluid–particle interaction: Data-free forward and motion-only inverse problems

Wentao Wang (王文韬) ; Jidong Zhao (赵吉东) ; Tongming Qu (瞿同明) ; Shiwei Zhao (赵仕威)

Physics of Fluids 38, 093328 (2026)
https://doi.org/10.1063/5.0341028

Physics of Fluids ARTICLE pubs.aip.org/aip/pof

Physics-informed neural networks for two-way fluid–particle interaction: Data-free forward and motion-only inverse problems

Cite as: Phys. Fluids **38**, 093328 (2026); doi: 10.1063/5.0341028
Submitted: 28 April 2026 · Accepted: 19 August 2026 ·
Published Online: 10 September 2026

Wentao Wang (王文韬),[1] Jidong Zhao (赵吉东),[1,a)] Tongming Qu (瞿同明),[1,2] and Shiwei Zhao (赵仕威)[1,2]

AFFILIATIONS

[1]Department of Civil and Environmental Engineering, Hong Kong University of Science and Technology, Hong Kong, China
[2]Institute of Water Engineering Sciences, Wuhan University, Wuhan, China

[a)]Author to whom correspondence should be addressed: jzhao@ust.hk

ABSTRACT

Free rigid-particle motion in a viscous fluid couples the incompressible Navier–Stokes and Newton–Euler equations through the traction on a moving interface; this same coupling underlies falling-body viscometry when the observed motion is used to infer fluid properties. We present a physics-informed neural-network framework with a single field-load-motion formulation comprising neural velocity and pressure fields on the moving space–time domain, surface-traction integration, and enforcement of the rigid-body momentum balance. In the data-free forward mode, the balance acts as the equation of motion: force and torque integrated from the network's own stress determine free translation and rotation through an under-relaxed fixed-point iteration, with no labeled flow or trajectory data. In the motion-only inverse mode, a single particle-centroid velocity record prescribes the moving-boundary kinematics, and its time derivative provides the acceleration that, together with the momentum balance, supplies an integral-force target. Following manufactured-solution verification, seven benchmarks against arbitrary Lagrangian–Eulerian reference solutions yield trajectory and velocity errors below 5% and flow-field errors within 9.0%; transfer learning reduces coupled wall-clock cost by a factor of 9.8. From one transient velocity record and without interior flow observations, the inverse mode identifies viscosity with 1.13% relative error and reconstructs the withheld fields to 6.8%–8.2% mean error, degrading gradually under up to 5% record noise. Accuracy is retained at $Re \approx 32$; at $Re \approx 118$, the formulation underpredicts instability-driven wake growth, delineating the present scope. The framework thus provides a label-free neural closure of the two-way fluid–particle coupling in forward mode and a controlled proof of concept for extending falling-body viscometry to joint recovery of viscosity and the unobserved flow.

I. INTRODUCTION

The free settling of a rigid particle in a viscous fluid is a canonical instance of two-way fluid–solid interaction (FSI). The fluid obeys the incompressible Navier–Stokes equations on a domain whose boundary moves with the particle; the particle obeys the Newton–Euler equations under gravity, buoyancy, and the hydrodynamic force and torque integrated from the fluid stress over its wetted surface; and the resulting translation and rotation relocate the interface and alter the flow, so neither subsystem can be advanced independently. This closed feedback underlies sedimentation, fluidization, and particle-laden transport. The same coupling is also the physical basis of falling-body viscometry, in which the observed motion of a body reveals a property of the surrounding fluid.

There has been a comprehensive computational literature attempting to resolve this coupling. Body-fitted arbitrary Lagrangian–Eulerian (ALE) formulations,[1,2] distributed-Lagrange-multiplier and fictitious-domain methods,[3] and immersed-boundary techniques[4,5] differ in how the moving interface is represented, but share the same closure: the resolved fluid stress is integrated into a resultant force and torque, the solid equations advance the particle, and the updated kinematics are imposed back on the fluid. These methods remain the reference for accurate forward simulation and supply the reference solutions used in this paper. Their strength, however, is tied to one mode of use: a fully specified problem, resolved once. Repeated queries, as in optimization or real-time prediction, each incur a complete re-solve with its mesh-motion or interface-tracking overhead, and when a material parameter is unknown and observations are sparse, the discretized equations offer no natural slot for the data, so assimilation and identification must wrap the solver in outer estimation loops. These two demands, economy under repetition, and a formulation that admits observations and unknowns directly have drawn two distinct neural responses.

The first response is data-driven: simulators that reproduce the coupled dynamics at a fraction of the online cost, drawing progressively more of the fluid-load-motion feedback loop into the model. Partitioned deep learning-based reduced-order models (DL-ROMs) synchronize interface and fluid dynamics learned in separate reduced spaces through interface representation and load recovery;[6] body-fitted graph models evolve coupled states and recover loads by integrating the Cauchy stress over predicted fields;[7] differentiable solvers embed discretized fluid, solid, and immersed-boundary operations in an end-to-end trainable architecture,[8] while formulations retaining explicit structural equations draw hydrodynamic loads from learned fields and wall stresses;[9] recent frameworks mirror ALE and partitioned procedures through fluid, solid, and interface surrogates with staged interfacial exchange.[10,11] These models capture genuinely two-way feedback and deliver the intended economy at inference. That economy is paid in advance: each model is calibrated against labeled solutions of precisely the solvers it is meant to relieve and generalizes reliably only near its training data. More consequentially, the calibrated object is a forward evolution operator: sparse observations and unknown physical parameters are not variables of the mapping, and the second demand remains unmet.

Physics-informed neural networks (PINNs)[12] are the complementary, equation-constrained response; they invert the roles of data and equations: the simulators above require labeled solutions and may embed the governing equations as structure, whereas a PINN requires the equations and may incorporate data where they exist. The unknown fields of one particular problem are represented by networks trained to minimize the residuals of the governing equations together with the initial and boundary conditions. Because the representation is continuous in space and time, no mesh, and for a moving boundary, no remeshing is required; because the solution is an optimization, sparse observations enter as loss terms and unknown parameters as trainable variables. Forward solution, data assimilation, and inverse identification thereby become instances of one formulation, distinguished only by which terms appear in the objective and which parameters are trainable, none requiring a labeled solution dataset. The paradigm is established across physics for forward solution,[13–38] surrogates,[39–42] field reconstruction,[43–45] and parameter estimation.[46] Its attraction here is therefore stated precisely: on a single fully specified solve, a PINN rarely outruns a mature mesh-based code (Sec. III reports this frankly); it is the unified, label-free forward–inverse formulation, not speed, that motivates its use.

Extending PINNs to FSI with moving boundaries has attracted sustained attention, though largely in the data-assimilation and inverse settings considered below. Forward solves remain few, and almost without exception the solid's motion is prescribed or externally supplied rather than predicted from the fluid load: oscillating, orbiting, and flapping bodies enter as Dirichlet velocity constraints on the space–time hypersurface traced by the interface,[47] and the moving Lagrangian interface of Ref. 48 is supplied without a structural equation closing its motion from the predicted load. Where the solid response is solved, it is distributed deformation rather than free rigid-body motion: an elastic vessel wall deflecting under pointwise stress continuity, with no resultant force-torque balance and no gross motion through the domain,[49] or an Euler–Bernoulli beam in steady potential flow,[50] which, although posed as forward, anchors its fluid network to interior pressure samples from which the interface force and beam acceleration are derived, data-assisted rather than equation-closed. To our knowledge, no PINN-based solver reported to date determines the trajectory of a freely moving rigid particle (translation and rotation) from hydrodynamic loads integrated from its own predicted stress field, at finite Reynolds number and without any labeled flow or trajectory data.

The data-assimilation and inverse literature divides by where the observations reside and what they recover. When a flow field is reconstructed, existing moving-solid formulations draw on data from the fluid interior, generally together with the solid kinematics;[47,50–57] when the data are confined to the solid, the recovered quantities remain at the body level, with no fluid parameter inferred and no flow field reconstructed.[52,53,58,59] The reciprocal question, whether the observed kinematics of the solid alone, without a single interior velocity or pressure datum, can determine a material property of the fluid together with its entire interior state, has, to our knowledge, not been posed in the PINN setting. At the scalar level, this is classical viscometry:[60–63] falling-ball and oscillating-body instruments infer viscosity from body motion, but return a bulk coefficient rather than the accompanying flow. Whether one transient velocity record, fused with the full nonlinear Navier–Stokes/Newton–Euler physics, carries enough information to determine both the viscosity and the unsteady interior flow is the second question this paper addresses.

Table I organizes the PINN-based FSI studies along these two axes: for forward problems, how the solid motion enters the solver and whether interior field data are additionally required; for inverse and data-assimilation problems, where the observations reside and what quantities are recovered.

As Table I shows, the forward gap separates prescribed or supplied motion and continuum deformation from free rigid-body motion closed, without field data, by hydrodynamic loads integrated from the solved flow. The inverse gap is orthogonal: flow-reconstruction studies draw on data sampled within the fluid, whereas body-only observations return structural quantities or load histories without identifying a fluid material property or reconstructing the surrounding flow. The present framework addresses both gaps: load-driven free translation and rotation in the forward problem, and joint recovery of viscosity and the unsteady flow field from body kinematics alone in the inverse problem.

This work closes the two gaps with a single framework operated in two modes, rather than two separate solvers. Neural networks represent the velocity and pressure fields of a given problem on its moving space–time domain, constrained by the Navier–Stokes residuals and the initial, boundary, and interface conditions. The particle's Newton–Euler balance closes the coupling, and its two readings define the two modes, mirroring the two readings of the physical coupling introduced at the outset. Read as a law of motion, the balance drives the data-free forward mode: an under-relaxed outer fixed-point iteration updates the hydrodynamic-load estimate from the network-predicted surface integrals, advances the particle by a velocity-Verlet update, and re-solves the fluid on the relocated boundary. An empirical drag estimate physically initializes the first iteration, with no reference field or trajectory data entering, and transfer learning (TL) across iterations (warm-starting each fluid solve from the last) makes the repeated moving-boundary training tractable. Read as a constraint on observed motion, the same balance defines the motion-constrained inverse mode. A particle-centroid velocity record serves

TABLE I. PINN-based solvers for fluid–solid interaction. Under "Forward/state solution," the columns distinguish prescribed or externally supplied motion, solved continuum deformation, and load-driven rigid-body motion, and record whether interior field samples enter the solution in addition to the governing equations and initial/boundary conditions. Under "Inverse/assimilation," "Data used" records the location and type of the observations and "Unknowns recovered" records the inferred quantities. "—" denotes that no result in the corresponding problem class is reported. Abbreviations: DOF, degrees of freedom; FVM, finite volume method; NS, Navier–Stokes equations; ODE, ordinary differential equation; PIV, particle image velocimetry; RANS, Reynolds-averaged Navier–Stokes equations; VIV, vortex-induced vibration; WIV, wake-induced vibration.

		Forward problem		Inverse/assimilation	
Study	Solid and flow	Solid motion in the solver	Field data	Data used	Unknowns recovered
Raissi *et al.*[52]	Elastically mounted rigid cylinder, 1-DOF crossflow VIV; 2D NS	—	—	(i) **Body motion + force records**: displacement + lift-force time series; (ii) **in-fluid + body motion**: velocity or passive-scalar data + displacement records (co-moving frame)	(i) Stiffness + damping, together with the continuous displacement/lift-force functions; **no fluid field**; (ii) velocity, pressure, lift and drag + structural parameters
Kharazmi *et al.*[58]	Flexible cylinder, distributed VIV (beam–string model)	—	—	**Body motion + force records**: scattered/six-sensor displacement and sectional hydrodynamic-force histories	Damping (and tension) + full structural displacement field from sparse sensors; **no flow field**
Cheng *et al.*[53]	Rigid cylinder(s), 2-DOF VIV and tandem WIV; 2D RANS	—	—	(i) **Body motion + force records**: limited displacement + lift/drag force samples; (ii) **in-fluid + body motion**: scattered velocity + displacement records	(i) Stiffness + damping, together with the continuous displacement/force functions; **no fluid field**; (ii) turbulent flow field, lift and drag + an eddy-viscosity closure
Tang *et al.*[54]	Rigid cylinder, 1-DOF crossflow VIV; 2D NS	—	—	**In-fluid + body motion**: scattered (u, v, p) + displacement labels; transfer learning cuts the labeled set to 1/2–1/8	Full (u, v, p) + the response, reconstructed and predicted forward in time
Davey *et al.*[59]	Settling rigid sphere, 1-D body ODE only; no flow-field model	—	—	**Body motion only**: displacement time series	Unsteady drag history (**no fluid field or fluid material property inferred**)
Calicchia *et al.*[55]	Swimming fish; 2D NS	—	—	**In-fluid + body motion**: PIV	Pressure

TABLE I. (*Continued.*)

Study	Solid and flow	Forward problem		Inverse/assimilation	
		Solid motion in the solver	Field data	Data used	Unknowns recovered
				velocity + measured body kinematics	
Sundar *et al.*[51]	Rigid plunging foil (moving IB); 2D NS	—	—	**In-fluid + body motion**: downsampled IBM velocity; prescribed foil kinematics	Pressure + reconstructed velocity
Zhu *et al.*[47]	Rigid moving bodies; 2D NS	**Prescribed** (analytical motions)	None	**In-fluid + body motion:** scattered velocity data; FVM-derived fitted trajectory	Pressure + reconstructed velocity
Zhang *et al.*[49]	Elastic vessel wall; axis-symmetric NS (ALE form)	**Solved: continuum wall deflection via pointwise stress continuity; no resultant force-torque balance, no gross motion**	None	—	—
Farea *et al.*[48]	Moving circular elastic solid; 2D NS (IB framework)	**Supplied**: moving Lagrangian interface from the IBM-generated configuration	None	—	—
Wang *et al.*[56]	Moving/deforming bodies; 2D and 3D NS	—	—	**In-fluid + body motion**: PIV velocity + measured boundary positions (a boundary network supplies their velocities by automatic differentiation)	Pressure + reconstructed velocity; fitted boundary position and AD-derived boundary velocity
Zhu *et al.*[57]	Hidden rigid boundaries; 2D NS	—	—	**In-fluid**: sparse flow measurements	Boundary shape and motion + velocity/pressure fields
Xia *et al.*[50]	Euler–Bernoulli beam; 2D potential flow	**Solved: continuum beam deflection via the Euler–Bernoulli equation with Robin fluid–structure coupling**	Required: fluid-domain pressure samples	**In-fluid + body motion**: fluid-domain pressure samples + - sparse beam-displacement data	Beam density (full beam deflection and fluid pressure field reconstructed en route)
This work	**Freely moving rigid particle(s), translation and rotation; 2D NS**	**Solved: free translation + rotation; Newton–Euler closed by force and torque integrated from the network's own stress**	**None**	**Body motion only: particle-centroid velocity record; no interior velocity or pressure datum**	**Dynamic viscosity + full (u, p)**

as the sole measurement, with no interior velocity or pressure observations, and is used twice. First, it prescribes the moving-boundary kinematics, dispensing with the outer iteration altogether. Second, differentiated and substituted into the momentum balance, it supplies the force history that the integrated network stress must reproduce. Relative to the forward objective, exactly two ingredients change: a force-consistency term is added and the viscosity becomes trainable. This constraint nevertheless plays the role of the missing interior probe: a single optimization jointly identifies the viscosity and reconstructs the never-observed flow, turning the falling-body viscometer of the opening paragraph into a method that now returns the flow field along with the coefficient.

The two-dimensional framework is assessed along a verification-validation-diagnostics sequence: hierarchical manufactured-solution verification; seven ALE-referenced benchmarks (settling, drafting, and inclined-particle configurations); an independently resampled physics-consistency audit; a neural-tangent-kernel analysis of transfer learning; and a wider-Reynolds-number stress test that delineates the limits of the present setting. Across the seven benchmarks, particle trajectory and velocity errors remain below 5% and velocity and pressure field errors do not exceed 9.0%; transfer learning across outer iterations reduces the wall-clock time of the coupled solve by approximately 9.8×, with further 1.18–1.34× gains when learned physics is reused across related cases. In the inverse mode, a single velocity record identifies the dynamic viscosity to within 1.13% and reconstructs the withheld velocity and pressure fields with mean errors of 6.8%–8.2%, degrading gradually up to 5% record noise.

The contributions are threefold. First, we establish a common field-load-motion PINN formulation in which the same moving-boundary flow representation, surface-traction integration, and rigid-body balance support two complementary closure modes: load-driven motion in the forward problem and motion-constrained fluid inference in the inverse problem. Second, in the data-free forward mode, we close the neural two-way coupling loop without labeled flow-field or particle-trajectory data: force and torque integrated from the PINN-predicted stress determine the free translation and rotation, while parameter transfer across outer iterations makes the repeated moving-boundary solves tractable. Third, in the motion-only inverse mode, we jointly identify the dynamic viscosity and reconstruct the unsteady velocity and pressure fields from a single transient particle-centroid velocity record, with no interior velocity or pressure observation; the record enters only through the moving boundary and the body momentum balance.

Based on a preliminary conference paper[64] introducing the iterative forward-coupling strategy, the present paper extends that forward-only study to the present forward-inverse formulation: it adds the motion-constrained inverse mode with its noise-robustness study, together with the 2D empirical load initialization, the hierarchical manufactured-solution verification, the independently resampled physics-consistency audit, the wider-regime assessment and the failure mode it identifies, and the neural-tangent-kernel analysis of cross-iteration parameter transfer.

The remainder of the paper is organized as follows. Section II presents the governing equations and the forward and inverse methodologies. Section III reports the manufactured-solution verification, forward benchmark validation, transfer-learning analyses, and inverse identification study. Section IV summarizes the principal findings and outlines future directions.

II. METHODOLOGY OF TL-ENHANCED PINN FOR TWO-WAY FSI

A. Brief overview of physics-informed neural networks

Physics-informed neural networks solve partial differential equations (PDEs) by minimizing their residuals during the training process of neural networks. In the PINNs framework, a PDE of interest is generally expressed as

$$\mathbf{u}_t(t,\mathbf{x}) + \mathcal{N}[\mathbf{u}] = 0, \quad t \in [0,T], \mathbf{x} \in \Omega, \tag{1}$$

subject to initial and boundary conditions,

$$\mathbf{u}(0,\mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \tag{2}$$

$$\mathcal{B}[\mathbf{u}] = 0, \quad t \in [0,T], \quad \mathbf{x} \in \partial\Omega, \tag{3}$$

where $\mathcal{N}$ represents a linear or nonlinear differential operator, and $\mathcal{B}$ denotes a boundary operator corresponding to Dirichlet, Neumann, Robin, or periodic conditions. Here, $\mathbf{u}(t, \mathbf{x})$ is the unknown latent solution that must satisfy the governing PDE system.

Following the seminal work by Raissi *et al.*,[46] we approximate the solution $\mathbf{u}(t, \mathbf{x})$ using a deep neural network $\mathbf{u}_{\boldsymbol{\theta}}(t, \mathbf{x})$, where $\boldsymbol{\theta}$ encapsulates all trainable parameters (e.g., weights and biases). The network is trained by minimizing a combined loss function that encodes the initial conditions, boundary conditions, and the governing PDE at selected collocation points. The loss function is formulated as

$$\mathcal{L}(\boldsymbol{\theta}) = \lambda_{\mathrm{IC}}\mathcal{L}_{\mathrm{IC}}(\boldsymbol{\theta}) + \lambda_{\mathrm{BC}}\mathcal{L}_{\mathrm{BC}}(\boldsymbol{\theta}) + \lambda_{\mathrm{r}}\mathcal{L}_{\mathrm{r}}(\boldsymbol{\theta}), \tag{4}$$

with

$$\mathcal{L}_{\mathrm{IC}}(\boldsymbol{\theta}) = \frac{1}{N_{\mathrm{IC}}}\sum_{j=1}^{N_{\mathrm{IC}}} \left\| \mathbf{u}_{\boldsymbol{\theta}}(0,\mathbf{x}_{\mathrm{IC},j}) - \mathbf{u}_0(\mathbf{x}_{\mathrm{IC},j}) \right\|_2^2, \tag{5}$$

$$\mathcal{L}_{\mathrm{BC}}(\boldsymbol{\theta}) = \frac{1}{N_{\mathrm{BC}}}\sum_{j=1}^{N_{\mathrm{BC}}} \left\| \mathcal{B}[\mathbf{u}_{\boldsymbol{\theta}}](t_{\mathrm{BC},j},\mathbf{x}_{\mathrm{BC},j}) \right\|_2^2, \tag{6}$$

$$\mathcal{L}_{\mathrm{r}}(\boldsymbol{\theta}) = \frac{1}{N_{\mathrm{r}}}\sum_{j=1}^{N_{\mathrm{r}}} \left\| \frac{\partial \mathbf{u}_{\boldsymbol{\theta}}}{\partial t}(t_{\mathrm{r},j},\mathbf{x}_{\mathrm{r},j}) + \mathcal{N}[\mathbf{u}_{\boldsymbol{\theta}}](t_{\mathrm{r},j},\mathbf{x}_{\mathrm{r},j}) \right\|_2^2, \tag{7}$$

where $\mathcal{L}_{\mathrm{IC}}(\boldsymbol{\theta})$ enforces the initial conditions, $\mathcal{L}_{\mathrm{BC}}(\boldsymbol{\theta})$ enforces the boundary conditions, and $\mathcal{L}_{\mathrm{r}}(\boldsymbol{\theta})$ is the residual loss that requires the network's outputs to satisfy the PDE at collocation points. The sets of points $\{\mathbf{x}_{IC,j}\}_{j=1}^{N_{IC}}$, $\{t_{BC,j}, \mathbf{x}_{BC,j}\}_{j=1}^{N_{BC}}$, and $\{t_{\mathrm{r},j}, \mathbf{x}_{\mathrm{r},j}\}_{j=1}^{N_r}$ can be chosen as vertices of a fixed mesh or generated via random sampling during each iteration of the gradient descent algorithm. A key advantage of PINNs is the use of automatic differentiation, which efficiently computes all necessary gradients analytically with respect to both input variables and network parameters $\boldsymbol{\theta}$. Note that the hyper-parameters $\{\lambda_{\mathrm{IC}}, \lambda_{\mathrm{BC}}, \text{and } \lambda_{\mathrm{r}}\}$ allow for balancing the contributions of each loss term during training and can be set manually or tuned automatically.[65,66]

B. Two-way fluid–solid interaction: Governing equations

This work focuses on two-dimensional, incompressible, isothermal flows interacting with rigid, non-deformable bodies. The

coupling between the fluid and the solid is bidirectional: the fluid exerts a drag force and moment on the solid, while the solid's motion enforces a no-slip boundary condition on the fluid.

1. Fluid dynamics

The fluid phase is governed by the incompressible Navier–Stokes equations as follows:

$$\nabla \cdot \mathbf{u}(t,\mathbf{x}) = 0 \quad t,\mathbf{x} \in [0,T] \times \Omega_\mathrm{f}, \tag{8}$$

$$\frac{\partial \rho_\mathrm{f} \mathbf{u}(t,\mathbf{x})}{\partial t} + \nabla \cdot (\rho_\mathrm{f} \mathbf{u}(t,\mathbf{x}) \otimes \mathbf{u}(t,\mathbf{x})) = -\nabla p + \mu \nabla^2 \mathbf{u}(t,\mathbf{x}) \quad t,\mathbf{x} \in [0,T] \times \Omega_\mathrm{f}, \tag{9}$$

where $\mathbf{u}$ is the fluid velocity, p is the dynamic pressure, ρ_f is the fluid density, and μ is the dynamic viscosity. The fluid domain is denoted by Ω_f and the simulation is carried out over the time interval T. The initial and boundary conditions for the fluid are defined as

$$\mathbf{u}(0,\mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega_\mathrm{f}, \tag{10}$$

$$\mathcal{B}_\mathrm{fixed}[\mathbf{u}(t,\mathbf{x}), p(t,\mathbf{x})] = 0, \quad t,\mathbf{x} \in [0,T] \times \partial\Omega_\mathrm{f}^\mathrm{fixed}, \tag{11}$$

$$\mathbf{u}(t,\mathbf{x}) = \mathbf{v}_b(t,\mathbf{x}), \quad t,\mathbf{x} \in [0,T] \times \partial\Omega_{\mathrm{s},b}(t), \tag{12}$$

where $\mathcal{B}_\mathrm{fixed}$ represents the boundary operator on fixed boundaries (e.g., slip, no-slip, inlet, and outlet), and Eq. (12) enforces the no-slip condition on the surface of solid b, with $\mathbf{v}_b$ denoting the rigid-body velocity.

2. Solid motion

The motion of body b is described by the two-dimensional planar Newton–Euler equations. The equations for linear and angular momentum are given as follows:

$$m_b \mathbf{a}_b(t) = \mathbf{F}_b(t) \quad t \in [0,T], \tag{13}$$

$$J_b \dot{\omega}_b(t) = M_b(t), \quad \dot{\phi}_b(t) = \dot{\omega}_b(t) \quad t \in [0,T]. \tag{14}$$

Because the motion is confined to the x–y plane, the angular velocity and the angular momentum both are aligned with the out-of-plane unit vector $\hat{\mathbf{z}}$. The gyroscopic term therefore vanishes identically, irrespective of the body geometry. Here, m_b and J_b denote, respectively, the mass and the centroidal polar moment of inertia per unit span; $\mathbf{a}_b$ is the acceleration of the body centroid; and ϕ_b, ω_b, and $\dot{\omega}_b$ are the rotation angle, angular velocity, and angular acceleration about the z-axis. The quantities $\mathbf{F}_b$ and M_b are the net external force and the net external moment about the body centroid, respectively. All force and moment quantities are understood per unit span.

The hydrodynamic resultant force and out-of-plane moment are evaluated by integrating the fluid traction over the instantaneous body boundary,

$$\mathbf{F}_{\mathrm{h},b}(t) = \int_{\partial\Omega_{\mathrm{s},b(t)}} \left[-p(t,\mathbf{x})\mathbf{I}_2 + \mu\left(\nabla\mathbf{u}(t,\mathbf{x}) + \nabla\mathbf{u}(t,\mathbf{x})^\mathrm{T}\right)\right] \cdot \mathbf{n} ds \quad t \in [0,T], \tag{15}$$

$$M_{\mathrm{h},b}(t) = \int_{\partial\Omega_{\mathrm{s},b}(t)} \boldsymbol{\xi}_b(t,\mathbf{x}) \times \left\{\left[-p(t,\mathbf{x})\mathbf{I}_2 + \mu\left(\nabla\mathbf{u}(t,\mathbf{x}) + \nabla\mathbf{u}(t,\mathbf{x})^\mathrm{T}\right)\right] \cdot \mathbf{n}\right\} \cdot \hat{\mathbf{z}}\, ds \quad t \in [0,T]. \tag{16}$$

Here, $\mathbf{n}$ is the unit normal directed from the body into the fluid, $\boldsymbol{\xi}_b(t,\mathbf{x}) = \mathbf{x} - \mathbf{X}_b(t)$ is the position vector relative to the body centroid, $r_b = \| \boldsymbol{\xi}_b \|_2$ is the corresponding radial distance, and $\mathbf{I}_2$ is the two-dimensional identity tensor. Since p denotes the dynamic pressure rather than the total pressure, the surface-stress integral in Eq. (15) yields the non-hydrostatic hydrodynamic force and therefore excludes buoyancy. The buoyancy contribution must consequently be accounted for separately in the external-force balance in Eq. (13).

To fully resolve the solid motion, the position and velocity can be obtained through time integration. The rigid-body velocity on the surface is

$$\mathbf{v}_b(t,\mathbf{x}) = \mathbf{V}_b(t) + \omega_b(t)\,\hat{\mathbf{z}} \times \boldsymbol{\xi}_b(t,\mathbf{x}) \quad t,\mathbf{x} \in [0,T] \times \partial\Omega_{\mathrm{s},b}(t), \tag{17}$$

where $\mathbf{X}_b$ and $\mathbf{V}_b$ are the centroid position and velocity. This velocity, together with Eq. (12), forms the no-slip boundary on moving boundaries, enabling the two-way fluid–solid coupling.

C. Transfer learning-enhanced coupling for PINNs-based fluids and moving solids

Our proposed framework leverages an iterative coupling strategy that is distinguished by two key features:

(1) **Empirical drag force initialization**: An established empirical drag force model provides a physics-informed initial guess for the drag forces.
(2) **Transfer learning–enhanced iterations**: The framework incorporates transfer learning by reusing the neural network parameters from previous iterations as a "warm start" for subsequent iterations, thus accelerating convergence and enhancing robustness.

In each iteration, the solid and fluid governing equations are solved alternately while updating the coupling terms and boundary conditions dynamically. Figure 1 schematically illustrates the overall strategy.

Step 1: Updating the hydrodynamic-load estimate

In the first iteration, an empirical drag-force correlation for a two-dimensional circular cylinder provides a physics-based initial estimate. Because all simulations are strictly two-dimensional, we use the quasi-steady drag force per unit span,

$$\mathbf{F}_{\mathrm{D},b}^{\mathrm{qs},(1)}(t) = -\frac{1}{2}\,\rho_\mathrm{f}\, C_\mathrm{D}(Re_\perp)\, D_\perp\, \|\mathbf{V}_{b\to\mathrm{f}}(t)\|_2\, \mathbf{V}_{b\to\mathrm{f}}(t), \tag{18}$$

$$C_\mathrm{D}(Re_\perp) = 1 + 10\, Re_\perp^{-2/3}, \tag{19}$$

$$Re_\perp = \frac{\rho_\mathrm{f}\, D_\perp\, \|\mathbf{V}_{b\to\mathrm{f}}\|_2}{\mu}, \tag{20}$$

where C_D is the two-dimensional circular-cylinder drag coefficient,[67] $Re_\perp$ is the initializer Reynolds number, ρ_f is the fluid density, and $\mathbf{V}_{b\to\mathrm{f}}$ is the body velocity relative to the fluid. Here, $D_\perp$ is the

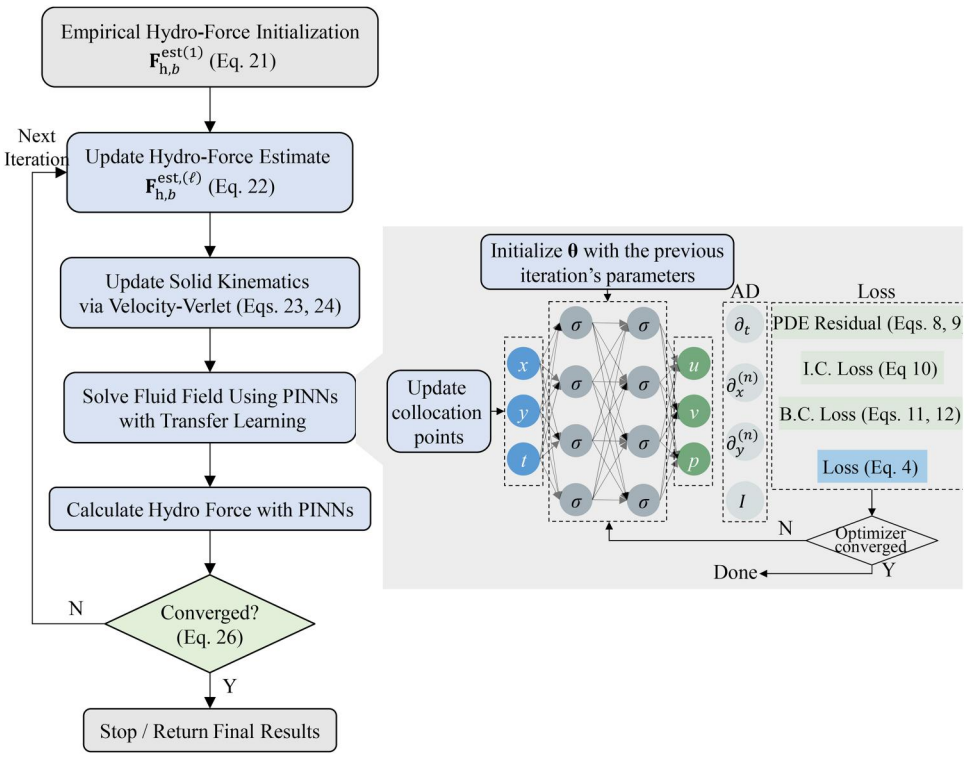

FIG. 1. Workflow of the transfer learning-enhanced iterative strategy.

cross-stream reference width supplied to the initializer; for a circular cylinder, $D_\perp = D_c = 2a$. The confined-cylinder initialization multiplies the quasi-steady part of Eq. (18) by an empirical linear wall-effect factor, $M_B = 1/(1 - 1.8\,BR)$, where $BR = D_\perp/W$ is the blockage ratio; this follows the two-dimensional bluff-body wall-effect correction form of Ota *et al.*[68] while retaining the coefficient as a case-initialization parameter rather than a physical wall model. The initialization also includes a potential-flow added-mass reaction $\mathbf{F}_{am,b} = -m_{add}\mathbf{a}_b^{est}$ with $m_{add} = C_a\rho_f\pi D_\perp^2/4$ and $C_a = 1$ per unit span. Equivalently, the explicit initialization solves $(m_b + m_{add})\mathbf{a}_b^{est} = m_b(1 - \rho_f/\rho_s)\mathbf{g} + M_B\mathbf{F}_{D,b}^{qs}$, and then uses

$$\mathbf{F}_{h,b}^{est,(1)} = M_B\mathbf{F}_{D,b}^{qs,(1)} + \mathbf{F}_{am,b}, \tag{21}$$

as the initial hydrodynamic-force history. In the very first iteration, before a fluid solution is available, the ambient fluid is assumed quiescent, $\boldsymbol{u}_f = 0$; hence, the relative velocity reduces to body velocity, $\boldsymbol{V}_{rel} = \boldsymbol{V}_b$. For the release-from-rest cases considered here, $\boldsymbol{V}_b(t = 0) = 0$.

In subsequent iterations, the hydrodynamic-force estimate is updated via a weighted combination of its previous estimate and the value calculated using PINNs from the previous iteration. This procedure resembles a relaxation method for solving nonlinear systems, where the hydrodynamic load acts as the unknown variable and the coupled Navier–Stokes and Newton–Euler equations constitute the nonlinear operator,

$$\mathbf{F}_{h,b}^{est,(\ell)}(t) = (1 - \zeta)\,\mathbf{F}_{h,b}^{est,(\ell-1)}(t) + \zeta\,\mathbf{F}_{h,b}^{PINN,(\ell-1)}(t). \tag{22}$$

Here, ζ serves as a relaxation factor to promote stability.

Step 2: Updating solid kinematics

The positions and velocities of the solid bodies are updated using the calculated forces and moments. To perform time integration, we employ the velocity-Verlet algorithm, which is favored in molecular dynamics and discrete element method because it balances accuracy, stability, and energy conservation with simple implementation. Although velocity-Verlet is second-order accurate for position-dependent forces, its direct application to the present hydrodynamic loading is formally first-order because the force depends on particle velocity (drag $\propto \|\mathbf{V}_{b\to f}\|_2\mathbf{V}_{b\to f}$) and acceleration (added mass).[69] In the coupled solver, this time integration is embedded within the outer load iteration. Equations (23) and (24) advance the solid with a fixed load history during each outer iteration. The overall temporal accuracy therefore depends on the uniform time step $\Delta t = T/N_{\Delta t}$, with $N_t = N_{\Delta t} + 1$ time nodes. Section III A 2 reports the measured temporal order; the time grids used in the physical studies are stated in Sec. III B,

$$\mathbf{X}_b(t + \Delta t) = \mathbf{X}_b(t) + \mathbf{V}_b(t)\,\Delta t + \frac{1}{2}\,\mathbf{a}_b(t)\,\Delta t^2, \tag{23}$$

$$\mathbf{V}_b(t + \Delta t) = \mathbf{V}_b(t) + \frac{\mathbf{a}_b(t) + \mathbf{a}_b(t + \Delta t)}{2}\,\Delta t. \tag{24}$$

Step 3: Solving the fluid field using transfer learning-enhanced PINNs

In this step, the fluid domain is adaptively sampled to optimize the distribution of collocation points. The update process involves:

- Re-sampling points on the moving boundary,

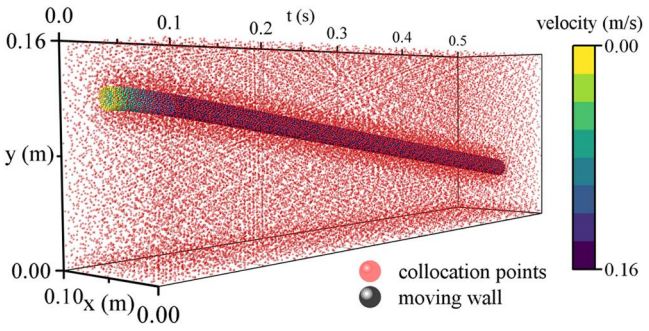

FIG. 2. Configuration of updated sampling points at an iteration.

- Increasing the density of points near the solid–fluid interface,
- Removing points that are located within the solid region.

We employ the Hammersley sampling method—a quasi-random sequence generation technique—to ensure an efficient and uniform distribution of collocation points.[70,71] Figure 2 illustrates the updated sampling points at an iteration. The sampled points on the moving boundary are annotated with their corresponding velocity information. It is apparent that a higher density of points is selected near the solid–fluid interface.

The fluid solution is approximated by the neural network outputs, where $\boldsymbol{\theta}$ represents the learnable parameters of the network,

$$\mathbf{u}(t,\mathbf{x}) \approx \mathbf{u}_{\boldsymbol{\theta}}(t,\mathbf{x}), \qquad p(t,\mathbf{x}) \approx p_{\boldsymbol{\theta}}(t,\mathbf{x}), \tag{25}$$

where the network parameters $\boldsymbol{\theta}$ before training are initialized randomly at the first fluid–solid iteration. During subsequent iterations, we leverage TL by initializing $\boldsymbol{\theta}$ with the previously trained parameters. Although the fluid field continues to evolve in iterations, their solutions remain close between two neighboring time steps. Thus, the use of transfer learning to inherit previously learned knowledge, together with minimal fine-tuning for the updated scenarios, can greatly accelerate the convergence of PINNs in coupling simulations of moving solids and fluids.

Step 4: Calculating the hydrodynamic loads with PINNs

After obtaining the fluid solution from PINNs, the hydrodynamic forces and moments acting on each solid are computed. Automatic differentiation is used to efficiently obtain the spatial gradients of the velocity and pressure fields at collocation points, enabling accurate evaluation of the stress acting on the solids. The hydrodynamic force and torque are then obtained by integrating the surface traction over the solid boundary with an end point-averaged trapezoidal panel rule (tractions averaged over the endpoints of each secant panel of the discretized surface), using a dedicated set of surface points represented through the analytic parametric form of the boundary. Section III A 2 confirms its second-order convergence with respect to the number of surface points. The verification and physical-study resolutions are reported with their respective configurations.

Step 5: Convergence check

Convergence is assessed by comparing the PINN-calculated (Step 4) and estimated (Step 1) hydrodynamic-load histories. This criterion is analogous to the relative change in the solution commonly used in the convergence analysis of relaxation methods,

$$\left[\frac{\sum_{n=0}^{N_t-1} \|\mathbf{F}_{\mathrm{h},b}^{\mathrm{PINN},(\ell)}(t^n) - \mathbf{F}_{\mathrm{h},b}^{\mathrm{est},(\ell)}(t^n)\|_2^2}{\sum_{n=0}^{N_t-1} \|\mathbf{F}_{\mathrm{h},b}^{\mathrm{est},(\ell)}(t^n)\|_2^2}\right]^{1/2} < \varepsilon_{\mathrm{out}}, \tag{26}$$

where $\varepsilon_{\mathrm{out}}$ is the prescribed outer tolerance. The values used in physical studies are stated in Sec. III B, and their sensitivity is examined in Appendix C. If the criteria are not met, return to Step 1 for the next iteration. This process continues until convergence, ensuring accurate capture of the two-way coupled fluid–solid dynamics.

D. Motion-constrained inverse formulation for joint viscosity identification and flow reconstruction

The inverse formulation reverses the information flow of the forward fluid–solid interaction problem: rather than predicting the solid motion and flow field through an outer fluid–solid iteration, it infers the constant viscosity and the latent fluid field consistent with a time-resolved solid-motion record. The observations are solid-centroid velocities $\left\{\left(t^n, \mathbf{v}_{b,\mathrm{obs}}^n\right)\right\}_{n=1}^{N_{obs}}$, where N_{obs} is the number of recorded time levels. The body geometry, solid and fluid densities, initial state, and fixed-domain boundary conditions are assumed known. The inverse unknowns are the network parameters $\boldsymbol{\theta}$ representing $(\mathbf{u}_{\boldsymbol{\theta}}, p_{\boldsymbol{\theta}})$ and the spatially and temporally constant viscosity μ. No interior fluid-velocity or pressure observations are supplied to the inverse optimization.

The motion record plays two roles in the inverse solve, but only one augments the forward PINN loss. First, it prescribes the moving-wall data already enforced through Eqs. (6), (12), and (17). Second, after temporal differentiation and application of the solid momentum balance, it supplies a motion-inferred target for an additional body-level force-consistency term. These two uses originate from the same velocity record and therefore do not constitute independent measurements.

Motion-prescribed moving boundary. The observed velocity sequence and the known initial centroid position determine the discrete solid motion used to position the known body geometry. The solid positions are propagated using the same discrete solid time-integration rule as in the forward formulation. The resulting motion is imposed through the no-slip condition in Eq. (12) on the corresponding moving boundary $\partial\Omega_{\mathrm{s},b}(t)$. No additional moving-boundary equation or separate moving-boundary loss is introduced in the inverse formulation. The pointwise velocity mismatch remains part of the boundary-condition loss $\mathcal{L}_{bc}$ defined in Eq. (6). The distinction from the forward problem lies in the source of its boundary data: in the forward solve, the solid motion is updated from the Newton–Euler equations during the outer fluid–solid iteration, whereas in the inverse solve it is prescribed by the processed motion record and remains fixed during optimization.

Motion-inferred integral-force constraint. The same velocity record is differentiated to obtain the solid acceleration required by the momentum balance. The acceleration is evaluated as

$$\mathbf{a}^n_{b,\mathrm{obs}} = \begin{cases} \dfrac{-3\mathbf{v}^1_{b,\mathrm{obs}} + 4\mathbf{v}^2_{b,\mathrm{obs}} - \mathbf{v}^3_{b,\mathrm{obs}}}{t^3 - t^1}, & n = 1, \\ \dfrac{-\mathbf{v}^{n-1}_{b,\mathrm{obs}} + \mathbf{v}^{n+1}_{b,\mathrm{obs}}}{t^{n+1} - t^{n-1}}, & 1 < n < N_{\mathrm{obs}}, \\ \dfrac{\mathbf{v}^{N_{\mathrm{obs}}-2}_{b,\mathrm{obs}} - 4\mathbf{v}^{N_{\mathrm{obs}}-1}_{b,\mathrm{obs}} + 3\mathbf{v}^{N_{\mathrm{obs}}}_{b,\mathrm{obs}}}{t^{N_{\mathrm{obs}}} - t^{N_{\mathrm{obs}}-2}}, & n = N_{\mathrm{obs}}, \end{cases} \tag{27}$$

which are the standard second-order forward, centered, and backward differences for uniformly sampled time sequence. Decomposing the net force in Eq. (13) as $\mathbf{F}_b = \mathbf{F}_{\mathrm{h},b} + \mathbf{F}_{\mathrm{known},b}$ yields the motion-inferred hydrodynamic force,

$$\mathbf{F}^{\mathrm{mot},n}_{\mathrm{h},b} = m_b \mathbf{a}^n_{b,\mathrm{obs}} - \mathbf{F}^n_{\mathrm{known},b}. \tag{28}$$

Here, the superscript "mot" denotes a force inferred from the solid-motion record rather than a directly measured force. $\mathbf{F}^n_{\mathrm{known},b}$ collects all prescribed or analytically known load contributions that are excluded from the surface-traction integral. The corresponding PINN force, $\mathbf{F}^{\mathrm{PINN}}_{\mathrm{h},b}$, is evaluated from the surface-stress integral in Eq. (15) over the motion-prescribed boundary $\partial\Omega_{\mathrm{s},b}(t)$. It is not obtained from the empirical drag-force initializer in Eq. (21), which is not used in the inverse solve. The additional force-consistency loss matches the motion-inferred and PINN-predicted surface-integrated forces,

$$\mathcal{L}_{\mathbf{F}_{\mathrm{h},b}}(\boldsymbol{\theta}, \mu) = \frac{1}{N_{\mathrm{obs}}} \sum_{n=1}^{N_{\mathrm{obs}}} \left\| \mathbf{F}^{\mathrm{PINN}}_{\mathrm{h},b}(t^n; \boldsymbol{\theta}, \mu) - \mathbf{F}^{\mathrm{mot},n}_{\mathrm{h},b} \right\|_2^2. \tag{29}$$

Equation (29) is therefore a motion-to-traction consistency condition at the body level. It compares the force inferred from the observed rigid-body acceleration with the force obtained by integrating the PINN-predicted surface stresses; it does not introduce an additional force-sensor measurement.

Joint inverse objective. Extending the notation of Eq. (4) to include the trainable viscosity, the inverse objective is

$$\mathcal{L}_{\mathrm{inv}}(\boldsymbol{\theta}, \mu) = \mathcal{L}(\boldsymbol{\theta}, \mu) + \lambda_{\mathbf{F}_{\mathrm{h}}} \mathcal{L}_{\mathbf{F}_{\mathrm{h}}}(\boldsymbol{\theta}, \mu). \tag{30}$$

Here, $\mathcal{L}(\boldsymbol{\theta}, \mu)$ retains the initial condition, boundary condition, and governing-equation residual terms defined in Eqs. (4)–(7). Relative to the forward formulation, the inverse objective augments Eq. (4) only with the force-consistency term $\mathcal{L}_{\mathbf{F}_h}$, while promoting the viscosity μ from a prescribed material parameter to a trainable scalar parameter. No interior velocity- or pressure-data mismatch terms are included in Eq. (30). The solid-velocity history is therefore the only time-dependent measurement stream entering the inverse calculation; all other inputs are prescribed physical and configuration information, rather than additional observations.

Identifiability rationale. Reconstructing an interior flow from moving-boundary kinematics alone is, in general, ill-posed. The present formulation avoids that form of the problem for two structural reasons: the unknown field collapses to a single scalar, and the observation record itself supplies the datum that selects this scalar.

The collapse follows from the forward structure of the constraints. Conditional on any admissible viscosity $\mu > 0$, the flow fields must satisfy a completely specified prescribed-motion forward problem, Eqs. (8)–(12) with the recorded motion entering through Eq. (17), whose velocity solution is unique at finite Reynolds number and finite time. The latent field is therefore not an independently adjustable unknown but the forward solution associated with μ, and field reconstruction reduces to identifying one scalar.

The governing equations, the initial state, and the boundary condition—including the record itself—cannot, however, determine this scalar: for every admissible μ, the corresponding forward solution satisfies them exactly. The viscosity is thus a flat direction of the physics loss. Equation (29) removes this degeneracy by using the record a second time. Differentiated and inserted into the solid momentum balance, Eqs. (27) and (28), the record yields the hydrodynamic force history that the predicted surface traction must reproduce. This force-consistency condition thereby plays the role of the missing interior probe.

In the inverse solve, every candidate viscosity is evaluated against the same recorded motion, so the question is whether two viscosities can produce the same force history at identical kinematics. A qualitative argument indicates that they cannot. At fixed kinematics, the quasi-steady drag force increases monotonically with μ across low-to-moderate Reynolds-number range ($\mathrm{Re} \propto \frac{1}{\mu}$ and $\frac{\mathrm{d}C_D}{\mathrm{dRe}} < 0$), so distinct viscosities produce distinct force histories. It is therefore reasonable to expect, although not proven here, that exactly one viscosity is consistent with the observed record; through the conditional forward problem, this also fixes the flow field. Section III F tests this expectation empirically.

The core notation is collected here, after all elements of the forward and inverse formulations have been introduced and before the results. Subscripts identify physical objects, components, parameterizations, or data roles; upright superscripts identify a solution or load source. The simulations are posed in the (x, y) plane, with rigid-body rotation about the z-axis (Table II).

III. RESULTS AND DISCUSSION

This section follows a verification-to-capability sequence. We first verify the numerical components and their two-way-coupled assembly with manufactured solutions. We then define the paired physical forward and inverse configurations and the distinct roles of their ALE references. The forward branch is validated first, with a circular-particle regime stress test delimiting its demonstrated scope. The next two sections examine iteration-level TL and reuse across related FSI cases. The section closes by testing empirical viscosity recovery and flow reconstruction in the motion-only inverse branch.

A. Manufactured-solution verification

Using the method of manufactured solutions (MMS), we verify the proposed framework progressively: first its numerical components, then an imposed-kinematics moving-boundary fluid problem (one-way coupling), and finally a complete two-way coupling problem. This hierarchy follows the verification architecture of Barbeau *et al.*[72]

1. Manufactured target and verification protocol

For a translating cylinder in two dimensions, the absence of a steady Stokes solution precludes the usual construction from a closed-form base flow. We instead use a dipole–rotlet stream function. Let $X = x - X_b(t)$, $Y = y - Y_b(t)$, and $r^2 = r_b^2 = X^2 + Y^2$ for coordinates relative to the circular particle center; the manufactured velocity and pressure fields are

TABLE II. Core symbols and index conventions for the 2D planar formulation.

Symbol	Meaning
$\mathbf{u} = (u, v)$, $\mathbf{u}_0$	Fluid velocity and prescribed initial velocity field
p	Dynamic pressure (with the hydrostatic body force absorbed); no separate total-pressure variable is used
ρ_f, ρ_s, μ, ν	Fluid and solid densities, dynamic viscosity, and kinematic viscosity $\nu = \mu/\rho_f$
Ω_f, $\Omega_{s,b}(t)$	Fluid domain and time-dependent domain of body b
$\mathbf{X}_b$, $\mathbf{V}_b$, $\mathbf{a}_b$	Centroid position, velocity, and acceleration of body b
ϕ_b, ω_b, $\dot{\omega}_b$	Rotation angle, angular velocity, and angular acceleration of body b
m_b, J_b, $\mathbf{I}_2$	Mass and planar moment of inertia (both per unit span), and two-dimensional identity tensor
$\boldsymbol{\xi}_b$, r_b	Centroid-relative position $\boldsymbol{\xi}_b = \mathbf{x} - \mathbf{X}_b$ and distance $r_b = \|\boldsymbol{\xi}_b\|_2$
$\mathbf{F}_b$, M_b	Net external force and moment on body b
$\mathbf{F}_{h,b}$, $M_{h,b}$	Full surface-stress hydrodynamic force and moment; only a force component along the relative-flow direction is called drag
$\mathbf{F}_{D,b}^{qs,(1)}$	Quasi-steady drag used only to initialize the first outer iteration
D_{eq}, $D_\perp$, D_c	Area-equivalent diameter, cross-stream reference width, and circular-cylinder diameter
$\boldsymbol{\theta}$, ζ, ε_{out}	Network parameters, load-relaxation factor, and outer-coupling tolerance
N_t, $N_{\Delta t} = N_t - 1$, N_{obs}	Number of time nodes, time steps, and observation records
N_{IC}, N_{BC}, N_r, N_q	Numbers of initial, boundary, residual-collocation, and surface-quadrature points
$b \in \{1, \ldots, N_b\}$	Rigid-body index; $N_b = 2$ for the two-particle drafting case and $N_b = 1$ otherwise
Subscripts c, j, q	Vector/load component, spatial or test point, and surface-quadrature point
Superscripts n, (ℓ)	Discrete-time or observation-record index and outer-iteration index
Upright labels	Descriptive/data labels are subscripts (e.g., IC, BC, and obs); solution or load sources are superscripts (e.g., PINN, ALE, est, mot, and MMS)

$$\psi^{\mathrm{MMS}} = \left[V_x(t)\, Y - V_y(t)\, X\right] \beta(r) - \omega_b(t)\, R^2 \ln \frac{r}{R}, \qquad \beta(r) = \frac{2R^2}{r^2} - \frac{R^4}{r^4}, \tag{31}$$

$$\mathbf{u}^{\mathrm{MMS}} = \left(\frac{\partial \psi^{\mathrm{MMS}}}{\partial y}, -\frac{\partial \psi^{\mathrm{MMS}}}{\partial x}\right), \qquad p^{\mathrm{MMS}} = p_0 + \gamma\mu\, \frac{V_x(t)\, X + V_y(t)\, Y}{r^2}, \tag{32}$$

where the subscript MMS marks the manufactured target; $\omega_b(t)$ denotes the angular velocity of the particle; and p_0 is an arbitrary gauge constant, removed before pressure errors are evaluated. The amplitude γ of the manufactured pressure is a free design parameter, any value yields an exact manufactured pair $(\mathbf{u}^{\mathrm{MMS}}, p^{\mathrm{MMS}})$, and is set to $\gamma = 4$ so that the pressure integral carries one-third of the closed-form hydrodynamic load,

$$\mathbf{F}_{h,b}^{\mathrm{MMS}} = \underbrace{-8\pi\mu\, \mathbf{V}_b}_{\text{viscous shear}} \underbrace{-\gamma\pi\mu\, \mathbf{V}_b}_{\text{pressure}} = -12\pi\mu\, \mathbf{V}_b, \qquad M_{h,b}^{\mathrm{MMS}} = -4\pi\mu R^2\, \omega_b. \tag{33}$$

Appendix A gives exactness, traction, and load-integration in full.

Mirroring the oscillating kinematics of Barbeau *et al.*,[72] the prescribed motion is a vertical oscillation without spin,

$$Y_b(t) = Y_0 + A \sin(\omega_{\mathrm{osc}} t), \qquad \mathbf{V}_b(t) = (0,\, A\omega_{\mathrm{osc}} \cos(\omega_{\mathrm{osc}} t)), \quad \omega_b \equiv 0, \tag{34}$$

so a single translational load channel is exercised. We use $A = 0.25R$ and $\omega_{\mathrm{osc}} = 2.0$ on the domain $[0, 6R]^2$ with the oscillation centered at $(3R,\ 3R)$, $T = 0.5$, and $R = \rho_f = \mu = 1$, giving $\mathrm{Re} = 2R\rho_f \max_t \|\mathbf{V}_b(t)\|_2/\mu = 2RA\omega_{\mathrm{osc}} = 1$. The momentum source is defined through the residual operator of Eq. (9), $\mathcal{R}_m(\mathbf{u}, p) = \rho_f[\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u}] + \nabla p - \mu\nabla^2\mathbf{u}$: the source $\mathbf{s}^{\mathrm{MMS}} = \mathcal{R}_m(\mathbf{u}^{\mathrm{MMS}}, p^{\mathrm{MMS}})$ is generated symbolically and supplied at the collocation points. The loss penalizes $\left[\mathcal{R}_m(\mathbf{u}_{\boldsymbol{\theta}}, p_{\boldsymbol{\theta}}) - \mathbf{s}^{\mathrm{MMS}}\right]/\rho_f$, which vanishes identically at the manufactured pair. Manufactured Dirichlet and initial data complete the exact target.

Increasing the number of PINN collocation points at fixed network architecture and training budget refines the space–time sampling, but need not produce a uniform power law in the total field error.[72,73] We therefore report measured orders for the isolated quadrature and time-integration tests, and fit an observed space–time collocation-refinement rate only where the field data exhibit a coherent descending branch.

The study spans a nine-level collocation ladder, $N_r = 10^2$ to 4×10^4 interior points, with the boundary and initial point counts scaled as $N_r^{2/3}$ so that N_r is the single resolution control; every level is repeated over three random seeds and reported as mean ± standard deviation. Field errors are evaluated on an independent, denser space–time test set ($\approx 8.4 \times 10^4$ points, excluding the particle interior); the pressure is compared after removal of the optimal constant (gauge) shift, $p_{\boldsymbol{\theta}}^{\circ} = p_{\boldsymbol{\theta}} - \langle p_{\boldsymbol{\theta}} - p^{\mathrm{MMS}} \rangle$. The space–time relative L_2 error for velocity and pressure are

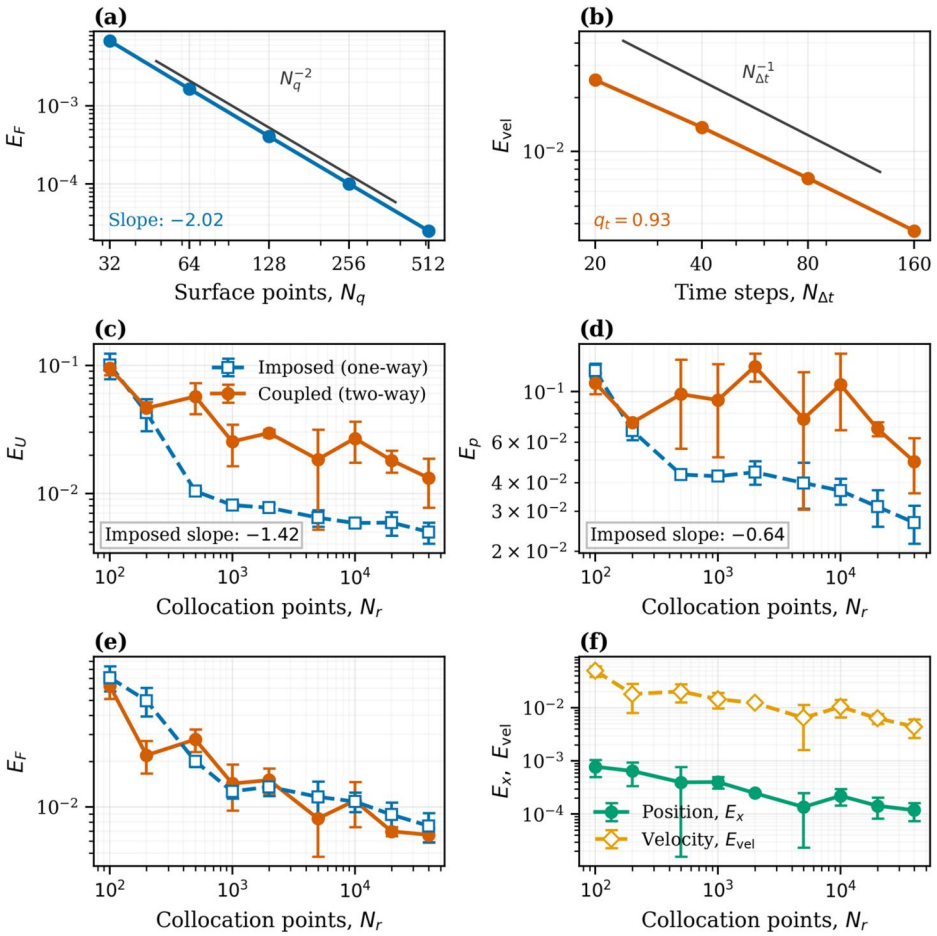

FIG. 3. Progressive manufactured-solution verification for the oscillating-cylinder dipole–rotlet case. (a) Exact-field force error E_F vs surface points N_q; (b) half-step solid-update error E_{vel} vs time steps $N_{\Delta t}$; (c) and (d) space–time E_U and gauge-pressure E_p for imposed (one-way) and coupled (two-way) dynamics; (e) drag error for both problems; and (f) coupled E_x and E_{vel}. Panels (c)–(f) show three-seed mean and standard deviation over $N_r = 10^2 - 4 \times 10^4$; the annotated imposed slopes fit $N_r = 100$–500.

$$E_U = \frac{\left(\sum_t \left\|\mathbf{u}_\theta - \mathbf{u}^{\mathrm{MMS}}\right\|_{L^2}^2\right)^{1/2}}{\left(\sum_t \left\|\mathbf{u}^{\mathrm{MMS}}\right\|_{L^2}^2\right)^{1/2}}, \qquad E_p = \frac{\left(\sum_t \left\|\mathring{p}_\theta - p^{\mathrm{MMS}}\right\|_{L^2}^2\right)^{1/2}}{\left(\sum_t \left\|p^{\mathrm{MMS}}\right\|_{L^2}^2\right)^{1/2}}. \tag{35}$$

Solid-response accuracy is measured by the relative L_2 errors of the y-position, y-velocity, and drag time series, denoted E_x, E_{vel}, and E_F. These definitions apply to each run; unless stated otherwise, all E_* values reported below are three-seed means, and the error bars in Fig. 3 denote one standard deviation.

2. Isolated-component verification

We first isolate the surface integration (Sec. II C, Step 4) and solid time integration (Step 2), both of which enter the coupled solver. For the exact fields, the end point-averaged trapezoidal panel rule exhibits second-order convergence with the number of surface points [slope −2.02; Fig. 3(a)]. Its relative load error is 6.71×10^{-4} at the benchmark setting $N_q = 100$ and 2.52×10^{-5} at the MMS setting $N_q = 512$. The latter is used throughout the assembled MMS tests, making the surface-integration error negligible relative to the field errors. For the isolated solid-update test, the implemented integrator is driven by the closed-form load $\mathbf{F}_b = -12\pi\mu\mathbf{V}_b + \mathbf{F}_{\mathrm{ext},b}$, for which the manufactured trajectory is exact. Evaluating the force at the half-step velocity gives successive temporal orders of 0.87, 0.94, and 0.97 over $N_{\Delta t} = 20$–160 [Fig. 3(b)]; thus $q_t \to 1$, as expected for velocity-Verlet with a velocity-dependent force.[69]

3. One-way coupling verification

We next prescribe the exact kinematics to define the imposed-kinematics problem (one-way coupling). Removing the load-to-motion update thereby isolates the moving-boundary fluid PINN, no-slip condition, and traction integration. With the kinematics prescribed, the error curves in Figs. 3(c) and 3(d) exhibit two clearly separated regimes. Below $N_r \approx 500$, the error is collocation-controlled: refining from $N_r = 100$ to 500 reduces E_U tenfold, from 10.1% to 1.0% (observed collocation-refinement rate −1.42 against N_r), and

E_p from 12.3% to 4.3% (rate −0.64), with the three seeds tightly clustered. Beyond $N_r \approx 500$, the changes become modest and non-monotone at fixed architecture and training budget, which means the network-approximation error and the optimization error dominate. We therefore fit the rates only over the initial descending interval; the later points do not form a second power-law regime. The drag error decreases from 7.1% to 0.75% across the ladder.

4. Two-way coupling verification

Finally, we restore two-way coupling. This is the assembly test of the proposed framework. The computed hydrodynamic load advances the solid, and the computed state relocates and updates the moving-boundary condition. To retain the same manufactured trajectory as the exact coupled solution, we add the external load $\mathbf{F}_{\mathrm{ext},b} = m_b\ddot{\mathbf{X}}_b - \mathbf{F}^{\mathrm{MMS}}_{\mathrm{h},b}$, with $m_b = \rho_s \pi R^2$ and $\rho_s/\rho_f = 2.5$. The production relaxation parameters are $\zeta = 0.1$ and $\varepsilon_{\mathrm{out}} = 0.005$. To test robustness to the initial guess, the first drag estimate is set to zero rather than taken from the empirical initialization. All 27 runs (nine resolutions and three seeds) converge from this uninformative guess to the prescribed tolerance within 35–44 outer iterations. The recovered trajectory remains within 0.4% of R for $N_r \geq 10^3$ and within 0.94% even at $N_r = 100$. At the finest level, $E_x = 1.2 \times 10^{-4}$, $E_{\mathrm{vel}} = 0.44\%$, and $E_F = 0.66\%$ [Fig. 3(e)].

Recovery of the exact trajectory in every coupled run verifies the consistent assembly of the fluid field, hydrodynamic load, solid update, and moving boundary. Figures 3(c) and 3(d) then shows how replacing prescribed kinematics with the computed motion affects field accuracy. Below $N_r = 500$, the two problems are nearly indistinguishable in velocity and pressure error. Between $N_r = 200$ and 500, the imposed curve continues its collocation-controlled decrease, whereas the coupled curve departs earlier from this trend and enters a higher, non-monotone band with a visibly larger three-seed spread. At $N_r = 4 \times 10^4$, the coupled errors are $E_U = 1.3\%$ and $E_p = 4.9\%$, compared with 0.50% and 2.7% under imposed dynamics.

This departure is consistent with the structural difference between the two problems. Under imposed dynamics, a load discrepancy remains an output; under coupled dynamics, it perturbs the solid state, hence the moving-boundary condition and subsequent fluid and load solutions. The mean net-drag errors nevertheless remain comparable over the six levels with $N_r \geq 10^3$ (coupled/imposed ratio 0.72–1.13), so the field departure is not accompanied by a systematic increase in drag error. The resulting field errors fluctuate instead of forming a power-law regime, and no independent space–time collocation-refinement rate is inferred. A tolerance-continuation test excludes premature outer termination as the primary cause of this departure. Tightening the coupling tolerance tenfold to $\varepsilon_{\mathrm{out}} = 5 \times 10^{-4}$ changes the coupled field errors by less than 20%, produces little further change below $\varepsilon_{\mathrm{out}} = 10^{-3}$, and does not bring them back to the imposed reference.

Together, these tests establish the quadrature and time-integration orders, verify the moving-boundary field and load evaluation under prescribed motion, and recover the manufactured trajectory with two-way coupling.

B. Physical benchmark setups and reference solutions

Figure 4 organizes the physical study around a common gravity-driven moving-boundary setting and two opposite information flows. In the forward benchmarks, the material properties are prescribed and the transient flow and rigid-body motion are predicted jointly, without ALE flow-field supervision or prescribed trajectory data. In the controlled inverse benchmark, one time-resolved particle-centroid vertical-velocity record supplies both the moving-wall kinematics and, through the momentum balance, the integral-force target, while the viscosity and latent flow are inferred without interior velocity or pressure observations. The comparison is therefore at the level of information flow.

Across these physical configurations, no-slip conditions are imposed on the lateral and bottom boundaries, as well as at the solid–fluid interface. A free-slip condition is applied at the top boundary (i.e., vertical velocity $v = 0$, and $\partial u/\partial y = 0$). Because free-surface deformation and gas-phase dynamics are not resolved in the present model, the upper boundary is treated as a flat, shear-free rigid lid, consistent with the intended single-phase, non-deforming-surface approximation.[74] The confinement level is quantified by the blockage ratio ($D_\perp/W$), which ranges from 0.15 to 0.335 across the primary benchmarks (Table III).

All dimensional quantities are reported in SI units. The baseline forward benchmarks use $\mu = 1 \times 10^{-3}$ Pa s, $\rho_f = 1\,\mathrm{kg/m^3}$, and

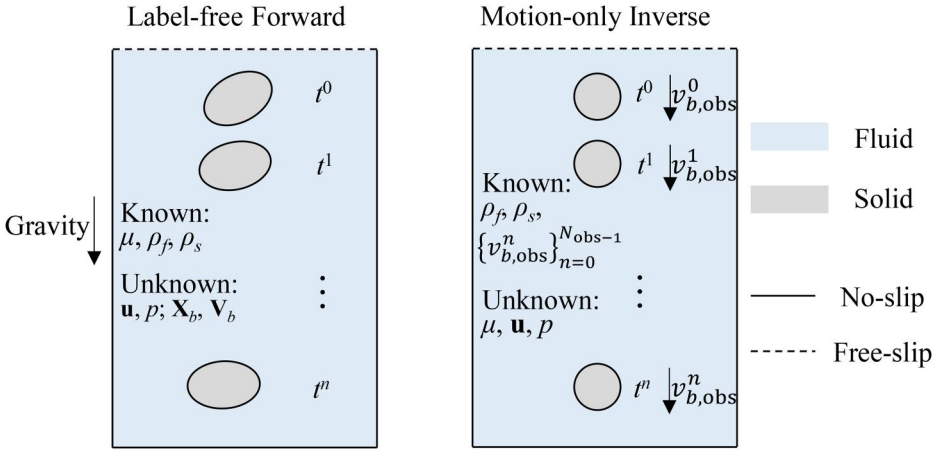

FIG. 4. Physical configurations and reversed information flow. The forward branch (left) predicts fluid and rigid-body motion from prescribed material properties. The motion-constrained inverse branch (right) uses the observed vertical particle-centroid velocity $v_{b,\mathrm{obs}}$ to infer μ, $\mathbf{u}$, and p without interior flow observations. Both are gravity-driven and share the wall-condition pattern; Table III lists the case-specific parameters.

TABLE III. Nondimensional parameters of the physical benchmark.

Case	Re	ρ_s/ρ_f	Ga	Blockage D/W
Single circular particle	2.8	2.5	7.0	0.15
Ellipses (AR 1.0–0.2)	2.9–3.2	2.5	7.0	0.168–0.335
Drafting (two particles)	3.4	2.5	7.0	0.15
Inclined ellipse (30°)	3.1	2.5	7.0	0.194
Wider regime	31.8–117.9	1.167	38.9–112.8	0.15
Inverse problem	2.0	1.155	5.9	0.15

$\rho_s = 2.5\,\mathrm{kg/m^3}$, giving $\rho_s/\rho_f = 2.5$ and Galileo number $Ga = 7.0$ at the area-equivalent particle diameter $D_{\mathrm{eq}} = 15\,\mathrm{mm}$ particle scale (Table III). For the incompressible Newtonian fluid–rigid-body problem considered here, at fixed geometry, gravity, and initial and boundary conditions, the common scaling $(\rho_f \rho_s \mu) \to c(\rho_f \rho_s \mu)$ leaves both ρ_s/ρ_f and Ga unchanged. Under the same scaling, the dynamic pressure and hydrodynamic force and moment scale by c, as do the rigid-body inertia, gravity, and buoyancy terms. The rescaled continuum problem therefore admits the same fluid velocity and rigid-body kinematics. The wider-regime tests in Sec. III C 6 uses $\mu = 0.373\,\mathrm{Pa\,s}$, $\rho_f = 970\,\mathrm{kg/m^3}$, and $\rho_s = 1120\,\mathrm{kg/m^3}$, while the motion-only inverse case in Sec. III F uses $\mu = 0.373\,\mathrm{Pa\,s}$, $\rho_f = 970\,\mathrm{kg/m^3}$, and $\rho_s = 1120\,\mathrm{kg/m^3}$.

The relevant nondimensional parameters for the physical cases are summarized in Table III. The Reynolds number is defined as $Re = \rho_f D_{\mathrm{eq}} \max_t \| \mathbf{V}_b^{ALE}(t) \|_2 / \mu$, where D_{eq} is the area-equivalent particle diameter and $\max_t \| \mathbf{V}_b^{ALE}(t) \|_2$ is the maximum particle-centroid settling speed of the corresponding ALE reference solution. The Galileo number is defined as $Ga = \left(\left| 1 - \rho_s/\rho_f \right| g D_{\mathrm{eq}}^3 \right)^{0.5} / \nu$, where $\nu = \mu/\rho_f$. The confinement, or blockage, ratio is defined as is $D_\perp / W$, where $D_\perp$ is the particle width normal to the principal settling direction and W is the domain width. For circular particles, $D_c = D_{\mathrm{eq}} = D_\perp$.

Reference solutions were obtained using COMSOL Multiphysics with the Arbitrary Lagrangian–Eulerian (ALE) moving mesh method.[74] The ALE solutions play different evidentiary roles in the two branches. For the forward benchmarks, they are used only for post-solve validation; no ALE flow field or particle trajectory supervises the PINN solve. For the inverse benchmark, ALE supplies the synthetic particle-centroid velocity record, while its velocity and pressure fields are withheld until post-training evaluation. For every physical-case field error reported below, each evaluated snapshot is treated independently. We triangulate the retained irregular COMSOL points by Delaunay reconstruction and accumulate one-third of each retained triangle's area at each vertex to obtain lumped weights w_j. For $f \in \{u, v\}$, the snapshot error is $\left[\sum_j w_j \left(f_{\vartheta,j} - f_j^{\mathrm{ALE}} \right)^2 / \sum_j w_j \left(f_j^{\mathrm{ALE}} \right)^2 \right]^{1/2}$. Before applying the same norm to pressure, the predicted and ALE pressures are each centered by their own weighted spatial mean at that snapshot. The reported run value is the unweighted arithmetic mean of the snapshot relative errors. The forward physical studies use $T = 0.5\,\mathrm{s}$ with $N_t = 101$ time nodes ($N_{\Delta t} = 100$ and $\Delta t \approx 5.0 \times 10^{-3}\,\mathrm{s}$) and $N_q = 100$ surface points per time slice. Their coupling controls are $\zeta = 0.1$ and $\varepsilon_{\mathrm{out}} = 0.005$. The motion-constrained inverse study uses $T = 1.0\,\mathrm{s}$, $N_{\mathrm{obs}} = 101$, and $N_q = 100$, without an outer load iteration. Appendix B gives the physical-study network, loss, sampling, and optimizer configurations; Appendix C reports the tested sensitivities.

C. Forward benchmark validation

We consider the following benchmark cases, chosen to represent a range of typical FSI scenarios involving different particle shapes and dynamics:

1. Settling of a single circular particle

This fundamental case involves a single circular particle (diameter 15 mm) settling under gravity in quiescent fluid within a domain measuring 0.10 m by 0.16 m. Initially, the particle is positioned 0.12 m above the bottom wall before being released to drop. Figure 5 presents the training loss of the TL-enhanced iterative strategy. The run comprises one initial solve followed by 18 warm-started iterations (19 fluid solves in total) and satisfies Eq. (26) ($\varepsilon_{\mathrm{out}} = 0.005$) in approximately 129 min. In the first iteration, the model is trained using the Adam optimizer with a learning rate of 0.001 for the initial 10 000 epochs. This approach helps prevent convergence to a local minimum due to the random initialization of the neural network. Afterward, the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) optimizer is employed

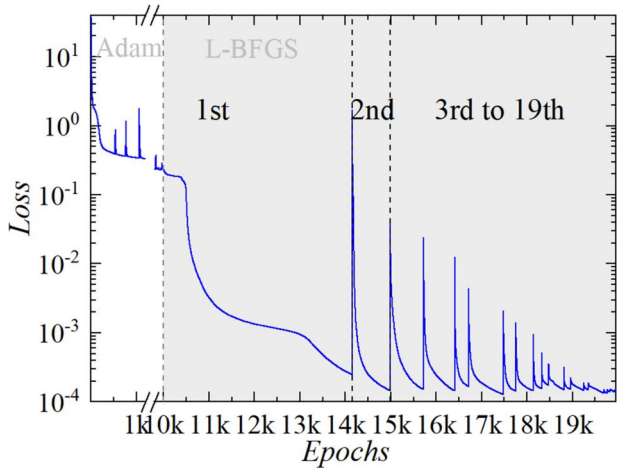

FIG. 5. Training loss curve for the TL–enhanced iterative strategy.

to fine-tune the model. Starting from the second iteration, and in subsequent iterations, only the L-BFGS optimizer is used for training, as the neural network is initialized using the results from the previous iteration and the low relaxation factor ($\zeta = 0.1$) for drag force updates ensures that the solution between consecutive iterations remains close. Subsection 2 of Appendix C shows that this strong under-relaxation is necessary rather than incidental: with every other control unchanged, the same benchmark is terminated as divergent for $\zeta = 0.3$ and above.

Figure 6 demonstrates close quantitative agreement (relative-L^2 errors of $< 1\%$ in position, velocity, and drag; Table IV) between the PINNs-based framework (denoted as ML or machine learning) and the COMSOL (ALE) reference for particle position, velocity, and drag force evolution. Figure 7 further illustrates a comparison of the spatial velocity and pressure fields at selected time. Table IV gives snapshot-mean relative-L^2 errors of 2.1%, 1.6%, and 2.0% in u, v, and p, respectively. Figure 8 illustrates the convergence of the drag force

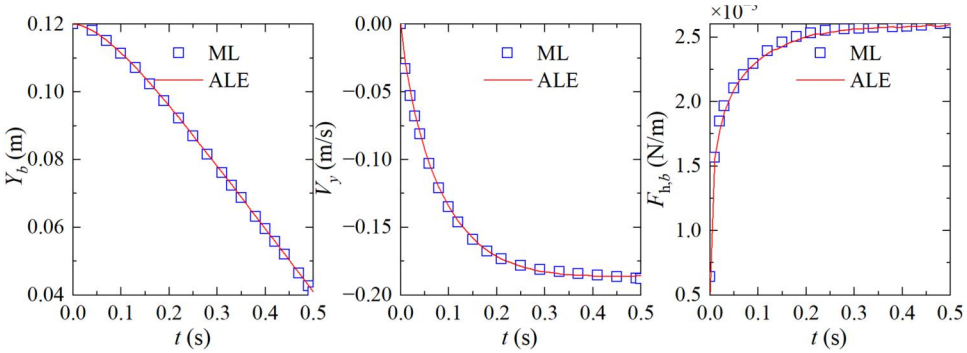

FIG. 6. Comparison of particle settling dynamics over time between the PINNs-based framework (ML) and the COMSOL (ALE) reference solution. Domain: $0.10 \times 0.16\ \mathrm{m}^2$; single circular particle of diameter 0.015 m released from rest; $Ga = 7.0$, and $Re = 2.8$.

TABLE IV. Relative L^2 errors (%) of the PINN predictions with respect to the ALE reference solutions for the forward benchmarks.

Case	Particle trajectory	Particle velocity	Hydrodynamic load	Flow field $u/v/p$
Single circular particle	0.23	0.55	0.88	2.1/1.6/2.0
Ellipses (AR = 0.8)	0.32	0.75	0.87	2.9/1.9/2.3
Ellipses (AR = 0.6)	0.37	0.87	2.47	3.1/2.0/2.2
Ellipses (AR = 0.4)	0.55	1.85	1.51	4.8/3.8/4.5
Ellipses (AR = 0.2)	0.79	4.26	2.62	7.9/8.7/9.0
Drafting (two particles)	Leading: 0.59; trailing: 0.50	Leading: 1.13; trailing: 1.06	Leading: 0.81; trailing: 1.10	5.1/4.8/3.6
Inclined ellipse (30°)	$x/y/\phi_b$: 0.24/0.42/0.41	$v_x/v_y/\omega$: 2.76/0.90/2.32	$F_x/F_y/M_z$: 7.47/2.19/3.03	3.5/2.5/2.3

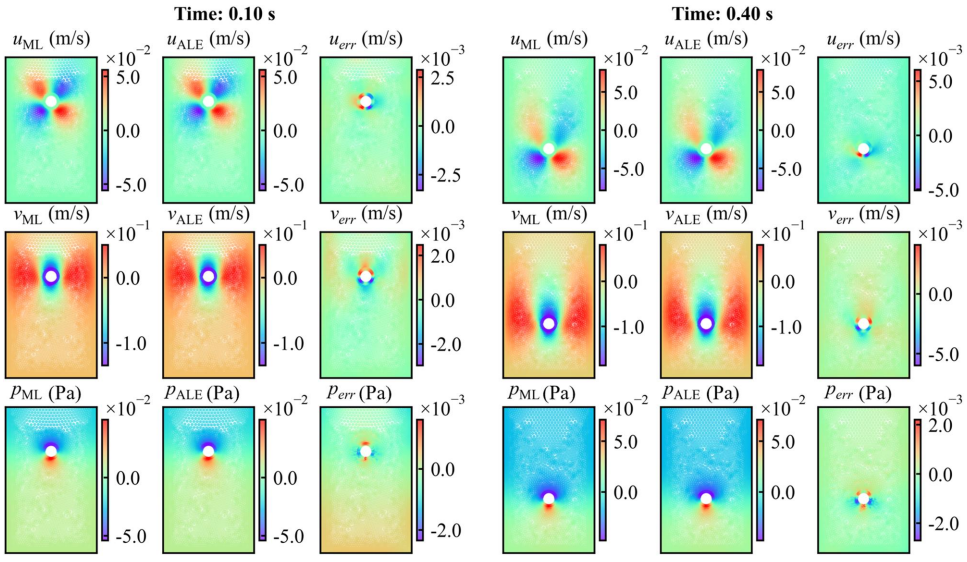

FIG. 7. Comparative snapshots of velocity components and pressure fields over time between the PINNs-based framework (denoted as ML) and the COMSOL (ALE reference). Domain: $0.10 \times 0.16\ \mathrm{m}^2$; single circular particle of diameter 0.015 m released from rest; $Ga = 7.0$, $Re = 2.8$; snapshots at t = 0.1 s and t = 0.4 s.

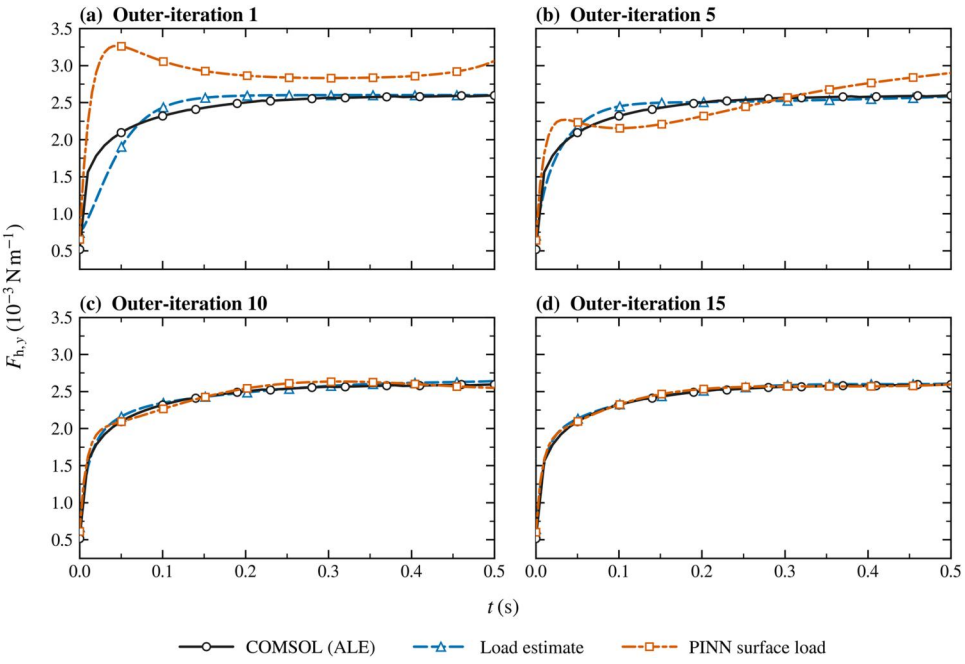

FIG. 8. Drag force update over selected outer iterations for the single-circular-particle study. Panels (a)–(d) show outer iterations 1, 5, 10, and 15, respectively. "Load estimate" denotes the force history entering that iteration, and "PINN surface load" denotes the traction-integrated result.

throughout the iterative process. The initial drag force estimate, derived from the empirical correlation [Eq. (21)], differs from the two-dimensional COMSOL reference. At the first iteration, a discrepancy emerges between the initial estimate and the drag force calculated by PINNs, with the COMSOL reference typically residing between these two approximations. However, with progressive iterations, the drag force estimates and the drag force generated by PINNs gradually align and converge toward the COMSOL reference solutions.

2. Settling of elliptical particles with varying aspect ratios

To test the proposed PINNs-based solver's capability in accounting for particle shape, we simulate the settling of elliptical particles. The shape is characterized by the aspect ratio $AR = b/a$, where a and b are the semi-major axes along the x and y directions, respectively. Five cases with AR values of 1.0 (circular), 0.8, 0.6, 0.4, and 0.2 are considered, all having an equivalent diameter of 15 mm. Figure 9 depicts the evolution of the settling velocity for each particle shape. Under the present axis-aligned and confined configuration, the settling velocity decreases with decreasing AR. Because the equivalent diameter and gravitational driving are unchanged, this difference mainly arises from hydrodynamic resistance. A smaller AR increases the particle projection normal to the settling direction and narrows the lateral fluid passages, thereby enhancing both pressure drag and gap-induced viscous resistance. The trend therefore reflects the combined effects of shape anisotropy and channel confinement, consistent with previous studies of non-spherical-particle drag and confined elliptical-particle sedimentation.[75,76] The predicted velocities closely match COMSOL results, thereby verifying our framework's ability to recover the shape-dependent hydrodynamic loads associated with non-circular geometries. The

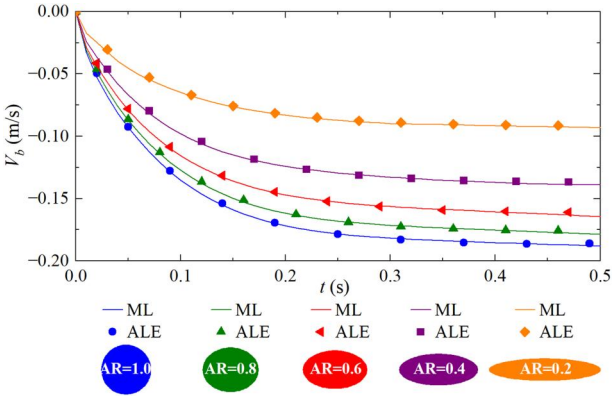

FIG. 9. Evolution of settling velocities for elliptical particles with AR values of 1.0, 0.8, 0.6, 0.4, and 0.2. Domain $0.10 \times 0.16\,\text{m}^2$; equivalent diameter 0.015 m; $Ga = 7.0$, and $Re = 2.9$–3.2. In the legend, "ML" denotes the PINN prediction and "ALE" denotes the COMSOL (ALE) reference.

corresponding trajectory, drag, and field errors are summarized in Table IV. The $AR = 0.4$ and 0.2 cases use the NA-B architecture rather than NA-A (Table X); Subsection 3 of Appendix C documents that this is forced by the regime, since NA-A carried unchanged to the most elongated case diverges under otherwise identical controls.

3. Drafting of two circular particles

This case involves two circular particles settling simultaneously in a rectangular channel (0.10 m by 0.22 m). The particles are initially positioned at 0.14 and 0.17 m from the bottom. The drafting phenomenon is examined when the trailing particle enters the wake region of the leading particle. Figure 10 shows that prior to approximately

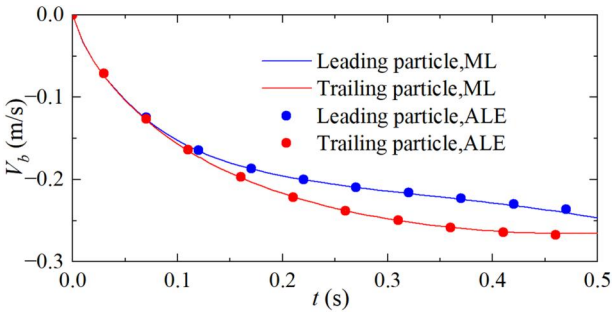

FIG. 10. Time evolution of the settling velocities for two circular particles in the drafting test. Domain $0.10 \times 0.22\ m^2$; particle diameter 0.015 m; $Ga = 7.0$, and $Re = 3.4$. "ML" denotes the PINN prediction and "ALE" the COMSOL (ALE) reference.

0.09 s, both particles settle at nearly identical speeds, indicating that the trailing particle has not yet been significantly influenced by the wake of the leading particle. As the trailing particle subsequently enters the developed wake, it encounters fluid that has been entrained downward by the leading particle. This downward wake flow reduces the slip velocity between the trailing particle and the surrounding fluid, while also lowering the stagnation pressure on its upstream surface. Consequently, the hydrodynamic drag acting on the trailing particle decreases, whereas its buoyancy-corrected weight remains essentially unchanged. The resulting increase in the net downward force accelerates the trailing particle, causing the interparticle gap to decrease. This sequence constitutes the physical mechanism underlying the classical drafting stage observed in experiments and direct numerical simulations.[1,77] The result predicted by the PINNs-based framework matches well with that in COMSOL (relative-L2 < 1.2%, Table IV), thereby verifying our framework's ability to capture the non-contact interaction between particles mediated by the surrounding flow. Figure 11 further shows the downward-moving wake and the associated pressure variation extending toward the trailing particle, providing a field-level explanation for the divergence of the two settling velocities. The corresponding trajectories and field errors are summarized in Table IV.

4. Settling of an elliptical particle with an initial slanting angle

An elliptical particle with an aspect ratio $AR = 0.6$ is released with an initial slanting angle of 30°. This case incorporates both translational and rotational dynamics. Initial guesses for the hydrodynamic force in the x direction and the torque are set to zero. These values are only numerical initializations; the lateral force and torque subsequently develop from the asymmetric flow around the inclined particle. Figure 12 compares the translational and rotational motion of the particle over time between the PINNs-based solution and COMSOL, demonstrating high agreement (relative-L^2 errors of <0.5% in trajectory, <3.0% in velocity; see Table IV). At a nonzero inclination, pressure and shear stresses are distributed asymmetrically over the two sides of the ellipse, producing both a lateral force and a hydrodynamic torque. The resulting rotation changes the projected area and flow separation around the particle, which in turn modifies its translational force. This feedback explains the observed coupling between horizontal motion, vertical settling, and rotation. Similar pressure- and shear-induced turning mechanisms have been identified for elliptical particles settling in confined channels.[76,78] Figure 13 provides a sequence of snapshots showing how the asymmetric velocity and pressure fields evolve together with the particle translation and rotation. The case with an initial slanting angle of 30° is selected as a representative example to demonstrate the capability of the proposed method in handling inclined elliptical particles. This choice does not imply any limitation of the method—our approach is applicable to other inclination angles as well.

5. Accuracy and physics-consistency

Table IV summarizes the relative L^2 errors of the PINN predictions with respect to the ALE reference solutions for all forward benchmarks, including particle trajectories, velocities, surface-integrated hydrodynamic loads, and the fluid velocity and pressure fields. Across all cases, the errors in particle trajectory and velocity remain below 5%, while those in the hydrodynamic loads remain below 8%. The corresponding field errors range from 2.1% to 7.9% for u, from 1.6% to 8.7% for v, and from 2.0% to 9.0% for p. The largest discrepancies occur for the slenderest particle, AR = 0.2, but all reported errors remain below 10%. Overall, the results demonstrate that the proposed framework retains quantitative accuracy across variations in particle geometry, particle number, and coupled translational–rotational motion. Appendix C quantifies how far the entries of Table III move when the ingredients that are not fixed by the governing equations are perturbed: the initial-load model (Subsection 1 of Appendix C), the coupling controls ζ and ε (Subsection 2 of Appendix C), and the network architecture (Subsection 3 of Appendix C).

Agreement with an ALE reference solution does not, by itself, establish that the learned fields satisfy the governing equations and boundary conditions. We therefore freeze each trained network and evaluate it on an independently resampled space–time audit set generated after training and not used for any subsequent optimization. At each of 21 equally spaced times over the simulation interval, the audit set contains 8000 uniformly sampled candidates in the computational domain and 512 points on each outer and moving boundary. Candidates located inside the particles are discarded. Surface RMS quantities on elliptical boundaries are evaluated using local arc length weights. Each diagnostic is first computed as a spatial RMS quantity at fixed time and is then averaged uniformly over the 21 evaluation times.

To place the PDE residuals on a dimensionless, scale-aware basis, each is normalized by the RMS magnitudes of its constituent equation terms. For momentum component $c \in \{x, y\}$, let $T_{c,t} = \partial_t u_c$, $T_{c,a} = \mathbf{u} \cdot \nabla u_c$, $T_{c,p} = \rho_f^{-1} \partial_c p$, and $T_{c,\nu} = -\nu \nabla^2 u_c$, with $R_{mom,c} = \sum_{\alpha \in \{t,a,p,\nu\}} T_{c,\alpha}$. Using $\|\cdot\|_{\mathrm{RMS}}$ to denote the spatial RMS at a given evaluation time, the normalized continuity and momentum imbalances are defined as

$$\eta_c(t) = \frac{\|\partial_x u + \partial_y v\|_{\mathrm{RMS}}}{\|\partial_x u\|_{\mathrm{RMS}} + \|\partial_y v\|_{\mathrm{RMS}}}, \qquad \eta_{m,i}(t) = \frac{\|R_{m,c}\|_{\mathrm{RMS}}}{\sum_{\alpha \in \{t,a,p,\nu\}} \|T_{c,\alpha}\|_{\mathrm{RMS}}}. \tag{36}$$

Boundary conditions with zero target values require an external physical scale for normalization. We take $U_c = \max_t \left|\mathbf{V}_{p,\mathrm{ALE}}(t)\right|$ over

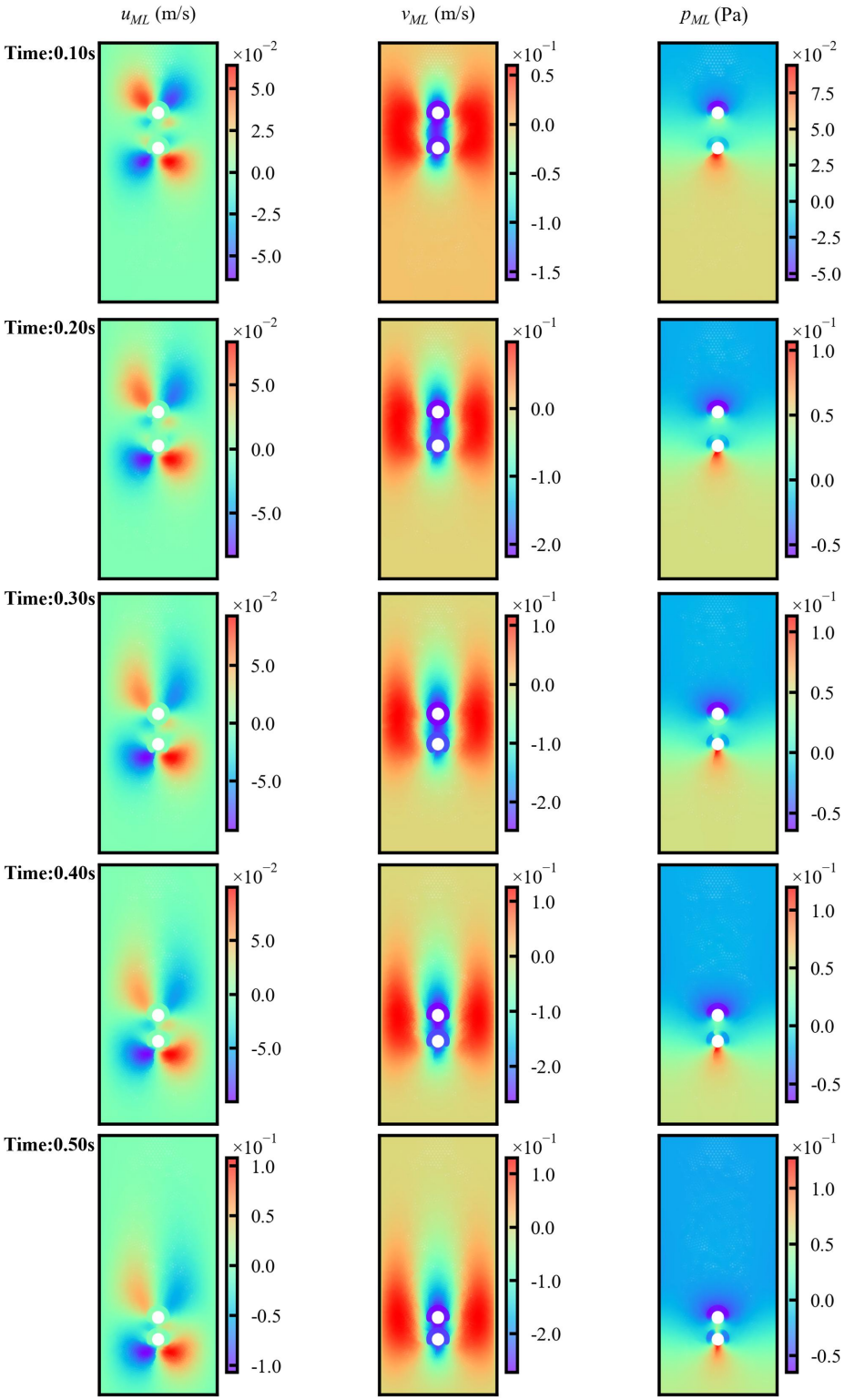

FIG. 11. Snapshots of particle positions with corresponding velocity and pressure fields illustrating the drafting phenomenon at 0.1, 0.2, 0.3, 0.4, and 0.5 s. Two circular particles (diameter 0.015 m) in a $0.10 \times 0.22\ m^2$ domain; $Ga = 7.0$, and $Re = 3.4$; fields are the PINNs (ML) prediction.

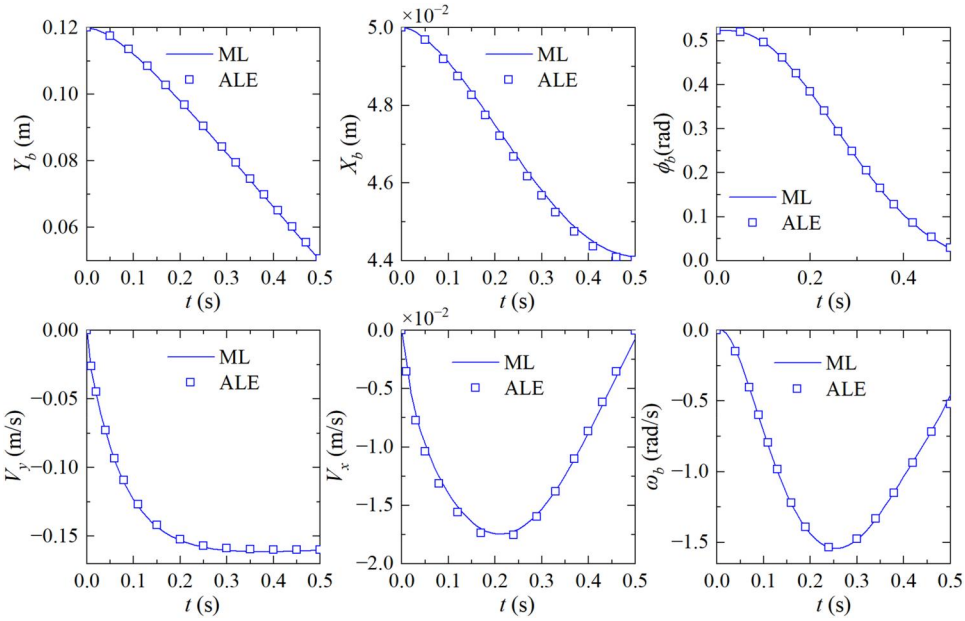

FIG. 12. Comparison of the translational and rotational dynamics of an elliptical particle with initial slanting angle between the PINNs-based framework and COMSOL. Domain $0.10 \times 0.16\,\mathrm{m}^2$; ellipse AR = 0.6, equivalent diameter 0.015 m, released at 30°; $Ga = 7.0$, $Re = 3.1$; ML = PINN, ALE = COMSOL reference.

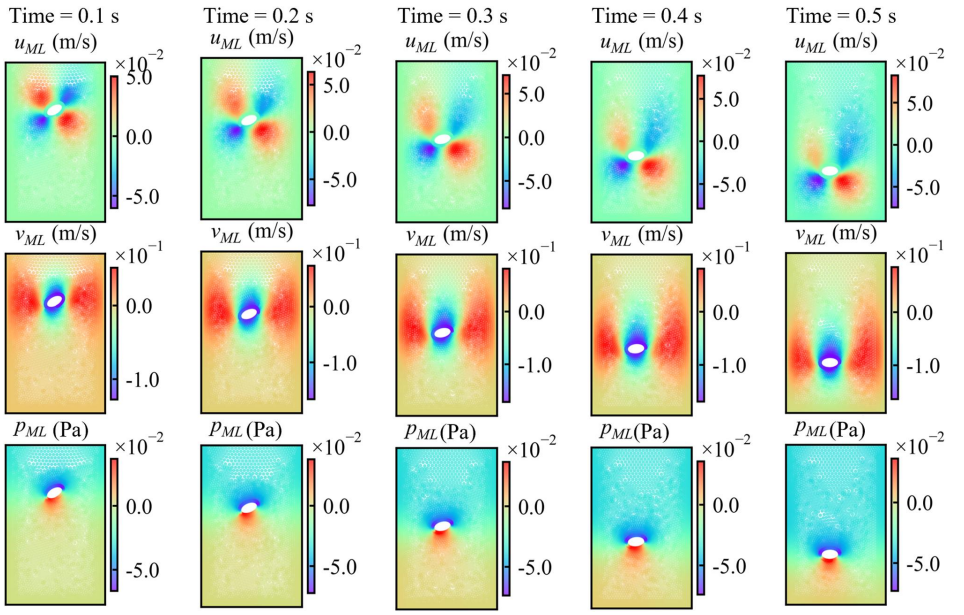

FIG. 13. Sequential snapshots of particle position along with velocity and pressure fields for the elliptical particle with a 30° initial slanting angle. Domain $0.10 \times 0.16\,\mathrm{m}^2$; ellipse AR = 0.6, equivalent diameter 0.015 m, released at 30°; $Ga = 7.0$, $Re = 3.1$; fields are the PINNs (ML) prediction.

the reported trajectory (the larger value over the two particles for drafting) and $D_{\mathrm{eq}} = 15$ mm. On moving and fixed no-slip boundaries, the velocity mismatch $\mathbf{u}_{\mathrm{PINN}} - \mathbf{u}_{\Gamma}$ is normalized by U_c. On the top free-slip boundary, the normal-velocity residual is normalized by U_c, whereas the normal gradient of tangential velocity is normalized by U_c/D_{eq}. Table V reports the uniform temporal averages of these equation and boundary residuals, denoted by $\bar{\eta}$ and $\bar{\varepsilon}$, respectively.

Table V focuses on the fluid governing equations and boundary conditions. Because the Newton–Euler equations are satisfied by construction in the discrete element method (DEM) update, a same-state rigid-body force balance would not constitute an independent diagnostic. Hydrodynamic-load accuracy relative to the ALE solution and stress-integration accuracy are already assessed in Table IV and the exact-field MMS tests, respectively. Across the post-training audit, the mean continuity imbalance ranges from 0.153% to 0.789%, while the two momentum imbalances range from 0.233% to 1.191%. All normalized velocity-based boundary residuals remain below 0.804%, and the normalized free-slip shear-gradient residual remains

TABLE V. Post-training, scale-aware resampling audit of the fluid governing equations and boundary conditions for the forward benchmarks. Equation entries are uniform temporal means of term-normalized spatial RMS residuals, whereas boundary entries are uniform temporal means normalized by U_c or U_c/D_{eq}. All entries other than U_c are percentages. "Side" denotes the larger of the left- and right-wall residuals; for the drafting case, the moving-wall entry is the larger of the two particle-surface residuals.

Case	U_c (m s^{-1})	Continuity $\bar{\eta}_c$ (%)	Momentum x/y $\bar{\eta}_{m,x}/\bar{\eta}_{m,y}$ (%)	Moving-wall no-slip $\bar{\varepsilon}_{\Gamma,u}$ (%)	Side/bottom $\bar{\varepsilon}_{\Gamma,u}$ (%)	Top free-slip $\bar{\varepsilon}_{\mathrm{top},n}/\bar{\varepsilon}_{\mathrm{top},\tau}$ (%)
Single circular particle	0.1863	0.173	0.268/0.251	0.316	0.126/0.109	0.148/0.0022
Ellipses (AR = 0.8)	0.1758	0.180	0.293/0.287	0.290	0.135/0.426	0.161/0.0022
Ellipses (AR = 0.6)	0.1608	0.156	0.259/0.233	0.295	0.124/0.307	0.170/0.0029
Ellipses (AR = 0.4)	0.1367	0.417	0.599/0.756	0.651	0.214/0.249	0.367/0.0061
Ellipses (AR = 0.2)	0.0915	0.789	1.066/1.191	0.804	0.697/0.697	0.765/0.0124
Drafting (two particles)	0.2692	0.294	0.435/0.406	0.476	0.149/0.317	0.303/0.0032
Inclined ellipse (30°)	0.1603	0.153	0.249/0.238	0.294	0.132/0.183	0.125/0.0024

below 0.0125%. Taken together, Tables IV and V provide complementary evidence: Table IV quantifies the external accuracy of the predicted solution relative to the ALE reference, whereas Table V demonstrates the satisfaction of the Navier–Stokes equations and prescribed boundary conditions at the reported scale-aware residual levels.

6. Wider-regime assessment and wake-instability-induced failure mode

The primary benchmark cases correspond to a low-Reynolds-number regime of $Re \approx 2$–4. To assess the performance of the proposed framework at higher Reynolds numbers, we consider two additional configurations obtained by jointly modifying the solid–fluid density contrast and the fluid viscosity, while retaining the circular-particle diameter $D = 15$ mm and the lateral blockage ratio $D_\perp/W = 0.15$. In both cases, $\rho_s = 1120\,\mathrm{kg/m^3}$ and $\rho_f = 960\,\mathrm{kg/m^3}$, giving $\rho_s/\rho_f = 1.167$, while the dynamic viscosities are set to 0.058 and 0.02 Pa s. These parameter combinations yield $Ga = 38.9$ and 112.8, respectively, and produce characteristic Reynolds numbers of $Re = 31.8$ and 117.9 in the corresponding ALE reference solutions. The $Re = 117.9$ case employs a taller domain and a higher release position, $0.10 \times 0.24\,\mathrm{m^2}$ with $y_0 = 0.20$ m, compared with $0.10 \times 0.16\,\mathrm{m^2}$ with $y_0 = 0.12$ m for $Re = 31.8$ case. Both configurations are simulated over a 1.0 s interval.

At $Re = 31.8$ ($Ga = 38.9$), the ALE reference exhibits centerline-aligned settling with a wake that remains approximately reflection-symmetric about the vertical centerline. At $Re = 117.9$ ($Ga = 112.8$), the reference solution develops appreciable transverse drift and particle rotation associated with an asymmetric, instability-driven wake, consistent with the dynamics reported for freely falling circular cylinders.[79] Within the present configurations, the two checkpoints therefore represent qualitatively distinct wake responses: one in which wake instability remains dynamically weak and another in which it becomes an important component of the coupled particle motion.

At $Re = 31.8$, the coupled iteration remains quantitatively accurate. The relative L^2 errors in the vertical particle position, vertical velocity, and vertical hydrodynamic force are 1.0%, 1.2%, and 1.8%, respectively, while the corresponding field errors in u, v, and p are 6.9%, 5.7%, and 5.1%. At $Re = 117.9$, the dominant vertical settling response remains reasonably reproduced, with errors of 3.1%, 6.1%, and 3.5% in the vertical position, velocity, and hydrodynamic force, respectively. The flow-field errors, however, increase substantially to 30.5%, 43.8%, and 25.1% in u, v, and p. The degradation is associated specifically with the instability-driven component of the response rather than with transverse or rotational motion in general. Over the simulated interval, the PINN predicts only approximately 44% of the transverse excursion and 57% of the rotation angle obtained from the ALE reference, indicating that it underpredicts the growth of the unstable wake modes.

This distinction is important because the framework accurately captures coupled transverse and rotational motion in the inclined-ellipse benchmark, where these responses arise from particle anisotropy, initial orientation, and the resulting deterministic fluid–particle interaction. The deterioration observed at $Re = 117.9$ should therefore not be attributed to the presence of transverse translation or rotation *per se*. Rather, it reflects the difficulty of the present physics-only PINN formulation in recovering the growth and nonlinear evolution of instability-driven wake structures. This behavior is qualitatively consistent with previous observations that data-free PINNs may converge toward overly steady solutions and fail to reproduce vortex shedding in cylinder wakes.[80,81] More generally, nonphysical attractors and local minima associated with unstable dynamical systems have been identified as an important source of PINN failure.[82]

Taken together, these results show that the framework is not intrinsically restricted to the $Re \approx 2$–4 range represented by the primary benchmarks. Its demonstrated range of validity extends to substantially higher Reynolds numbers when instability-driven wake dynamics have not become dynamically dominant. The relevant limitation identified here is not the occurrence of transverse or rotational particle motion, but the ability of the present formulation to represent the emergence and growth of unstable wake modes. The transfer-learning results discussed next concern temporal conditioning and computational reuse within this demonstrated scope, not a remedy for wake-instability-driven dynamics.

D. Analysis of the transfer learning-enhanced iterative strategy

Iterative strategies are essential in resolving coupling problems, yet deploying PINNs as solvers within these strategies presents two critical challenges. First, PINNs often struggle with transient problems

exhibiting strong temporal dependencies,[83] which can lead to non-physical solutions within an iteration which propagates errors throughout the iteration process. Second, PINNs need to be re-trained for each iteration to accommodate evolving boundary conditions, causing expensive computational costs.

We therefore examine whether transferring network parameters between successive outer coupling iterations can alleviate these difficulties. Specifically, using the benchmark case of a single circular particle settling under gravity (Sec. III C 1), we quantitatively compare the performance with and without transfer learning across iterations. In our strategy, neural network parameters from previous iterations serve as physics-informed initializations, providing a robust starting point. In contrast, the baseline approach initializes neural networks randomly for each iteration. We explicitly evaluate both strategies regarding: (a) the temporal distribution of the PDE residuals and (b) the computational efficiency across iterations.

1. NTK diagnostics and temporal residual distribution

To characterize the temporal optimization behavior, we analyze the empirical neural tangent kernel (NTK)[66,84] associated with the temporal residual loss $\mathcal{L}_r(\boldsymbol{\theta}, t)$ at each prescribed time slice. Let $\mathcal{R}_{\boldsymbol{\theta}}(t, \mathbf{x}_i)$ denote the PDE residual evaluated at spatial point $\mathbf{x}_i$ and time t. The empirical residual NTK is defined as

$$[\mathbf{K}_{\boldsymbol{\theta}}(t)]_{ij} = \left\langle \frac{\partial \mathcal{R}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}(t, \mathbf{x}_i), \frac{\partial \mathcal{R}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}}(t, \mathbf{x}_j) \right\rangle, \quad i, j = 1, 2, \ldots, N_x(t), \tag{37}$$

where $N_x(t)$ is the number of spatial points retained in the fluid domain at time t. For each diagnostic time slice, 2000 candidate points are initially sampled over the computational domain, after which points located inside the particle are removed. Consequently, the dimension of $\mathbf{K}_{\boldsymbol{\theta}}(t)$ is determined by the retained count $N_x(t)$, rather than by the nominal candidate count of 2000.

Under the local NTK approximation, the eigenvalues of $\mathbf{K}_{\boldsymbol{\theta}}(t)$ characterize the mode-wise rates at which the residual at time t can be reduced during optimization.[66] Following Wang *et al.*,[83] we use the mean NTK eigenvalue as a scalar temporal convergence rate diagnostic,

$$C(t) = \frac{\sum_{k=1}^{N_x(t)} \lambda_k(t)}{N_x(t)} = \frac{\mathrm{Trace}(\mathbf{K}_{\boldsymbol{\theta}}(t))}{N_x(t)}, \tag{38}$$

where $\{\lambda_k(t)\}_{k=1}^{N_x(t)}$ are the eigenvalues of $\mathbf{K}_{\boldsymbol{\theta}}(t)$. $N_x(t)$ is the number of spatial collocation points at time slice t, matching the dimensionality of the NTK matrix defined in Eq. (37).

Figure 14 illustrates the temporal residual loss $\mathcal{L}_r(\boldsymbol{\theta}, t)$ and temporal convergence rate $C(t)$ at the end of selected outer coupling iterations. The diagnostics are evaluated at 20 prescribed time slices. For both initialization strategies, $C(t)$ generally increases with time, indicating faster local residual optimization at later times. This temporal imbalance is consistent with the "violation of causality" problem reported for PINNs: insufficient optimization near the beginning of the time interval can generate errors that propagate forward, so small later-time residuals do not necessarily imply an accurate transient solution.[83]

Without transfer learning across iterations [Fig. 14(a)], comparatively large early-time residuals persist as the outer coupling iteration proceeds. In contrast, the TL-enhanced iteration strategy [Fig. 14(b)] progressively reduces both the early-time and overall residual levels while producing a more balanced temporal residual profile. At the common comparison point of outer iteration 19, the mean residual over the first five time slices and that over all 20 time slices are reduced to 12.6% and 13.6%, respectively, of the corresponding values obtained without transfer learning. Meanwhile, the ratio of the early-quarter to late-quarter mean residual decreases from 2.38 to 1.73, indicating that the residuals become less concentrated near the beginning of the time interval.

The NTK results nevertheless provide an important qualification. The convergence-rate diagnostic remains larger at later times under both strategies, and transfer learning does not reduce the late-to-early contrast in $C(t)$. Thus, parameter transfer does not eliminate the underlying temporal bias. Rather, it supplies a substantially better initialization that lowers the overall residual level and mitigates the resulting early-time residual concentration.

The same diagnostic was applied to an additional elliptical-particle benchmark with AR = 0.6. Between the end of the first outer iteration and the final evaluation, the early-quarter and all-slice mean residuals decrease to 25.7% and 30.1% of their initial values, respectively, while the early-to-late residual ratio decreases from 2.59 to

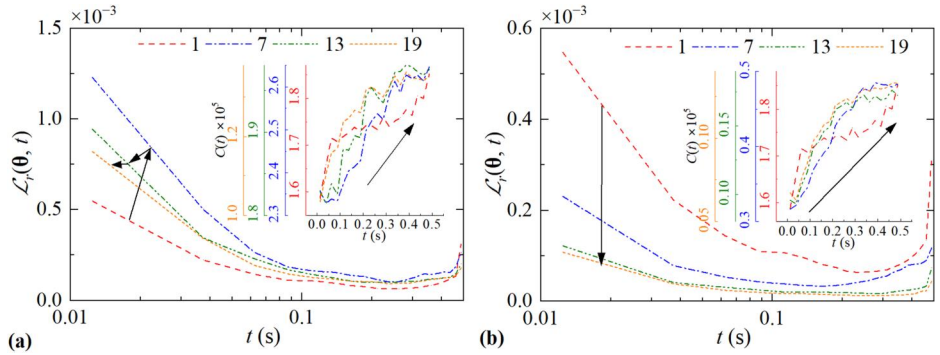

FIG. 14. Temporal residual loss $\mathcal{L}_r(\boldsymbol{\theta}, t)$ in the main panels and the empirical-NTK convergence-rate diagnostic $C(t)$ in the insets, evaluated at 20 prescribed time slices at the end of selected outer coupling iterations for the single circular-particle benchmark: (a) independent random initialization at each outer iteration, without transfer learning across iterations, and (b) initialization from the parameters obtained at the preceding outer iteration, with transfer learning across iterations.

2.08. This confirms that the improvement in temporal residual balance is not case-dependent. Causal PINNs[83] and time-marching PINNs[85] address related temporal-training difficulties by, respectively, modifying the temporal weighting of the residual loss and decomposing the time domain into sequential subintervals. The present strategy instead retains a global-in-time loss and modifies the initialization between successive FSI coupling iterations. These approaches therefore act through different mechanisms and are complementary; incorporating causal weighting or temporal domain decomposition into the present coupling framework is a promising direction for future work.

2. Computational efficiency

Because the particle trajectory is not prescribed but obtained from the stress-integrated hydrodynamic load, each outer coupling iteration defines an updated moving-boundary fluid problem. The PINN must therefore be re-optimized after the particle position, velocity, and boundary data have been updated. This is an intrinsic cost of embedding a global-in-time PINN fluid solver inside a fixed-point FSI iteration. If these fluid solves were treated as independent trainings from random initialization (without TL across iterations), the resulting cost would indeed be prohibitive for a single FSI simulation.

The present strategy changes the cost structure of this repeated optimization. The loss functional, network architecture, and governing physics are not redesigned at every outer iteration; rather, the same physics-informed objective is solved on an updated moving-boundary configuration. Only the first fluid solve is initialized randomly. All subsequent solves are initialized from the converged parameters of the preceding outer iteration, so that the optimization acts as a continuation or fine-tuning step from a nearby space–time solution rather than as a new solve from scratch. For the single-circular-particle benchmark, both strategies are converged after 19 fluid solves: one initial solve followed by 18 outer-iteration updates. Table VI combines the solver-level comparison with an iteration-resolved decomposition of this repeated-optimization cost.

The iteration-resolved timing data make this distinction quantitative. The first solve costs essentially the same with and without TL across iterations, 55.37 and 57.38 min, respectively, as expected because no preceding network is yet available for a warm start. Over the full 19 fluid solves, however, independent retraining accumulates 1270.8 min, whereas the TL-enhanced strategy accumulates 129.1 min, giving a total speedup of 9.8×. More revealingly, after excluding the first solve, the 18 subsequent fluid updates require 1213.42 min without TL but only 73.73 min with TL. Thus, the mean cost of a repeated update decreases from 67.41 to 4.09 min, corresponding to a 16.5× speedup for the repeated-update phase.

This result clarifies the practical role of transfer learning in the proposed solver. The coupled PINN framework still requires multiple fluid-network optimizations for one FSI trajectory; TL does not remove this requirement. Instead, it converts the repeated trainings from independent full optimizations into inexpensive continuation updates. Therefore, all 18 repeated updates together require only 1.33 times the cost of the first solve. The accuracy is not degraded by this reduction in cost: the particle-velocity relative error is 0.55% with TL, compared with 0.78% without TL.

The comparison with COMSOL/ALE should be interpreted in the same spirit. The ALE reference requires only 3.0 min for this forward benchmark and remains substantially faster than the PINN solver. The present result therefore does not establish a computational advantage over mature mesh-based forward solvers. Rather, it identifies transfer learning as the enabling mechanism that makes the repeated moving-boundary PINN solves tractable within the proposed two-way neural FSI framework.

In summary, our analysis shows that the transfer learning substantially alleviates, but does not eliminate, both major challenges associated with applying PINNs as the solver in the iterative strategy. It provides a warm start that markedly reduces the temporal residual magnitude, improves the balance between early- and late-time residuals, and decreases the cost of the repeated fluid-network optimizations. At the same time, the persistent late-time bias in the NTK diagnostic and the substantially lower cost of the ALE reference delimit the scope of the conclusions. Within the tested benchmarks, transfer learning should therefore be viewed as a key enabling component of the proposed PINN-based iterative strategy, rather than as a complete remedy for temporal optimization bias or a computational replacement for conventional forward solvers.

TABLE VI. Computational cost, peak memory, and particle-velocity accuracy for the single circular-particle benchmark: (a) solver-level computational cost and accuracy and (b) iteration-level decomposition of the PINN training time.

(a)

Solver	Number of outer-iteration	Wall-clock time	Peak memory	Accuracy (vel. rel. L^2)
COMSOL/ALE	⋯	3.0 min	2.21 GB	Reference
PINN without TL across iterations	19	1270.8 min	2.7 GB	0.78%
PINN with TL across iterations	19	129.1 min	2.7 GB	0.55%

(b)

Across-iteration strategy	Initial solve, iteration 1	Subsequent updates, iterations 2–19, total	Mean subsequent-update time
Without TL	57.38 min	1213.42 min	67.41 min/iteration
With TL	55.37 min	73.73 min	4.09 min/iteration

TABLE VII. Quantitative comparison of training runtime and accuracy between training from scratch and transfer learning (TL) across three test scenarios.

Scenario	Base case → target case	Metric	Train from scratch	Transfer learning	Speedup with TL
Different particle shapes	Circular particle (Sec. III C 1) → elliptical particle with equivalent diameter (Sec. III C 2)	Runtime (min)	119.5	89.0	×1.34
		L2RE (%)	0.87%	0.86%	
Different particle orientations	Section III C 2 → Sec. III C 4	Runtime (min)	149.4	125.9	×1.18
		L2RE (%)	0.90%	0.81%	
Different particle count and fluid domain	A single circular particle (Sec. III C 1) → two circular particles (Sec. III C 3)	Runtime (min)	177.3	147.6	×1.20
		L2RE (%)	1.13%	0.68%	

E. Transfer learning across different cases

In addition to improving convergence within iterative solutions (discussed in Sec. III D), the PINNs-based framework inherently enables effective transfer learning across different but related FSI scenarios. Unlike traditional numerical solvers, PINNs can repurpose the learned neural network to solve slightly modified problems, significantly accelerating convergence for new cases with common underlying physics. To demonstrate this advantage, we investigated transfer learning performance across three representative scenarios, as summarized in Table VII. In each case, the PINN pretrained on a base case was used to initialize training in a related but distinct target case. The scenarios include: (i) a change in particle shape (from circular to elliptical), (ii) a change in initial orientation of an elliptical particle, and (iii) a change in the number of particles and domain size (from single to two interacting particles).

Across all three scenarios, TL yields speedup of 1.34×, 1.18×, and 1.20× for shape, orientation, and particle-count transfers, respectively. Meanwhile, the final L2 relative errors (L2RE) for the predicted particle velocities remained comparable, or lower than, those obtained by training from scratch. These results demonstrate that the reduction in computational cost does not compromise the accuracy of the converged solutions. This efficiency gain primarily arises from two key factors. First, provided the physical properties and governing equations remain consistent, the previously learned parameters encode valuable physics insights, thereby positioning the solution closer to the global optimum from the outset. Second, initial drag force estimates derived from the base-case scenarios provide a more accurate initial guess than traditional empirical drag models, making them closer to the true drag forces experienced for the target-case particles.

F. Viscosity identification and flow-field reconstruction from particle-motion observations

We now evaluate the motion-constrained inverse formulation of Sec. II D using the motion-only information flow illustrated in Fig. 4 (right). The evaluation addresses a joint parameter–state inverse problem: whether one time-resolved rigid-body motion record is sufficient to identify the viscosity while simultaneously reconstructing velocity and pressure fields that are never supplied during training. The controlled configuration is the gravity-driven settling of a single circular particle described in Sec. III B and Table III. The reference dynamic viscosity is $\mu_{\mathrm{ref}} = 0.373$ Pa s, and the characteristic Reynolds number is $Re \approx 2$. Symmetry restricts the motion to vertical translation, with neither horizontal displacement nor rotation, and the force-consistency constraint is therefore applied only in the vertical direction. Under the dynamic-pressure convention, the analytically known load is constant in time and reduces to the submerged weight, $\mathbf{F}_{\mathrm{known},b} = (1 - \rho_{\mathrm{f}}/\rho_{\mathrm{s}})m_b\mathbf{g}$.

The inverse solve receives only the COMSOL ALE particle-centroid vertical-velocity record over $t \in [0, 1]$ s, sampled at $\Delta t_{\mathrm{obs}} = 0.01$ s ($N_{\mathrm{obs}} = 101$ observations). No interior fluid velocity or pressure sample from the reference solution enters the objective in any form, and no independently measured force is supplied: the ALE velocity and pressure fields are withheld entirely and used only for post-training evaluation. Reconstructing an interior Navier–Stokes field from moving-boundary kinematics alone would be ill-posed, and the formulation does not rely on doing so. As set out in Sec. II D, the record is used twice, both times through its temporal derivative. Differentiation and the solid momentum balance give the motion-inferred vertical-force target of Eq. (28), which is enforced through Eq. (29); propagating the same motion-derived acceleration sequence with the discrete solid time-integration rule gives the trajectory that prescribes the moving no-slip boundary. The raw record is therefore not imposed pointwise as wall data; it enters through the reconstructed trajectory. It is the second use, the body-level integral-force constraint, that supplies the observable the moving-wall condition cannot, and it is this constraint, rather than any interior measurement, that makes μ identifiable in practice.

Continuity with the forward branch is deliberate. The inverse runs use the same network architecture as the forward benchmarks (NA-A), the same surface-traction quadrature, and the same sampling scheme (S-A); the objective adds a single body-level term to the forward loss (LW-B, with force weight $\lambda_{\mathbf{F}_b} = 100$). Because the motion and its force target are fixed by the record, the outer fixed-point iteration on the hydrodynamic load is not needed at all: each inverse result below is a single Adam-L-BFGS training run. The same physics and the same machinery therefore carry the framework from forward prediction to inverse identification.

Three claims are assessed in turn: that viscosity and the withheld flow state are recovered together from the clean record and that this outcome is stable with respect to network initialization; that the recovery degrades in a characterizable way when the record is perturbed; and that the resulting capability, while distinct from classical falling-body viscometry, remains a controlled proof of concept. Only quantities compared against data that never enter the objective are reported as errors: the withheld ALE fields for the state and μ_{ref} for

the parameter. We write $\hat{\mu}$ for the terminal viscosity estimate and $E_\mu = |\hat{\mu} - \mu_{\text{ref}}|/\mu_{\text{ref}}$ for its relative error.

1. Joint recovery of viscosity and latent flow, and stability of the joint optimization

Because μ and the network parameters are optimized together, the first question is whether the outcome is a property of the formulation or of a particular initialization. The clean inversion is therefore repeated with network seeds 1234, 2345, and 3456. The observation record and the viscosity initialization, $\mu_0 = 0.1$ Pa s, a factor of 3.7 below μ_{ref}, are held fixed so that the comparison isolates the effect of network initialization within this tested setting.

Figure 15 illustrates the convergence history of the inferred viscosity under three different random initializations of the neural network parameters (seeds 1234, 2345, and 3456). In all cases, the inferred viscosity converges robustly to the true value (0.373 Pa s), despite differences in initialization. During the early stages of training, due to insufficient training of the neural network, the PINNs underestimate the flow-induced drag force, prompting the optimizer to artificially elevate the viscosity to compensate. With progressive training, the network predictions improve, leading to a gradual reduction and stabilization of the viscosity estimate at the true value. The final estimates are 0.3772, 0.3736, and 0.3723 Pa s, respectively, with mean 0.3744 Pa s. The mean estimate is approximately 0.38% above the reference, the largest individual error is $E_\mu = 1.13\%$, and the peak-to-peak spread is 1.31% of the three-run mean.

Parameter agreement alone, however, does not establish recovery of the latent flow. Table VIII therefore places the terminal viscosity and the withheld-field errors on the same seed-wise basis. The withheld ALE fields provide the independent state-reconstruction test. The u and v errors occupy comparatively narrow ranges of 7.85%–8.93% and 6.34%–7.54%, respectively. The pressure reconstruction is more sensitive to network initialization, with p errors ranging from 5.56% to 10.78%. Hence, the small spread in the identified viscosity does not imply equally small seed-to-seed variability in every latent variable. Nevertheless, all three runs recover velocity and pressure fields of comparable accuracy without receiving any interior flow data.

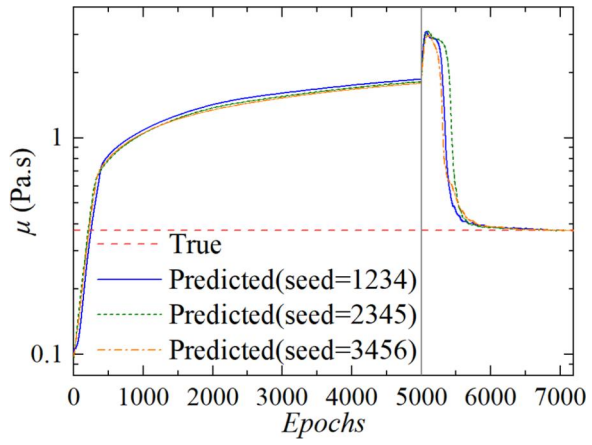

FIG. 15. Optimization histories of the jointly learned viscosity for the clean motion-only inverse benchmark. The gray vertical line marks the switch after 5000 Adam epochs to L-BFGS. All runs start from $\mu_0 = 0.1$ Pa s; the horizontal dashed line denotes $\mu_{\text{ref}} = 0.373$ Pa s.

TABLE VIII. Clean-data motion-only inverse results across network initializations. $E_\mu = |\hat{\mu} - \mu_{\text{ref}}|/\mu_{\text{ref}}$ is relative error of the terminal viscosity estimate. The u, v, and p entries are relative-L^2 errors against ALE fields withheld from training.

Network seed	Runtime (min)	E_μ (%)	Withheld-field error (%)		
			u	v	p
1234	36	1.13	8.93	7.54	10.78
2345	43	0.16	7.95	6.34	5.56
3456	35	0.19	7.85	6.65	5.79
Mean	38	0.49	8.24	6.84	7.38

Two statements follow from the clean benchmark. The viscosity and the withheld flow state are recovered together from a single motion record: all three runs reach $\hat{\mu}$ within 1.13% of μ_{ref} and reconstruct fields never used in training to the accuracy of Table VIII, without interior observation of any kind. That outcome is reached from markedly different network initializations, so the joint parameter-state training is stable in the sense tested here. The robustness to errors in the measured motion is examined next.

2. Noise propagation and empirical robustness limits

The inverse demonstration above uses clean synthetic data. We next probe one practical vulnerability, measurement noise, by perturbing only the quantity that is directly measured (the observed settling velocity) and propagating that perturbation through the same preprocessing chain used to generate each noisy solver input. Specifically, the differentiation [Eq. (27)] defined in Sec. II D maps velocity to acceleration, Eq. (28) gives $F_{\text{h},b,y}$, and re-integration of the same acceleration sequence supplies the corresponding moving boundary. The perturbation therefore affects both the moving-boundary condition in Eq. (12) and the force mismatch in Eq. (29). Two error sources accompany the differentiation in Eq. (27), and both reach the two consumers of the record, the motion-inferred force target of Eq. (28) and the re-integrated moving boundary. On the clean record the finite-difference contribution is the $\mathrm{O}(\Delta t_{obs}{}^2)$ truncation error of the second-order stencils, proportional to the third time derivative of the record, entering the force target scaled by m_b, with a corresponding finite-step error in the reconstructed boundary. The clean identification of Sec. III F 1 already contains the combined effect of these terms together with optimization and representation error, $E_\mu \leq 1.13\%$ across seeds (Table VIII); this is an end-to-end bound on the clean pipeline, not an isolated estimate of the finite-difference contribution.

Let s_{V_y} denote the sample standard deviation of the clean vertical particle-velocity history. For realization k, the perturbed observation at time t^n is

$$V^{n,(k)}_{b,\text{obs},y} = V^n_{b,\text{clean},y} + \alpha_V s_{V_y} Z^n_k, \qquad k = 1, 2, 3. \tag{39}$$

Here, Z^n_k are the independent standard Gaussian variables across times n and realizations k. Thus, $\alpha_V = 0.01$ denotes a noise standard

deviation equal to 1% of the temporal standard deviation of the clean record, rather than 1% of each instantaneous velocity. We consider $\alpha_V \in \{0.005, 0.01, 0.02, 0.05\}$, reported as 0.5%, 1%, 2%, and 5%, using noise seeds 1001–1003. The network initialization is fixed at seed 1234. On a perturbed record, differencing in Eq. (27) additionally amplifies the noise: independent velocity noise of standard deviation $\sigma_v = \alpha_V \cdot s_{V_y}$ produces acceleration noise of standard deviation $\sigma_v/(\sqrt{2}\Delta t_{\mathrm{obs}})$ at interior time levels and $(\sqrt{26}/2)\sigma_v/\Delta t_{\mathrm{obs}}$ at the two one-sided end levels. At $\Delta t_{\mathrm{obs}} = 0.01$ s, this corresponds to differentiation gains of approximately 71 and 255 s^{-1}, respectively. This acceleration noise enters the motion-inferred force target in Eq. (28) scaled by m_b, and hence the force-consistency loss of Eq. (29). The boundary branch does not retain this amplification: reintegration removes the $1/\Delta t_{\mathrm{obs}}$ scaling at leading order, cumulative sums of the differentiated perturbations reduce to $O(\sigma_v)$ combinations of the original velocity perturbations, while the boundary position acquires the corresponding time-integrated error. No smoothing or regularized differentiation is applied at any α_V: each perturbed record is propagated raw through this chain, so Table IX reports the end-to-end sensitivity of the unfiltered preprocessing chain. Filtering or regularized differentiation could alter these sensitivities, but is not evaluated here.

Table IX summarizes the sensitivity of viscosity identification and withheld-field reconstruction to raw velocity noise. The mean viscosity error grows monotonically with α_V, from 0.31% to 1.93%. The most accurate realization shows no systematic degradation across the whole range tested, 0.18%, 0.14%, 0.16%, and 0.12%, while the least accurate deteriorates steadily, from 0.40% to 1.33%, 3.25%, and 4.97%. Therefore, the rising mean is a consequence of that lengthening upper tail rather than a shift of the whole distribution.

The withheld fields degrade in step with the scalar parameter. At 0.5% and 1% noise, the mean $u/v/p$ errors remain comparable to the clean seed-1234 values 8.93%/7.54%/10.78% in Table VIII. At 2%, the pressure error rises to 12.40% while the velocity-field errors hold at 9.15%/7.62%; at 5%, all three increase to 10.81%/9.93%/14.10%. This progression attributes the loss of accuracy to perturbations of the observed motion within the present test.

TABLE IX. Sensitivity of viscosity identification and withheld-field reconstruction to raw velocity noise. For each velocity-noise amplitude α_V, entries are the mean ± sample standard deviation [minimum–maximum] over three independent noise realizations (seeds 1001–1003). $E_\mu = |\hat{\mu} - \mu_{\mathrm{ref}}|/\mu_{\mathrm{ref}}$ is relative error of the terminal viscosity estimate. The u, v, and p entries are relative-L^2 errors against ALE fields withheld from training.

		Withheld-field error (%)		
α_V (%)	E_μ (%)	u	v	p
0.5	0.31 ± 0.12	8.63 ± 0.16	7.13 ± 0.20	9.97 ± 1.66
	[0.18–0.40]	[8.46–8.78]	[6.97–7.36]	[8.93–11.88]
1.0	0.82 ± 0.61	8.61 ± 0.76	7.07 ± 0.16	11.32 ± 1.72
	[0.14–1.33]	[7.88–9.40]	[6.94–7.24]	[9.54–12.96]
2.0	1.38 ± 1.64	9.15 ± 0.59	7.62 ± 0.42	12.40 ± 1.37
	[0.16–3.25]	[8.79–9.83]	[7.31–8.10]	[11.01–13.76]
5.0	1.93 ± 2.64	10.81 ± 1.75	9.93 ± 1.42	14.10 ± 2.89
	[0.12–4.97]	[8.79–11.97]	[8.32–10.99]	[10.84–16.36]

The relatively modest viscosity errors despite the differentiation gains above can be interpreted in terms of two structural features of the inverse formulation. These are qualitative explanations, not bounds. First, the force-consistency loss of Eq. (29) distributes the constraint over all N_{obs} observation levels rather than allowing any single amplified acceleration sample to determine μ, and zero-mean fluctuations can partially average across the force history; the differentiated errors are temporally correlated, and no quantitative attenuation rate is claimed. Second, as noted above, the boundary branch carries the perturbation at the scale of the raw record, not at the amplified one. Table IX is the measurement.

Because an experiment yields one record rather than an ensemble, what matters in practice is the upper end of these spreads rather than their means, and that upper end stays bounded across the whole tested decade of amplitude. Over all 12 noise runs, no realization exceeded 4.97% in E_μ, or 11.97%, 10.99%, and 16.36% in u, v, and p; each bound is attained at $\alpha_V = 5\%$, so the largest amplitude tested bounds the study as a whole. Nor does a perturbed record put accurate recovery out of reach: at that same amplitude, one realization returned μ_{ref} to within 0.12%. What noise removes is the assurance that any particular inversion will be accurate, not the accuracy attainable in principle, and degradation over a tenfold increase in perturbation amplitude is in this sense gradual rather than abrupt.

3. Synthesis and relation to falling-body viscometry

Taken together, the clean-benchmark and perturbed-record results validate our inverse formulation in Sec. II D and test two different aspects of stability. The clean runs share the same viscosity initialization but use independent network initializations and converge to a narrow terminal viscosity band; this supports training repeatability with respect to the tested network seeds. The noise runs, for which the network initialization is fixed, instead assess the end-to-end sensitivity of the inverse pipeline to perturbations of the observed velocity and support recovery in the tested noise regime. Together with the withheld-field comparison, these results support viscosity recovery and flow reconstruction from one transient particle-velocity record, without interior fluid supervision, for the present settling configuration and the clean and low-noise settings examined here.

Classical falling-ball viscometry infers fluid viscosity from the settling speed of a spherical particle, generally after terminal motion has been established. Its simplest analytical interpretation relies on a low-Reynolds-number drag relation, while finite tube diameter and length necessitate corrections for wall and end effects.[61–63] The present inverse formulation instead uses the time-resolved transient velocity record and represents confinement through the modeled geometry and boundary conditions. It therefore does not require the particle to attain terminal velocity or the measurement to be reduced to an unbounded-domain Stokes relation. Moreover, rather than returning only a scalar viscosity, the same optimization reconstructs the time-dependent velocity and pressure fields throughout the modeled domain. This joint parameter-and-state inference constitutes an important distinction from conventional falling-body viscometry.

These findings should nevertheless be interpreted as a controlled proof of concept rather than evidence of immediate experimental readiness. The present validation is limited to a two-dimensional synthetic setting, a single settling configuration, and the clean and low-noise records examined here. Sparse temporal sampling, uncertainty

in the prescribed initial state, and errors in the assumed particle density or geometry were not perturbed in the present study. Translation to practical inverse analysis will therefore require extension to three dimensions and systematic evaluation of these factors, together with broader sources of experimental and model discrepancy. Accordingly, the present contribution is not a replacement for established viscometry, but the demonstration of a motion-constrained, physics-informed route that jointly identifies viscosity and reconstructs an otherwise unobserved flow field from one transient particle-velocity record without interior fluid supervision.

IV. CONCLUSION

We have addressed two questions posed at the outset: whether a physics-informed solver can determine the free translation and rotation of a rigid particle from hydrodynamic loads integrated over its own predicted stress field, with no labeled flow or trajectory data; and whether the observed kinematics of the body alone, without any interior velocity or pressure datum, can identify a fluid material property together with the entire unsteady interior state. Within the regime delineated below, we have provided affirmative answers and delivered one formulation rather than two solvers. Neural networks represent velocity and pressure on the moving space–time domain; the surface-integrated traction closes the planar Newton–Euler balance; and the two readings of that balance, law of motion and constraint on observed motion, define the forward and inverse modes. Only two ingredients separate them: the inverse objective adds a body-level force-consistency term, and the viscosity becomes trainable.

The forward mode was verified hierarchically against manufactured solutions, confirming second-order surface quadrature, the expected first-order solid update under velocity-dependent loading, and recovery of the exact coupled trajectory from an uninformative zero initial load in all 27 assembled runs. It was further validated against ALE references on seven benchmarks spanning particle shape, count, and inclination. Trajectory and velocity errors remain below 5%, hydrodynamic-load errors below 8%, and field errors within 9.0%; an independently resampled post-training audit bounds the term-normalized equation imbalances by 1.2% and the boundary residuals by 1%, so agreement with the reference is corroborated by direct satisfaction of the governing equations. Transfer learning across outer iterations is what renders the solver tractable: warm starts convert the repeated moving-boundary trainings into continuation updates, reducing the mean update cost from 67.4 to 4.1 min and the total wall-clock time by 9.8× at slightly improved accuracy, and relieving the early-time residual concentration, although the neural-tangent-kernel diagnostic shows that the late-time convergence bias remains; reuse across related cases yields a further 1.18–1.34×. Since a mature ALE code remains faster on any single fully specified solve, the framework's claim rests on its unified, mesh-free, label-free forward-inverse formulation, not on forward economy. A wider-regime stress test identifies one clear failure mode: accuracy persists at $\mathrm{Re} = 31.8$, whereas at $\mathrm{Re} = 117.9$ the field errors grow to 25%–44% because the physics-only formulation underpredicts the growth of unstable wake modes, capturing only about half of the transverse drift and rotation—a limitation of instability representation rather than of transverse or rotational motion as such, which the inclined-ellipse benchmark recovers accurately.

In the inverse mode, one transient centroid-velocity record is used twice, first as moving-boundary kinematics and differentiated through the momentum balance, and second as an integral-force target. We show that it suffices for joint parameter–state recovery. Starting from a viscosity a factor of 3.7 below the reference, independent network initializations identify it to within 1.13% and reconstruct the withheld velocity and pressure fields to 6.8%–8.2% mean error, with no interior observation of any kind; under record noise up to 5% of the signal's temporal standard deviation, recovery degrades gradually rather than abruptly, the worst realization remaining within 4.97% in viscosity and 16.4% in field error. The formulation thereby extends falling-body viscometry: it requires neither terminal motion nor an unbounded-domain drag relation, represents confinement through the modeled geometry, and returns the unsteady flow along with the coefficient.

Of the boundaries encountered in this study, instability-driven wake dynamics constitutes a directly observed failure mode; the others largely mark untested regimes or extensions rather than demonstrated failures. Extension to three dimensions, which enlarges the rotational state and the collocation budget but preserves the same field-load-motion closure structure, and to experimental records, whose sparse sampling and uncertainties in initial conditions and body properties must be systematically assessed, are natural next steps. Near-contact configurations, particle–particle collisions, and particle–wall collisions lie beyond the pre-contact drafting stage validated here and will require lubrication-resolving sampling together with an explicit contact closure. Long-horizon settling compounds the temporal-causality bias exposed by the same diagnostic and the cost of a single global-in-time representation, pointing to causal weighting or temporal domain decomposition as complements to cross-iteration transfer. Deformable solids would require replacing the Newton–Euler balance by a structural model with interface continuity, while retaining the broader field-load-motion coupling principle. Low solid-to-fluid density ratios are likewise uncharacterized; in the present fixed-load-history, under-relaxed outer iteration, reduced solid inertia may make the history-level fluid–solid fixed point harder to converge. Confronting the identified instability and completing these extensions are the steps that separate this controlled proof of concept from predictive and metrological use.

ACKNOWLEDGMENTS

This work was financially supported by the National Natural Science Foundation of China (Key Project #52439001) and the Research Grants Council of Hong Kong (GRF 16203123, 16214525, RIF R6008-24, TRS T22-607/24N).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Wentao Wang: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Jidong Zhao:** Funding acquisition (equal); Project administration (equal); Resources (equal); Supervision (equal); Writing – original draft (equal); Writing – review & editing (equal). **Tongming Qu:** Conceptualization (equal); Methodology (equal); Resources (equal);

Writing – original draft (equal); Writing – review & editing (equal). **Shiwei Zhao:** Writing – original draft (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The computational settings required to reproduce the reported calculations, including the network architectures, loss weights, sampling configurations, optimizer schedules, time grids, coupling controls, and random-seed protocols, are documented in the manuscript and Appendices. The code and the COMSOL ALE setup files will be released at https://github.com/wwangdg/TwoWayFSI2D, Ref. 87.

APPENDIX A: THE MANUFACTURED SOLUTION: EXACTNESS AND CLOSED-FORM LOADS

This appendix supports Sec. III A along two short derivation routes,

$$\left(\psi^{\mathrm{MMS}}, p^{\mathrm{MMS}}\right) \rightarrow \mathbf{u}^{\mathrm{MMS}} \rightarrow \mathrm{no-slip,\ incompressibility} \rightarrow \mathbf{s}^{\mathrm{MMS}} \quad (\mathrm{exactness:\ Subsection\ 1\ of\ Appendix\ A}), \tag{A1}$$

$$\left(\mathbf{u}^{\mathrm{MMS}}, p^{\mathrm{MMS}}\right) \rightarrow \boldsymbol{\sigma}^{\mathrm{MMS}}\mathbf{n} \rightarrow \left(\mathbf{F}_{\mathrm{h},b}^{\mathrm{MMS}}, M_{\mathrm{h},b}^{\mathrm{MMS}}\right) \quad (\mathrm{closed-form\ loads:\ Subsections\ 2–5\ of\ Appendix\ A}.) \tag{A2}$$

Notation follows Sec. III A; the auxiliary symbols (s, f, φ, $\mathbf{e}_r$, $\mathbf{e}_\varphi$) are defined below. Three frames are used, each with a fixed role (Fig. 16): the global Cartesian frame (x, y) of Sec. II, in which the governing equations and the load integrals of Eqs. (15) and (16) are posed; the relative Cartesian frame $X = x - X_b(t)$, $Y = y - Y_b(t)$, $r^2 = X^2 + Y^2$, in which the manufactured fields are written; and the polar basis of the relative frame (with φ its standard polar angle),

$$\mathbf{e}_r = \frac{(X,Y)}{r} = (\cos\varphi, \sin\varphi), \quad \mathbf{e}_\varphi = \hat{\mathbf{z}} \times \mathbf{e}_r = \frac{(-Y,X)}{r}, \quad r > 0, \tag{A3}$$

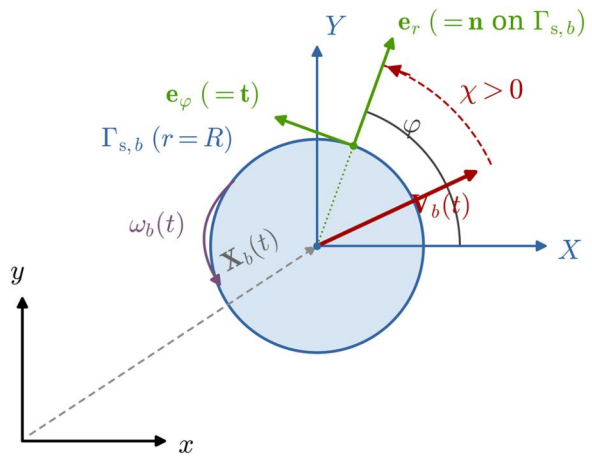

FIG. 16. General instantaneous configuration used in Appendix A: global frame (x, y); relative frame (X, Y) attached to the particle center $\mathbf{X}_b(t)$; polar basis $\mathbf{e}_r$, $\mathbf{e}_\varphi = \hat{\mathbf{z}} \times \mathbf{e}_r$, coinciding with $(\mathbf{n}, \mathbf{t})$ on the interface $\Gamma_{s,b}$. The instantaneous translation $\mathbf{V}_b(t)$ and spin $\omega_b(t)$ are arbitrary. The directed angle χ is measured from $\mathbf{V}_b$ toward $\mathbf{e}_r$ and is positive counterclockwise; and φ is the standard polar angle of the relative frame.

which on the interface $\Gamma_{s,b}$ (the circle $r = R$) coincides with the surface basis of Sec. III A, $\mathbf{n} = \mathbf{e}_r$ and $\mathbf{t} = \mathbf{e}_\varphi$, the normal pointing from the solid into the fluid. All spatial derivatives below are taken at fixed time t, and the two-dimensional loads are per unit spanwise length. The derivation is carried in terms of the projections $\mathbf{V}_b \cdot \mathbf{e}_r$ and $\mathbf{V}_b \cdot \mathbf{e}_\varphi$. For presentation, when $\| \mathbf{V}_b \|_2 > 0$, their angular dependence may be represented by the derived, signed angle,

$$\chi = \mathrm{atan2}(\hat{\mathbf{z}} \cdot (\mathbf{V}_b \times \mathbf{e}_r), \mathbf{V}_b \cdot \mathbf{e}_r), \tag{A4}$$

so that

$$\cos\chi = \frac{\mathbf{V}_b \cdot \mathbf{e}_r}{\| \mathbf{V}_b \|_2}, \qquad \sin\chi = \frac{\hat{\mathbf{z}} \cdot (\mathbf{V}_b \times \mathbf{e}_r)}{\| \mathbf{V}_b \|_2}. \tag{A5}$$

The angle is used for presentation only. At $\mathbf{V}_b = \mathbf{0}$, it is undefined, but both projections, and hence the derivation, remain well defined. It is distinct from the particle rotation angle ϕ_b of Sec. II B (identically zero in the verification case) and from the network parameters $\boldsymbol{\theta}$.

1. Velocity field and exactness

Writing the stream function of Sec. III A as $\psi^{\mathrm{MMS}} = s\beta(r) - \omega_b R^2 \ln(r/R)$ with $s = V_x Y - V_y X$, and using $\partial_x s = -V_y$, $\partial_y s = V_x$, $\partial_x r = X/r$, $\partial_y r = Y/r$, the components of $\mathbf{u}^{\mathrm{MMS}} = \left(\partial_y \psi^{\mathrm{MMS}}, -\partial_x \psi^{\mathrm{MMS}}\right)$ follow by direct differentiation,

$$u^{\mathrm{MMS}} = \partial_y \psi^{\mathrm{MMS}} = V_x \beta(r) + s\beta'(r)\frac{Y}{r} - \omega_b R^2 \frac{Y}{r^2}, \tag{A6}$$

$$v^{\mathrm{MMS}} = -\partial_x \psi^{\mathrm{MMS}} = V_y \beta(r) - s\beta'(r)\frac{X}{r} + \omega_b R^2 \frac{X}{r^2}, \tag{A7}$$

i.e., in vector form, since $(Y, -X)/r = -\mathbf{e}_\varphi$ and $(-Y, X)/r = \mathbf{e}_\varphi$,

$$\mathbf{u}^{\mathrm{MMS}} = \mathbf{V}_b \beta(r) - s\beta'(r)\mathbf{e}_\varphi + \omega_b \frac{R^2}{r}\mathbf{e}_\varphi. \tag{A8}$$

At $r = R$, where $\beta = 1$ and $\beta' = 0$, this reduces to $\mathbf{u}^{\mathrm{MMS}} = \mathbf{V}_b + \omega_b R \mathbf{e}_\varphi = \mathbf{V}_b + \omega_b \hat{\mathbf{z}} \times (\mathbf{x} - \mathbf{X}_b)$, the rigid-body surface velocity of Eq. (17): the moving no-slip condition holds exactly. Incompressibility is automatic for any stream function. The momentum equation is satisfied by construction: with $\mathcal{R}_m$ the residual operator of Sec. III A, the source is $\mathbf{s}^{\mathrm{MMS}} = \mathcal{R}_m\left(\mathbf{u}^{\mathrm{MMS}}, p^{\mathrm{MMS}}\right)$, the time derivative including the chain-rule contribution of the moving frame, $\partial_t|_{\mathbf{x}} = \partial_t|_{(X,Y)} - \mathbf{V}_b \cdot \nabla$; the loss penalizes $\left[\mathcal{R}_m(\mathbf{u}_{\boldsymbol{\theta}}, p_{\boldsymbol{\theta}}) - \mathbf{s}^{\mathrm{MMS}}\right]/\rho_{\mathrm{f}}$, which vanishes identically at the manufactured pair by definition.

2. Polar-basis projections

Two identities follow directly from the definitions: $\mathbf{e}_\varphi \cdot \mathbf{e}_r = 0$, and

$$s = V_x Y - V_y X = -r(\mathbf{V}_b \cdot \mathbf{e}_\varphi) = r \| \mathbf{V}_b \|_2 \sin\chi. \tag{A9}$$

Projecting $\mathbf{u}^{\mathrm{MMS}}$ of Eq. (A8) onto $\mathbf{e}_r$ (the $\mathbf{e}_\varphi$ terms drop out) gives

$$u_r = \mathbf{u}^{\mathrm{MMS}} \cdot \mathbf{e}_r = \beta(r)(\mathbf{V}_b \cdot \mathbf{e}_r) = \| \mathbf{V}_b \|_2 \beta(r) \cos\chi, \tag{A10}$$

projecting onto $\mathbf{e}_\varphi$ and writing $f(r) = r\beta(r) = 2R^2/r - R^4/r^3$,

$$u_\varphi = \mathbf{u}^{\mathrm{MMS}} \cdot \mathbf{e}_\varphi = (\mathbf{V}_b \cdot \mathbf{e}_\varphi)\beta - s\beta' + \omega_b \frac{R^2}{r}$$
$$= (\mathbf{V}_b \cdot \mathbf{e}_\varphi)\left[\beta + r\beta'\right] + \omega_b \frac{R^2}{r} = (\mathbf{V}_b \cdot \mathbf{e}_\varphi) f'(r) + \omega_b \frac{R^2}{r}, \quad \text{(A11)}$$

the middle step substituting $s = -r(\mathbf{V}_b \cdot \mathbf{e}_\varphi)$ and the last using the product rule $f' = \beta + r\beta'$. The surface values used below are $f(R) = R$, $f'(R) = 1$, and $f''(R) = -8/R$: the two derivatives are the origin of the coefficients $9 - 1 = 8$ in Eq. (A14).

3. Surface stresses

In the cylindrical components of the relative frame, $\sigma_{rr} = -p^{\mathrm{MMS}} + 2\mu\partial_r u_r$ and $\sigma_{r\varphi} = \mu\left[r\partial_r(u_\varphi/r) + r^{-1}\partial_\varphi u_r\right]$. The viscous normal stress vanishes on the surface, $\partial_r u_r|_{r=R} = (\mathbf{V}_b \cdot \mathbf{e}_r)\beta'(R) = 0$ (and the rotlet has $u_r = 0$). The shear stress is linear in the velocity field, so the translational and rotlet parts of $u_\varphi = u_\varphi^{\mathrm{tr}} + u_\varphi^{\mathrm{rot}}$ from Eq. (A11), with $u_\varphi^{\mathrm{tr}} = (\mathbf{V}_b \cdot \mathbf{e}_\varphi) f'(r)$ and $u_\varphi^{\mathrm{rot}} = \omega_b R^2/r$, contribute separately. The translational part enters through both terms of $\sigma_{r\varphi}$: using $\partial_\varphi \mathbf{e}_r = \mathbf{e}_\varphi$,

$$r\partial_r\left(\frac{u_\varphi^{\mathrm{tr}}}{r}\right) = (\mathbf{V}_b \cdot \mathbf{e}_\varphi)\left(f'' - \frac{f'}{r}\right)^{r=R} = -\frac{9}{R}(\mathbf{V}_b \cdot \mathbf{e}_\varphi),$$
$$\frac{1}{r}\partial_\varphi u_r = \frac{\beta(r)}{r}(\mathbf{V}_b \cdot \mathbf{e}_\varphi)^{r=R} = \frac{1}{R}(\mathbf{V}_b \cdot \mathbf{e}_\varphi), \quad \text{(A12)}$$

the rotlet part has $u_r = 0$, so it enters through the first term only,

$$r\partial_r\left(\frac{u_\varphi^{\mathrm{rot}}}{r}\right) = r\partial_r\left(\frac{\omega_b R^2}{r^2}\right)^{r=R} = -2\omega_b. \quad \text{(A13)}$$

Assembling the three contributions,

$$\sigma_{r\varphi}\|_{r=R} = \mu\left[(-9+1)\frac{\mathbf{V}_b \cdot \mathbf{e}_\varphi}{R} - 2\omega_b\right]$$
$$= -\frac{8\mu}{R}(\mathbf{V}_b \cdot \mathbf{e}_\varphi) - 2\mu\omega_b$$
$$= 8\mu\frac{\|\mathbf{V}_b\|_2 \sin\chi}{R} - 2\mu\omega_b, \quad \text{(A14)}$$

the last equality using $\mathbf{V}_b \cdot \mathbf{e}_\varphi = -\|\mathbf{V}_b\|_2 \sin\chi$ [Eq. (A9)].

4. Traction

On $\Gamma_{s,b}$, where $\mathbf{n} = \mathbf{e}_r$ and $\mathbf{t} = \mathbf{e}_\varphi$, the traction is $\boldsymbol{\sigma}^{\mathrm{MMS}}\mathbf{n} = \sigma_{rr}\mathbf{n} + \sigma_{r\varphi}\mathbf{t}$; with $\sigma_{rr}|_{r=R} = -p^{\mathrm{MMS}}$ Subsection 3 of Appendix A and the decomposition $(\mathbf{I}_2 - \mathbf{n}\mathbf{n}^\top)\mathbf{V}_b = (\mathbf{V}_b \cdot \mathbf{t})\mathbf{t}$,

$$\boldsymbol{\sigma}^{\mathrm{MMS}}\mathbf{n}\|_{\Gamma_{s,b}} = -p^{\mathrm{MMS}}\mathbf{n} - \frac{8\mu}{R}(\mathbf{I}_2 - \mathbf{n}\mathbf{n}^\top)\mathbf{V}_b - 2\mu\omega_b\hat{\mathbf{z}} \times \mathbf{n}. \quad \text{(A15)}$$

5. Load integrals

On $\Gamma_{s,b}$ the position vector is $(X, Y) = R\,\mathbf{n}$, so $p^{\mathrm{MMS}}|_{r=R} = p_0 + \gamma\mu(\mathbf{V}_b \cdot \mathbf{n})/R$. With $\mathbf{n} = (\cos\varphi, \sin\varphi)$ and $d\Gamma = R\,d\varphi$,

$$\oint_{\Gamma_{s,b}} \mathbf{n}\,d\Gamma = \mathbf{0}, \quad \oint_{\Gamma_{s,b}} \mathbf{t}\,d\Gamma = \mathbf{0}, \quad \text{(A16)}$$

$$\oint_{\Gamma_{s,b}} \mathbf{n}\mathbf{n}^\top d\Gamma = R\int_0^{2\pi}\begin{pmatrix}\cos^2\varphi & \sin\varphi\cos\varphi \\ \sin\varphi\cos\varphi & \sin^2\varphi\end{pmatrix} d\varphi = \pi R\,\mathbf{I}_2. \quad \text{(A17)}$$

Substituting the traction of Eq. (A15) into the force integral of Eq. (15),

$$\mathbf{F}_{\mathrm{h},b}^{\mathrm{MMS}} = -p_0\oint\mathbf{n}\,d\Gamma - \frac{\gamma\mu}{R}\left(\oint\mathbf{n}\mathbf{n}^\top d\Gamma\right)\mathbf{V}_b$$
$$- \frac{8\mu}{R}\left(\oint(\mathbf{I}_2 - \mathbf{n}\mathbf{n}^\top)\,d\Gamma\right)\mathbf{V}_b - 2\mu\,\omega_b\oint\hat{\mathbf{z}} \times \mathbf{n}\,d\Gamma$$
$$= \mathbf{0} - \frac{\gamma\mu}{R}\,\pi R\,\mathbf{V}_b - \frac{8\mu}{R}(2\pi R\,\mathbf{I}_2 - \pi R\,\mathbf{I}_2)\mathbf{V}_b - \mathbf{0}$$
$$= -\gamma\pi\mu\,\mathbf{V}_b - 8\pi\mu\,\mathbf{V}_b = -(8+\gamma)\,\pi\mu\,\mathbf{V}_b, \quad \text{(A18)}$$

which for $\gamma = 4$ is $-12\pi\mu\,\mathbf{V}_b$. For the torque, Eq. (16) with $\boldsymbol{\xi}_b = R\mathbf{n}$: the pressure traction is parallel to $\mathbf{n}$ ($R\,\mathbf{n} \times \mathbf{n} = 0$), only the tangential traction contributes ($\mathbf{n} \times \mathbf{t} = \hat{\mathbf{z}}$), and

$$M_{\mathrm{h},b}^{\mathrm{MMS}} = R\oint\sigma_{r\varphi}\,d\Gamma$$
$$= R\left[-\frac{8\mu}{R}\underbrace{\oint(\mathbf{V}_b \cdot \mathbf{t})\,d\Gamma}_{=\mathbf{V}_b\cdot\oint\mathbf{t}d\Gamma\,=\,0} - 2\mu\,\omega_b\oint d\Gamma\right] = -4\pi\mu R^2\omega_b, \quad \text{(A19)}$$

the classical rotlet torque. In the verification case of Sec. III A, $\omega_b \equiv 0$, so $M_{\mathrm{h},b}^{\mathrm{MMS}} = 0$ and only the translational channel carries load.

APPENDIX B: TRAINING CONFIGURATIONS

Table X below outlines the configurations adopted for loss weighting, neural network architecture, sampling strategies, and optimization procedures across all tested cases: Apart from the collocation counts, which scale with domain size and particle number, the network architecture is the only case-dependent entry; Subsection 3 of Appendix C reports the evidence supporting that one exception.

1. Loss weights

- **LW-A:** (1, 1, 1, 10, 10, 100, 100), corresponding to: continuity, x-momentum, y-momentum, initial condition (IC), free-slip condition, fixed wall, and moving wall, respectively.
- **LW-B**: (1, 1, 1, 10, 10, 200, 1000, 100), with an additional term for drag force loss.

2. Network architecture

- **NA-A:** A standard fully connected network (FCN) comprising seven hidden layers, each with 50 neurons, and using the tan h activation function.
- **NA-B:** A modified FCN incorporating random Fourier feature embeddings (with embedding dimension 128 and scale 10.0) at the input layer, combined with random weight factorization (mean 0.5, standard deviation 0.1) applied to the network

TABLE X. Summary of training configurations.

Section	Losses weights	Network architecture	Sampling	Optimizer
III C 1	LW-A	NA-A	S-A	OP-A
III C 2 (1.0)	LW-A	NA-A	S-A	OP-A
III C 2 (0.8)	LW-A	NA-A	S-A	OP-A
III C 2 (0.6)	LW-A	NA-A	S-A	OP-A
III C 2 (0.4)	LW-A	NA-B	S-A	OP-A
III C 2 (0.2)	LW-A	NA-B	S-A	OP-A
III C 3	LW-A	NA-A	S-B	OP-A
III C 4	LW-A	NA-A	S-A	OP-A
III C 6	LW-A	NA-A	S-A	OP-A
III D (without TL across iterations)	LW-A	NA-A	S-A	OP-B
III D (with TL across iterations)	LW-A	NA-A	S-A	OP-A
III F	LW-B	NA-A	S-A	OP-A

parameters. The hidden layers use the GELU activation function and consist of four layers with 128 neurons each. Further details on the implementation of random Fourier feature embeddings and random weight factorization can be found in Ref. 86.

3. Sampling strategy

- S-A: Hammersley sampling is used unless otherwise specified.
  - **Initial condition:** 3000 points
  - **Boundary conditions:**
    - Fixed wall (x direction): 5000 points
    - Fixed wall (y direction): 3000 points
    - No-slip condition: 3000 points
    - Moving wall: 10 000 points sampled in the angular-temporal space $[0,\ 2\pi] \times [0,\ T]$, representing local polar coordinates (angle and time) for moving wall. These points are first mapped onto the particle's local surface using parametric expression (e.g., circles or ellipses in Cartesian space) and then transformed to the global frame via time-dependent rigid-body rotation and translation.
  - **Collocation points:**
    - 20 000 points over $[0, L] \times [0, H] \times [0, T]$
    - Refinement near fixed walls:
      - Left/right walls (x direction): 5000 points each within $0.1L$
      - Down wall (y direction): 3000 points within $0.07H$
    - Refinement near moving wall: 10 000 points sampled in the space $[r, \varphi, t] \in [1, 3] \times [0, 2\pi] \times [0, T]$, where r is a normalized radial scaling factor applied to the parametric boundary shape. These points are mapped to Cartesian coordinates using the parametric form (e.g., ellipses or circles in Cartesian space) scaled by r, and then subjected to time-dependent rigid-body motion to obtain their global positions.
    - Wall drag force calculation: 10 000 points sampled on a structured 100×100 grid in angular-temporal space $[0, 2\pi] \times [0,\ T]$. These grid points ensure consistent spatial–temporal resolution for reconstructing a time series of drag force. The points are mapped to the local parametric surface and transformed to the global frame via time-dependent rigid-body motion.
- **S-B:** Identical to S-A, except
  - **Boundary conditions:**
    - Fixed wall (x direction): 7000 points
  - **Collocation points:**
    - 30 000 points over $[0, L] \times [0, H] \times [0, T]$

4. Optimizer (OP)

- **OP-A:**
  - **First iteration:** Adam optimizer (learning rate = 0.001, 10 000 epochs), followed by L-BFGS (using the PyTorch default settings).
  - **Subsequent iterations:** L-BFGS (using the PyTorch default settings) only.
- **OP-B:**
  - **In every iteration:** Adam (learning rate = 0.001, 10 000 epochs) followed by L-BFGS (using the PyTorch default settings).

APPENDIX C: SENSITIVITY TO THE INITIAL-LOAD ESTIMATE, COUPLING CONTROLS, AND NETWORK ARCHITECTURE

The forward results of Sec. III C rest on three ingredients that are not fixed by the governing equations: the empirical hydrodynamic-load estimate that starts the outer iteration [Eqs. (18)–(21)], the relaxation factor ζ and outer tolerance ε_{out} that control it [Eqs. (22) and (26)], and the network architecture. None of the three is fixed by the physics, so each is a candidate explanation for the reported forward errors rather than a property of the formulation. This appendix quantifies how far those errors move when each ingredient is perturbed in turn.

A prior question is how much varies at all. Of the four control groups in Table X, two are identical for every one of the seven

forward benchmarks: the loss weights (LW-A) and the optimizer schedule (OP-A). LW-B appears only in the motion-constrained inverse study of Sec. III F, which carries an additional observation term absent from the forward loss, and OP-B only in the no-transfer ablation of Sec. III D, where the warm start is removed by construction rather than by tuning. The sampling group changes once, from S-A to S-B for the drafting case, and only in point counts, which scale with the larger domain and the second particle. The single genuinely case-dependent choice is therefore the network architecture used for the $AR = 0.4$ and 0.2 ellipses.

The outcome, in short, is that the initializer alters the iteration path but leaves the converged endpoints within bounded differences (Subsection 1 of Appendix C); strong under-relaxation is necessary and the setting of the outer tolerance is adopted as a work–precision compromise (Subsection 2 of Appendix C); and the network architecture is immaterial for the single circular particle but genuinely regime-dependent for strongly elongated bodies, which is the documented basis for the one case-dependent entry in Table X (Subsection 3 of Appendix C). Throughout, $N_{\mathrm{solve}} = \ell + 1$ denotes the one-based number of fluid solves through outer iteration ℓ, and "pp" denotes percentage points of a relative-L^2 error. All seven forward benchmarks use the same coupling controls, $\zeta = 0.1$ and $\varepsilon_{\mathrm{out}} = 0.005$.

Three conventions apply throughout. First, every comparison is paired: within a pair only the stated ingredient is changed, while the physical problem, sampling, loss weights, optimizer schedule, time grid, and solve budget are held at the production values of Appendix B. Second, all errors use the metrics defined in Sec. III B so that the entries below are directly comparable with Table IV. Third, unless stated otherwise each configuration is a single run at one random initialization; the seed-to-seed spread resolved by the three-seed protocol of Sec. III A is not resolved here, so differences of a few hundredths of a percentage point are not interpreted. A run is reported as divergent when the left-hand side of Eq. (26) remains above unity without showing a decreasing trend for three consecutive outer iterations. All runtimes are end-to-end wall-clock values obtained on the HKUST HPC4 cluster, with one NVIDIA GeForce RTX 4090D GPU, two CPU cores, and 16 GB of host memory allocated per run; they are recorded for reproducibility and are not hardware-normalized, so they are not read as a throughput comparison.

1. The empirical initial-load model

Equations (18)–(21) supply only $\mathbf{F}^{\mathrm{est},(1)}_{\mathrm{h},b}$; from the second outer iteration onward, Eq. (22) progressively replaces that estimate with surface tractions evaluated from the two-dimensional PINN solution. The question is therefore whether the fixed point inherits its starting estimate. We answer it with a paired study over all seven physical forward benchmarks, comparing the production two-dimensional-cylinder model against a deliberately mismatched alternative built from a three-dimensional sphere correlation, with every other control held fixed within each pair. The sphere-based arm is not an arbitrary perturbation. A three-dimensional sphere correlation is the drag law a particle-scale initialization would most naturally supply, and transferring it to a planar body is exactly the approximation whose consequences are at issue; the pairing therefore measures that approximation rather than a generic disturbance of the initial guess.

The alternative is constructed as follows. Writing the planar body mass per unit span in terms of an area-equivalent diameter, $D_{\mathrm{eq}} = (4m_b/\pi\rho_{\mathrm{s}})^{1/2}$, a notional sphere of the same density and diameter has mass $m_{\mathrm{sph},b} = \pi/6\rho_{\mathrm{s}}D^3_{\mathrm{eq}}$ and drag,

$$\mathbf{F}^{\mathrm{sph}}_{\mathrm{h},b}(t) = -\frac{\pi}{8}\rho_{\mathrm{f}}D^2_{\mathrm{eq}}C^{\mathrm{sph}}_{\mathrm{D}}(\mathrm{Re}_{\mathrm{eq}}) \parallel \mathbf{V}_{b\to\mathrm{f}}(t)\|_2\mathbf{V}_{b\to\mathrm{f}}(t),$$
$$C^{\mathrm{sph}}_{\mathrm{D}} = \frac{24}{\mathrm{Re}_{\mathrm{eq}}}\left(1 + \frac{\mathrm{Re}_{\mathrm{eq}}}{48}\right), \tag{C1}$$

with $\mathrm{Re}_{\mathrm{eq}} = \rho_{\mathrm{f}}D_{\mathrm{eq}} \parallel \mathbf{V}_{b\to\mathrm{f}}\|_2/\mu$. This force is not applied to the planar Newton–Euler equation directly; the acceleration it implies is mapped to the planar body through $m_b/m_{\mathrm{sph},b} = 3/(2D_{\mathrm{eq}})$, which has dimensions of inverse length and therefore returns a force per unit span,

$$\mathbf{F}^{\mathrm{sph},(1)}_{\mathrm{h},b}(t) = \frac{m_b}{m_{\mathrm{sph},b}}\mathbf{F}_{\mathrm{sph},b}(t)$$
$$= -\frac{1}{2}\rho_{\mathrm{f}}D_{\mathrm{eq}}C^{\mathrm{sph}}_{\mathrm{D,eff}}(\mathrm{Re}_{\mathrm{eq}})\|\mathbf{V}_{b\to\mathrm{f}}(t)\|_2\mathbf{V}_{b\to\mathrm{f}}(t),$$
$$C^{\mathrm{sph}}_{\mathrm{D,eff}} = \frac{3\pi}{8}C^{\mathrm{sph}}_{\mathrm{D}}. \tag{C2}$$

The two models differ in drag law, in their geometry and confinement treatment, and in their added-mass prescription. The pairing therefore perturbs the complete initial-load model rather than isolating C_{D}, and $C^{\mathrm{sph}}_{\mathrm{D,eff}}$ is a dimensionally consistent construction, not a physical drag coefficient for a two-dimensional cylinder.

Three observations follow from Table XI. First, both models satisfy the full outer criterion in all seven benchmarks: convergence of the coupled iteration is a property of the relaxation scheme, not of the starting estimate. Second, neither model holds a systematic path advantage. The cylinder-based model reaches the criterion earlier for the single circular particle, the AR = 0.4 and 0.2 ellipses, and drafting; the sphere-correlation-based model does so for the AR = 0.8 and 0.6 ellipses and the inclined ellipse. The spread is one to three fluid solves, and the end-to-end wall times of both arms fall in the same 103–177 min band with no consistent direction. Third, the paired endpoints differ by at most 0.269/0.494/2.084 pp in the solid $y/v_y/F_{\mathrm{h},b,y}$ errors and 2.051/2.366/2.079 pp in the Delaunay-weighted $u/v/p^{\circ}$ errors.

These bounds are absolute and for channels whose reported values are themselves of order 1%. We therefore claim bounded end point sensitivity to the two initial-load models tested, not initializer-independent digits. The bound is nevertheless consistent with the algorithmic role of the estimate, which enters only through $\mathbf{F}^{\mathrm{est},(1)}_{\mathrm{h},b}$, and it is corroborated at a third, structurally unrelated starting point: in the manufactured-solution setting of Sec. III A, all 27 coupled runs converge from a zero initial load and recover the exact trajectory. Read together, a quasi-steady cylinder estimate, a mapped sphere correlation, and a null estimate all drive the same iteration to the same solutions within the differences quantified above. The cylinder model of Eqs. (18)–(21) is retained for all primary results because it is the physically appropriate two-dimensional choice, not because it was found to converge faster.

TABLE XI. Paired sensitivity of the seven physical forward benchmarks to the initial hydrodynamic-load model. Paired entries are ordered cylinder-based/sphere-correlation-based, and the cylinder arm is the production run reported in Table IV. Here E_0 is the error of each initial history with respect to the reference, and $D_0 = \| F^{\mathrm{est},(1),\mathrm{sph}}_{\mathrm{h},b,y} - F^{\mathrm{est},(1),\mathrm{cyl}}_{\mathrm{h},b,y} \|_2 / \| F^{\mathrm{ref}}_{\mathrm{h},b,y} \|_2$ is the separation of the two initial histories from each other. N_{solve} is the one-based number of fluid solves through the first outer iteration at which every active load channel satisfies $\varepsilon_{\mathrm{out}} = 0.005$. Accuracy entries are absolute differences between the two arms' relative-L^2 error percentages, in percentage points; for drafting, each arm first takes the worse of its two particles.

Case	E_0 cyl./sph. (%)	D_0 (%)	N_{solve} cyl./sph.	Runtime cyl./sph. (min)	Solid $\lvert\Delta E_{y/v_y/F\mathrm{h,b,y}}\rvert$ (pp)	Flow $\lvert\Delta E_{u/v/p^\circ}\rvert$ (pp)
Single circular	6.72/7.27	6.30	19/21	129.1/103.0	0.007/0.063/0.026	0.515/0.246/0.066
Ellipse $AR = 0.8$	6.29/7.66	6.42	20/19	112.1/113.1	0.049/0.115/0.509	0.411/0.211/0.123
Ellipse $AR = 0.6$	5.67/8.56	6.98	22/19	119.5/125.5	0.017/0.110/0.285	0.007/0.091/0.449
Ellipse $AR = 0.4$	4.76/10.22	8.39	18/19	110.6/121.8	0.093/0.494/2.084	0.711/1.284/2.043
Ellipse $AR = 0.2$	3.20/13.16	11.48	25/27	125.5/153.5	0.178/0.491/1.526	1.665/0.057/2.079
Drafting	7.48/9.40	6.04	27/29	177.3/170.9	0.269/0.483/0.501	2.051/2.366/1.234
Inclined ellipse	5.21/7.79	6.98	34/32	149.4/157.2	0.019/0.055/0.100	0.167/0.0003/0.326

2. The coupling controls

The same pair $(\zeta, \varepsilon_{\mathrm{out}}) = (0.1, 0.005)$ is used for every forward benchmark, so a single question governs this subsection: whether that uniform choice is necessary, whether it is sufficient, and whether it flatters the reported errors. The single circular-particle benchmark serves as the controlled setting, with the physical problem, NA-A architecture, sampling, optimizer schedule, empirical initialization, and solve budget held fixed throughout.

a. The relaxation factor

The role of ζ is best read from Eq. (22), which gives the applied load increment exactly as

$$\mathbf{F}^{\mathrm{est},(\ell)}_{\mathrm{h},b} - \mathbf{F}^{\mathrm{est},(\ell-1)}_{\mathrm{h},b} = \zeta\left(\mathbf{F}^{\mathrm{PINN},(\ell-1)}_{\mathrm{h},b} - \mathbf{F}^{\mathrm{est},(\ell-1)}_{\mathrm{h},b}\right),$$

so that ζ is a continuation step size in load space. A smaller ζ damps the load history passed to the solid update, limits the change in particle kinematics and moving boundary, and therefore presents the L-BFGS-only stage of Sec. III C 1 with a fluid problem closer to the one just solved—the mechanism on which the transfer learning across iterations depends. Varying only $\zeta \in \{0.1, 0.3, 0.5, 0.7\}$ is consistent with this reading: the $\zeta = 0.1$ arm meets the outer criterion after 19 fluid solves with a final relative vertical-force change of 0.483%, whereas the $\zeta = 0.3$, 0.5, and 0.7 arms are terminated as divergent outer-iteration after 12, 4, and 4 fluid solves.

The observed failure sequence is consistent with a loss of cross-iteration tracking at the larger load steps. A larger ζ produces a larger change in the load history passed to the solid update and hence, in general, a larger change in the particle kinematics, surface velocity, and moving boundary between consecutive outer iterations. The inherited network parameters may then no longer provide a sufficiently close initialization for the updated fluid problem. In the divergent arms, L-BFGS terminates while the composite loss remains large, so the resulting flow field and surface traction cannot be regarded as converged. Feeding this underconverged load back through Eq. (22) can further increase or disorder the next kinematic and boundary update, producing a self-reinforcing loss of warm-start tracking.

Under the stated optimizer schedule, Adam followed by L-BFGS for the initial solve and L-BFGS only for the warm-started updates, and the stated solve budget, strong under-relaxation is therefore required for this benchmark. The same $\zeta = 0.1$ brings all seven primary forward benchmarks to the outer criterion (Table XI), indicating that it is also sufficient over the tested range of cases.

b. The outer tolerance

Fixing $\zeta = 0.1$ and varying $\varepsilon_{\mathrm{out}} \in \{0.01, 0.005, 0.001\}$ then separates cost from accuracy. All three arms meet their prescribed criteria, after 15, 19, and 39 fluid solves, with final relative vertical-force changes of 0.955%, 0.483%, and 0.0787%. The corresponding ALE-referenced relative-L^2 errors in $y/v_y/F_{\mathrm{h},b,y}$ are 0.276/0.707/0.992%, 0.226/0.545/0.885%, and 0.248/0.608/0.852%. Tightening the criterion by an order of magnitude therefore more than doubles the number of fluid solves while leaving the agreement with the ALE reference unchanged to within a few hundredths of a percentage point, and non-monotonically so.

Two consequences matter for the main text. The reported setting $\varepsilon_{\mathrm{out}} = 0.005$ is adopted as a work-precision compromise (relative to 0.001 it reduces the solve count from 39 to 19 at comparable ALE agreement) and, more importantly, the discrepancies reported in Table IV are not set by where the outer iteration was stopped. Over the tested range, they are controlled by the fluid-field approximation and the optimization instead, which is the same conclusion reached by the tolerance continuation of the manufactured-solution study in Sec. III A.

3. The network architecture

NA-A is used for every forward benchmark except the $AR = 0.4$ and 0.2 ellipses, which use NA-B, and that exception is expected on physical grounds. The five ellipses of Sec. III C 2 share an area-equivalent diameter, so reducing AR at fixed area does not shrink the body but stretches it across the stream, $D_\perp = D_{\mathrm{eq}}/AR^{1/2}$, and that single stretching acts twice. It sharpens the tips, whose radius of curvature $(b^2/a = AR^{3/2}D_{\mathrm{eq}}/2)$ contracts from $0.5D_{\mathrm{eq}}$ for the circle to $0.13D_{\mathrm{eq}}$ at $AR = 0.4$ and $0.045D_{\mathrm{eq}}$ at $AR = 0.2$; and it carries them toward the side walls, the blockage ratio rising from 0.15 to 0.335 (Table III) and the confinement

multiplier carried by the initializer itself, $M_B = 1/(1 - 1.8BR)$, from 1.37 to 2.52. At $Re \approx 3$, no thin shear layer sets the near-surface scales, so they are geometric: the most elongated case asks one network to resolve a tip an order of magnitude sharper than the circle's while meeting a wall condition about one body width away. Its spectral content is correspondingly richer in high wavenumbers, and fully connected networks with smooth activations resolve such content far more slowly than low-frequency content, a bias visible in the NTK eigenspectrum.[66] The random Fourier feature embedding of NA-B counteracts that bias directly, with random weight factorization improving the conditioning of the optimization.[86] Spectral enrichment should therefore buy little where the solution is already smooth and become enabling where it is not.

The exception nevertheless raises two distinct questions: whether the single-circle conclusions survive ordinary network-design choices, and whether it is forced by the regime or merely selected for favorable numbers. For the first, we perform a strict one-factor-at-a-time (OAT) audit about NA-A on the single circular-particle benchmark, varying width (30–128), depth (4–9 hidden layers), and activation (tanh/GELU/SiLU/sin) while holding the physical problem, sampling, optimizer schedule, relaxation, outer tolerance, and budget fixed. For the second, we transfer the complete NA-A and NA-B architecture definitions reciprocally between the single circle and the strongly elongated AR = 0.2 case. NA-B changes several attributes at once and is therefore used only as this bundled transfer control, never as an OAT variant (Table XII).

All 10 configurations meet the same outer criterion after 18–21 fluid solves. Their solid $y/v_y/F_{h,b,y}$ errors span 0.17%–0.59%, 0.43%–1.20%, and 0.87%–2.21%; the flow fields are more sensitive, with $u/v/p^\circ$ spanning 1.37%–4.96%, 0.88%–4.17%, and 1.14%–3.25%. The largest degradation follows the reduction of depth to four layers, and enlargement is not monotonically beneficial—width 128 approximately doubles the baseline runtime while worsening every displayed flow error. The activation variants reduce field errors but trade those gains against force error or runtime, so no tested variant dominates across solid motion, load, flow reconstruction, and cost simultaneously. The single-circle conclusions of Sec. III C 1 are consequently not an artifact of NA-A, and NA-A is retained as the production reference on those grounds rather than selected post hoc as the best performer of this sweep.

The reciprocal transfer answers the second question. Carried unchanged to the single circle, NA-B meets the criterion after the same 19 fluid solves but moves the solid errors from 0.23/0.55/0.88% to 0.28/0.58/1.43% and the field errors from 2.08/1.55/1.98% to 2.30/1.66/2.47%: it offers no systematic accuracy gain where NA-A already works. Carried unchanged to the AR = 0.2 ellipse under controls matched to the production arm, NA-A diverges during the third fluid solve, whereas NA-B converges to the values reported in Table IV. Both limbs of the expectation are thus borne out: where the field is smooth, the enriched NA-B architecture offers no systematic advantage over NA-A; where the tips are sharp and the walls close, it separates divergence from convergence. The exception recorded in Table X

TABLE XII. Architecture sensitivity. (a) Strict one-factor-at-a-time audit about NA-A on the single circular-particle benchmark; the baseline row is the production run of Table IV and is printed once. (b) Reciprocal transfer of the complete NA-A and NA-B architecture definitions between the single-circle and AR = 0.2 cases; NA-B is a bundled full-architecture control, not an OAT family variant. Within each comparison, all nonarchitecture controls are fixed at their production values. Solid entries are ALE-referenced relative-L^2 errors in $y/v_y/F_{h,b,y}$; flow entries are $u/v/p^\circ$ errors formed from per-snapshot Delaunay lumped-area weights, independent pressure centering, and arithmetic averaging over snapshots. All errors are percentages.

(a) Local audit on the single circular-particle benchmark.

Variation	Type	Layers × width	Act.	N_{solve}	Runtime (min)	Solid $y/v_y/F_{h,b,y}$	Flow $u/v/p$
NA-A baseline	FCN	7 × 50	tan h	19	129	0.23/0.55/0.88	2.08/1.55/1.98
Width 30	FCN	7 × 30	tan h	18	85	0.24/0.76/1.57	2.77/1.84/2.67
Width 80	FCN	7 × 80	tan h	20	165	0.21/0.54/0.87	1.89/1.40/1.93
Width 128	FCN	7 × 128	tan h	20	261	0.27/0.75/1.24	2.53/1.84/2.43
Depth 4	FCN	4 × 50	tan h	20	97	0.59/1.20/2.21	4.96/4.17/3.25
Depth 5	FCN	5 × 50	tan h	20	99	0.39/0.90/1.80	3.76/2.80/2.65
Depth 9	FCN	9 × 50	tan h	18	120	0.25/0.71/1.32	2.90/1.85/2.32
GELU activation	FCN	7 × 50	GELU	20	85	0.20/0.47/0.89	1.70/1.15/1.60
SiLU activation	FCN	7 × 50	SiLU	21	137	0.17/0.43/1.36	1.37/0.88/1.14
Sin activation	FCN	7 × 50	Sin	20	118	0.17/0.52/0.89	1.57/1.09/1.67

(b) Reciprocal full-architecture transfer across regimes.

Case	Architecture	Outer-solve outcome	Solid $y/v_y/F_{h,b,y}$	Flow $u/v/p$
Single circular	NA-A (production)	Met after 19 solves	0.23/0.55/0.88	2.08/1.55/1.98
Single circular	NA-B, as is	Met after 19 solves	0.28/0.58/1.43	2.30/1.66/2.47
$AR = 0.2$ ellipse	NA-B (production)	Met after 25 solves	0.79/4.26/2.62	7.9/8.7/9.0
$AR = 0.2$ ellipse	NA-A, as is	Diverged during solve 3	···	···

is therefore forced by the regime rather than adopted to improve already-converging results, and it is reported as a scope statement: the present configuration does not admit a single architecture across the full range of aspect ratios tested.

REFERENCES

[1]J. Feng, H. H. Hu, and D. D. Joseph, "Direct simulation of initial value problems for the motion of solid bodies in a Newtonian fluid. Part 1. Sedimentation," J. Fluid Mech. **261**, 95–134 (1994).
[2]H. H. Hu, N. A. Patankar, and M. Y. Zhu, "Direct numerical simulations of fluid-solid systems using the arbitrary Lagrangian-Eulerian technique," J. Comput. Phys. **169**(2), 427–462 (2001).
[3]R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. Périaux, "A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow," J. Comput. Phys. **169**(2), 363–426 (2001).
[4]C. S. Peskin, "The immersed boundary method," Acta Numer. **11**, 479–517 (2002).
[5]M. Uhlmann, "An immersed boundary method with direct forcing for the simulation of particulate flows," J. Comput. Phys. **209**(2), 448–476 (2005).
[6]R. Gupta and R. Jaiman, "A hybrid partitioned deep learning methodology for moving interface and fluid–structure interaction," Comput. Fluids **233**, 105239 (2022).
[7]R. Gao and R. K. Jaiman, "Predicting fluid–structure interaction with graph neural networks," Phys. Fluids **36**(1), 013622 (2024).
[8]X. Fan and J.-X. Wang, "Differentiable hybrid neural modeling for fluid-structure interaction," J. Comput. Phys. **496**, 112584 (2024).
[9]Y. Lyu, Y. Zhang, Z. Gong, X. Kang, W. Yao, and Y. Pei, "A novel hybrid neural network of fluid-structure interaction prediction for two cylinders in tandem arrangement," Eng. Appl. Comput. Fluid Mech. **19**(1), 2513663 (2025).
[10]S. Tao, Z. Feng, S. Chen, W. Zhang, Z. Zhu, and Y. Liu, "Neural latent arbitrary Lagrangian-Eulerian grids for fluid-solid interaction," arXiv:2603.00792 (2026).
[11]S. Zhao, M. Saravia, H. Jiang, Z. Xue, and S. Cao, "An ALE-consistent graph neural operator-transformer framework for fluid-structure interaction," arXiv:2605.00937 (2026).
[12]G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," Nat. Rev. Phys. **3**(6), 422–440 (2021).
[13]S. Cai, Z. Mao, Z. Wang, M. Yin, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for fluid mechanics: A review," Acta Mech. Sin. **37**(12), 1727–1738 (2021).
[14]P. Sharma, W. T. Chung, B. Akoush, and M. Ihme, "A review of physics-informed machine learning in fluid mechanics," Energies **16**(5), 2343 (2023).
[15]C. Zhao, F. Zhang, W. Lou, X. Wang, and J. Yang, "A comprehensive review of advances in physics-informed neural networks and their applications in complex fluid dynamics," Phys. Fluids **36**(10), 101301 (2024).
[16]S. Goswami, C. Anitescu, S. Chakraborty, and T. Rabczuk, "Transfer learning enhanced physics informed neural network for phase-field modeling of fracture," Theor. Appl. Fract. Mech. **106**, 102447 (2020).
[17]C. P. Rao, H. Sun, and Y. Liu, "Physics-informed deep learning for computational elastodynamics without labeled data," J. Eng. Mech. **147**(8), 04021043 (2021).
[18]B. Zheng, T. C. Li, H. J. Qi, L. G. Gao, X. Q. Liu, and L. Yuan, "Physics-informed machine learning model for computational fracture of quasi-brittle materials without labelled data," Int. J. Mech. Sci. **223**, 107282 (2022).
[19]S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for heat transfer problems," J. Heat Transfer **143**(6), 060801 (2021).
[20]R. Laubscher, "Simulation of multi-species flow and heat transfer using physics-informed neural networks," Phys. Fluids **33**(8), 087101 (2021).
[21]N. Zobeiry and K. D. Humfeld, "A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications," Eng. Appl. Artif. Intell. **101**, 104232 (2021).
[22]M. Baldan, P. Di Barba, and D. A. Lowther, "Physics-informed neural networks for inverse electromagnetic problems," IEEE Trans. Magn. **59**(5), 1–5 (2023).
[23]A. Khan and D. A. Lowther, "Physics informed neural networks for electromagnetic analysis," IEEE Trans. Magn. **58**(9), 1–4 (2022).
[24]G. P. P. Pun, R. Batra, R. Ramprasad, and Y. Mishin, "Physically informed artificial neural networks for atomistic modeling of materials," Nat. Commun. **10**(1), 2339 (2019).
[25]E. Zhang, M. Dao, G. E. Karniadakis, and S. Suresh, "Analyses of internal structures and defects in materials using physics-informed neural networks," Sci. Adv. **8**(7), eabk0644 (2022).
[26]P. Li, M. L. Liu, M. Alfarraj, P. Tahmasebi, and D. Grana, "Probabilistic physics-informed neural network for seismic petrophysical inversion," Geophysics **89**(2), M17–M32 (2024).
[27]M. Rasht-Behesht, C. Huber, K. Shukla, and G. E. Karniadakis, "Physics-informed neural networks (PINNs) for wave propagation and full waveform inversions," J. Geophys. Res. **127**(5), e2021JB023120, https://doi.org/10.1029/2021JB023120 (2022).
[28]G. T. Schuster, Y. Q. Chen, and S. H. Feng, "Review of physics-informed machine-learning inversion of geophysical data," Geophysics **89**(6), T337–T356 (2024).
[29]C. Rao, H. Sun, and Y. Liu, "Physics-informed deep learning for incompressible laminar flows," Theor. Appl. Mech. Lett. **10**(3), 207–212 (2020).
[30]X. Jin, S. Cai, H. Li, and G. E. Karniadakis, "NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations," J. Comput. Phys. **426**, 109951 (2021).
[31]D. J. S. Aulakh, S. B. Beale, and J. G. Pharoah, "A generalized framework for unsupervised learning and data recovery in computational fluid dynamics using discretized loss functions," Phys. Fluids **34**(7), 077111 (2022).
[32]R. Ranade, C. Hill, and J. Pathak, "DiscretizationNet: A machine-learning based solver for Navier–Stokes equations using finite volume discretization," Comput. Methods Appl. Mech. Eng. **378**, 113722 (2021).
[33]H. Eivazi, M. Tahani, P. Schlatter, and R. Vinuesa, "Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations," Phys. Fluids **34**(7), 075117 (2022).
[34]A. Gafoor Ctp, S. Kumar Boya, R. Jinka, A. Gupta, A. Tyagi, S. Sarkar, and D. N. Subramani, "A physics-informed neural network for turbulent wake simulations behind wind turbines," Phys. Fluids **37**(1), 015110 (2025).
[35]Z. Mao, A. D. Jagtap, and G. E. Karniadakis, "Physics-informed neural networks for high-speed flows," Comput. Methods Appl. Mech. Eng. **360**, 112789 (2020).
[36]R. Qiu, R. Huang, Y. Xiao, J. Wang, Z. Zhang, J. Yue, Z. Zeng, and Y. Wang, "Physics-informed neural networks for phase-field method in two-phase flow," Phys. Fluids **34**(5), 052109 (2022).
[37]Z. Zhang, X. Yan, P. Liu, K. Zhang, R. Han, and S. Wang, "A physics-informed convolutional neural network for the simulation and prediction of two-phase Darcy flows in heterogeneous porous media," J. Comput. Phys. **477**, 111919 (2023).
[38]A. Chakraborty, A. Rabinovich, and Z. Moreno, "Physics-informed neural networks for modeling two-phase steady state flow with capillary heterogeneity at varying flow conditions," Adv. Water Resour. **185**, 104639 (2024).
[39]K. Kashinath, M. Mustafa, A. Albert, J. L. Wu, C. Jiang, S. Esmaeilzadeh, K. Azizzadenesheli, R. Wang, A. Chattopadhyay, A. Singh, A. Manepalli, D. Chirila, R. Yu, R. Walters, B. White, H. Xiao, H. A. Tchelepi, P. Marcus, A. Anandkumar, P. Hassanzadeh, and Prabhat, "Physics-informed machine learning: Case studies for weather and climate modelling," Philos. Trans. R. Soc., A **379**(2194), 20200093 (2021).
[40]Q. S. Xu, Y. L. Shi, J. L. Bamber, C. J. Ouyang, and X. X. Zhu, "Large-scale flood modeling and forecasting with FloodCast," Water Res. **264**, 122162 (2024).
[41]Z. C. Zhu, Z. F. Wang, C. M. Dong, M. Yu, H. R. Xie, X. D. Cao, L. Han, and J. S. Qi, "Physics informed neural network modelling for storm surge forecasting—A case study in the Bohai Sea, China," Coastal Eng. **197**, 104686 (2025).
[42]Y. Sun, U. Sengupta, and M. Juniper, "Physics-informed deep learning for simultaneous surrogate modeling and PDE-constrained optimization of an airfoil geometry," Comput. Methods Appl. Mech. Eng. **411**, 116042 (2023).
[43]M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," Science **367**(6481), 1026–1030 (2020).

[44]T. S. Wang, Z. Huang, Z. G. Sun, and G. Xi, "Reconstruction of natural convection within an enclosure using deep neural network," Int. J. Heat Mass Transfer **164**, 120626 (2021).
[45]R. Eusebi, G. A. Vecchi, C. Y. Lai, and M. J. Tong, "Realistic tropical cyclone wind and pressure fields can be reconstructed from sparse data using deep learning," Commun. Earth Environ. **5**(1), 8 (2024).
[46]M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," J. Comput. Phys. **378**, 686–707 (2019).
[47]Y. Zhu, W. Kong, J. Deng, and X. Bian, "Physics-informed neural networks for incompressible flows with moving boundaries," Phys. Fluids **36**(1), 013617 (2024).
[48]A. Farea, S. Khan, R. Daryani, E. C. Ersan, and M. S. Celebi, "Learning fluid-structure interaction with physics-informed machine learning and immersed boundary methods," arXiv:2505.18565 (2025).
[49]H. Zhang, R. H. Chan, and X.-C. Tai, "A meshless solver for blood flow simulations in elastic vessels using a physics-informed neural network," SIAM J. Sci. Comput. **46**(4), C479–C507 (2024).
[50]X. Xia, X. Li, J. Zhang, S. Cao, and G. Wang, "A physics-informed neural network for fluid–structure coupled simulation of a Euler–Bernoulli beam under steady flow," Ocean **2**, 9470017 (2026).
[51]R. Sundar, D. Majumdar, D. Lucor, and S. Sarkar, "Physics-informed neural networks modelling for systems with moving immersed boundaries: Application to an unsteady flow past a plunging foil," J. Fluids Struct. **125**, 104066 (2024).
[52]M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, "Deep learning of vortex-induced vibrations," J. Fluid Mech. **861**, 119–137 (2019).
[53]C. Cheng, H. Meng, Y. Z. Li, and G. T. Zhang, "Deep learning based on PINN for solving 2-DOF vortex-induced vibration of cylinder," Ocean Eng. **240**, 109932 (2021).
[54]H. Tang, Y. Liao, H. Yang, and L. Xie, "A transfer learning-physics informed neural network (TL-PINN) for vortex-induced vibration," Ocean Eng. **266**, 113101 (2022).
[55]M. A. Calicchia, R. Mittal, J. H. Seo, and R. Ni, "Reconstructing the pressure field around swimming fish using a physics-informed neural network," J. Exp. Biol. **226**(8), jeb244983 (2023).
[56]H. Wang, F. Wu, Y. Liu, X. He, S. Feng, and S. Wang, "Machine-learning-based pressure reconstruction with moving boundaries," J. Fluid Mech. **1008**, A21 (2025).
[57]Y. Zhu, W. Chen, J. Deng, and X. Bian, "Physics-informed neural networks for hidden boundary detection and flow field reconstruction," Acta Mech. Sin. **42**(7), 725273 (2026).
[58]E. Kharazmi, D. Fan, Z. Wang, and M. S. Triantafyllou, "Inferring vortex-induced vibrations of flexible cylinders using physics-informed neural networks," J. Fluids Struct. **107**, 103367 (2021).
[59]S. Davey, C. Atkinson, and J. Soria, "Measuring unsteady drag of the flow around a sphere based on time series displacement measurements using physics-informed neural networks," Exp. Therm. Fluid Sci. **144**, 110824 (2023).
[60]G. G. Stokes, "On the effect of the internal friction of fluids on the motion of pendulums," Trans. Cambridge Philos. Soc. **9**, 8–106 (1851).
[61]L. Kulisiewicz and A. Delgado, "High-pressure rheological measurement methods: A review," Appl. Rheol. **20**(1), 13018 (2010).
[62]S. Anggrayni, H. Mubarok, N. P. Putri, N. Suprapto, and A. Kholiq, "The differences in analysing strategy of viscosity experiment between freshmen and laboratory assistant," J. Phys.: Conf. Ser. **997**, 012026 (2018).
[63]K. Kumari, S. Sathiya, and A. Sikander, "A comprehensive review on real-time viscosity and density measuring sensors," Mapan **40**(1), 287–309 (2025).
[64]W. Wang and J. Zhao, "Physics-informed neural networks for the two-way coupling between fluid flow and moving bodies," in 27th Annual Conference of the Hong Kong Society of Theoretical and Applied Mechanics (HKSTAM), Hong Kong, 2024.
[65]S. Wang, Y. Teng, and P. Perdikaris, "Understanding and mitigating gradient flow pathologies in physics-informed neural networks," SIAM J. Sci. Comput. **43**(5), A3055–A3081 (2021).
[66]S. Wang, X. Yu, and P. Perdikaris, "When and why PINNs fail to train: A neural tangent kernel perspective," J. Comput. Phys. **449**, 110768 (2022).
[67]F. M. White and J. Majdalani, *Viscous Fluid Flow* (McGraw-Hill, New York, 2006), Vol. 3.
[68]T. Ota, Y. Okamoto, and H. Yoshikawa, "A correction formula for wall effects on unsteady forces of two-dimensional bluff bodies," J. Fluids Eng. **116**(3), 414–418 (1994).
[69]B. Delacroix, A. Bouarab, L. Fradette, F. Bertrand, and B. Blais, "Simulation of granular flow in a rotating frame of reference using the discrete element method," Powder Technol. **369**, 146–161 (2020).
[70]W. Zhou, S. Miwa, and K. Okamoto, "Advancing fluid dynamics simulations: A comprehensive approach to optimizing physics-informed neural networks," Phys. Fluids **36**(1), 013615 (2024).
[71]C. Wu, M. Zhu, Q. Tan, Y. Kartha, and L. Lu, "A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks," Comput. Methods Appl. Mech. Eng. **403**, 115671 (2023).
[72]L. Barbeau, S. Golshan, J. Deng, S. Étienne, C. Béguin, and B. Blais, "High-order moving immersed boundary and its application to a resolved CFD-DEM model," Comput. Fluids **268**, 106094 (2024).
[73]S. Mishra and R. Molinaro, "Estimates on the generalization error of physics-informed neural networks for approximating PDEs," IMA J. Numer. Anal. **43**(1), 1–43 (2023).
[74]H. von Wahl, T. Richter, S. Frei, and T. Hagemeier, "Falling balls in a viscous fluid with contact: Comparing numerical simulations with experimental data," Phys. Fluids **33**(3), 033304 (2021).
[75]A. Hölzer and M. Sommerfeld, "New simple correlation formula for the drag coefficient of non-spherical particles," Powder Technol. **184**(3), 361–365 (2008).
[76]Z. Xia, K. W. Connington, S. Rapaka, P. Yue, J. J. Feng, and S. Chen, "Flow patterns in the sedimentation of an elliptical particle," J. Fluid Mech. **625**, 249–272 (2009).
[77]A. F. Fortes, D. D. Joseph, and T. S. Lundgren, "Nonlinear mechanics of fluidization of beds of spherical particles," J. Fluid Mech. **177**, 467–483 (1987).
[78]P. Y. Huang, J. Feng, and D. D. Joseph, "The turning couples on an elliptic particle settling in a vertical channel," J. Fluid Mech. **271**, 1–16 (1994).
[79]K. A. K. Namkoong, J. Y. Yoo, and H. G. Choi, "Numerical analysis of two-dimensional motion of a freely falling circular cylinder in an infinite fluid," J. Fluid Mech. **604**, 33–53 (2008).
[80]P.-Y. Chuang and L. A. Barba, "Experience report of physics-informed neural networks in fluid simulations: Pitfalls and frustration," arXiv:2205.14249 (2022).
[81]P.-Y. Chuang and L. A. Barba, "Predictive limitations of physics-informed neural networks in vortex shedding," arXiv:2306.00230 (2023).
[82]F. M. Rohrhofer, S. Posch, C. Gößnitzer, and B. Geiger, "On the role of fixed points of dynamical systems in training physics-informed neural networks," Trans. Mach. Learn. Res. (2023).
[83]S. Wang, S. Sankaran, and P. Perdikaris, "Respecting causality for training physics-informed neural networks," Comput. Methods Appl. Mech. Eng. **421**, 116813 (2024).
[84]A. Jacot, F. Gabriel, and C. Hongler, "Neural tangent kernel: Convergence and generalization in neural networks," in Advances in *Neural Information Processing Systems* (Curran Associates, Inc., 2018), Vol. 31.
[85]A. Krishnapriyan, A. Gholami, S. Zhe, R. Kirby, and M. Mahoney, "Characterizing possible failure modes in physics-informed neural networks," Adv. Neural Inf. Process. Syst. **34**, 26548–26560 (2021).
[86]S. Wang, S. Sankaran, H. Wang, and P. Perdikaris, "An expert's guide to training physics-informed neural networks," arXiv:2308.08468 (2023).
[87]W. Wang, "TwoWayFSI2D," GitHub. https://github.com/wwangdg/TwoWayFSI2D