



HAL
open science

Turbulence model augmented physics-informed neural networks for mean-flow reconstruction

Yusuf Patel, Vincent Mons, Olivier Marquet, Georgios Rigas

► **To cite this version:**

Yusuf Patel, Vincent Mons, Olivier Marquet, Georgios Rigas. Turbulence model augmented physics-informed neural networks for mean-flow reconstruction. *Physical Review Fluids*, 2024, 9, 10.1103/physrevfluids.9.034605 . hal-04567559

HAL Id: hal-04567559

<https://hal.science/hal-04567559v1>

Submitted on 3 May 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Turbulence model augmented physics-informed neural networks for mean-flow reconstruction

Yusuf Patel*

Imperial College London, Department of Aeronautics, London SW7 2AZ, United Kingdom

Vincent Mons[†] and Olivier Marquet[‡]

ONERA, The French Aerospace Lab, 92190 Meudon, France

Georgios Rigas[§]

Imperial College London, Department of Aeronautics, London SW7 2AZ, United Kingdom



(Received 1 June 2023; accepted 13 February 2024; published 11 March 2024)

Experimental measurements and numerical simulations of turbulent flows are characterized by a tradeoff between accuracy and resolution. In this study, we combine accurate sparse pointwise mean velocity measurements with the Reynolds-averaged Navier-Stokes (RANS) equations using data assimilation methods. Importantly, we bridge the gap between data assimilation (DA) using physics-informed neural networks (PINNs) and variational methods based on classical spatial discretization of the flow equations, by comparing both approaches on the same turbulent flow case. First, by constraining the PINN with sparse data and the underdetermined RANS equations without closure, we show that the mean flow is reconstructed to a higher accuracy than a RANS solver using the Spalart-Allmaras (SA) turbulence model. Second, we propose the SA turbulence model augmented PINN (PINN-DA-SA), which outperforms the former approach by up to 73% reduction in mean velocity reconstruction error with coarse measurements. The additional SA physics constraints improve flow reconstructions in regions with high velocity and pressure gradients and separation. Third, we compare the PINN-DA-SA approach to a variational data assimilation using the same sparse velocity measurements and physics constraints. The PINN-DA-SA achieves lower reconstruction error across a range of data resolutions. This is attributed to discretization errors in the variational methodology that are avoided by PINNs. We demonstrate the method using high-fidelity measurements from direct numerical simulation of the turbulent periodic hill at $Re = 5600$.

DOI: [10.1103/PhysRevFluids.9.034605](https://doi.org/10.1103/PhysRevFluids.9.034605)

*yusuf.patel15@imperial.ac.uk

†vincent.mons@onera.fr

‡olivier.marquet@onera.fr

§g.rigas@imperial.ac.uk

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

I. INTRODUCTION

While the Navier-Stokes equations describe accurately the evolution of fluid flow in space and time, their direct numerical simulation (DNS) is intractable in turbulent regimes. As a compromise, industrial simulation is dominated by Reynolds-averaged Navier-Stokes simulations (RANS), which govern the time-averaged flow quantities instead of unsteady time-varying values. However, the RANS equations are not in closed form and solving for these first-order statistics (mean-flow quantities) requires knowledge of the second-order velocity statistics, known as the Reynolds stresses. This closure problem is tackled by use of turbulence models, most commonly Boussinesq linear eddy viscosity models (LEVM), including the Spalart-Allmaras (SA) model [1], $k - \omega$ [2] and $k - \varepsilon$ [3] models. While solutions of the RANS equations are computationally tractable, they are also less accurate than scale-resolving simulations. Although the turbulence models fully close the system of equations for the mean-flow quantities, they typically rely on empirical estimates and parameter tuning, lacking accuracy and generality, as quantified in a review by Xiao *et al.* [4]. However, experimental methods may provide valuable information about real-world flows, which are nevertheless generally limited in terms of spatio-temporal resolution and do not give access to a full flow description. Error from sources, such as imperfect test parts and sensor noise may also impact measurement reliability. Particle Image Velocimetry (PIV) is often used to measure flow velocity but its resolution may be hardware restricted and limited to specific planes of interest (2D planes in a full 3D field), while pressure measurements are difficult to obtain in the bulk flow without being intrusive.

In conjunction with the large growth in the use of data-based methodologies to tackle a wide range of fluid mechanics problems [5,6], data assimilation methods have enabled the augmentation of low-fidelity numerical simulations with experimental data, to reconstruct full and accurate mean-flow descriptions from the latter that obey the RANS equations. Such methods thus allow, among others, to extract information not present in the original dataset, interpolate missing data by super-resolving the flow fields (such as coarse PIV fields), and infer missing fields (such as pressure and Reynolds stresses). Different methods exist for tackling such a reconstruction problem: variational methods [7,8], Bayesian inference [9,10], real-state observer [11,12], and among various machine-learning-based approaches, physics-informed neural networks (PINNs), which have recently been used in conjunction with RANS [13–16]. As further detailed in the following, both variational approaches and PINNs formulate data assimilation as an inverse problem, and more specifically as an optimization one, where one aims to minimize the discrepancies between data and the reconstructed mean flow. However, variational approaches and PINNs vastly differ in the solution method of such an optimization problem, such that one might expect significant differences between their respective results.

As mentioned, the variational approach formulates data assimilation as an optimization problem, where the discrepancies between the reconstructed flow and data are minimized. This minimization is performed in conjunction with the strong imposition of the flow governing equations, here the RANS equations, through the adjoint method. Both the RANS equations and their adjoint counterpart are solved based on classical computational fluid dynamics (CFD) discretization methods, such as the finite-element method that is employed here. Foures *et al.* [17] notably applied variational data assimilation method to super-resolve the mean flow for a laminar 2D circular cylinder. By providing sparse, synthetic experimental data (data that has been generated using high-fidelity computational methods but treated as experimentally generated) on a rectangular grid, the mean flow was successfully super-resolved. Notably, a Helmholtz decomposition is applied to the divergence of the Reynolds stress tensor, separating it into a potential component and a divergence-free component, the latter forming the control vector in the optimization procedure. This work was expanded by Symon *et al.* [18] using the same variational data assimilation method but applied to a turbulent mean flow around an aerofoil at $Re = 13\,500$ using experimental two-component planar PIV data. The work by Franceschini *et al.* [8] extends this work further. Instead of the Helmholtz decomposition, the Reynolds stress is broken down into a eddy viscosity term and a non eddy

viscosity forcing term. The eddy viscosity term is solved by including the SA model to the governing equations, while the variational data assimilation is used to find the non eddy viscosity forcing and also a corrective forcing within the SA equation itself. This is intended to reduce the corrections required in regions where the LEVM is insufficient.

While the present study focuses on flow reconstruction, it may still be worth mentioning that the above-mentioned corrections that are identified by variational data assimilation may then form a dataset to build machine-learning-based turbulence models, as performed in the field-inversion machine-learning approach [19–21].

As an alternative to variational data assimilation, we here also consider the PINN approach first presented by Raissi *et al.* [22]. In conventional neural network approaches, models are optimized by minimizing a cost function, which is purely driven by error to data. The PINN framework enforces weakly the governing equations, by adding the residual PDE errors to the cost function. This is similar to the aforementioned variational approaches, although PINNs only softly constrain the governing equations. Constraining the optimization with fully or partially known governing equations (i.e., Navier Stokes or RANS), in addition to measurement data, ensures that solutions are physical, improving model performance, reduces the possible solution space during optimization and the dependence on high volumes of data to train models. This approach has already shown great promise across a range of inverse flow problems (water wave problem in Jagtap *et al.* [23] and high speed and supersonic flows in Mao *et al.* [24] and Jagtap *et al.* [25]) and can even be extended to convolutional neural networks, as demonstrated in Gao *et al.* [26] and Kelshaw *et al.* [27]. Eivazi *et al.* [14] and Hasanuzzman *et al.* [28] used PINNs that are constrained by the RANS equations to reconstruct the flow field within a subdomain, by providing data at the domain boundaries for all flow fields (from experimental PIV data). Sliwinski and Rigas [29] approached the same problem as Foures *et al.* [17], albeit using the PINN framework instead. First, the Helmholtz decomposition is applied to the unknown divergence of the Reynolds stress tensor (forcing vector), as used in Foures *et al.* [17], to reconstruct the flow from mean velocity data (first-order statistics) alone. Second, by considering the full Reynolds stress tensor and providing both sparse first- and second-order velocity statistical data, the pressure field was also successfully extracted. A similar approach was taken by von Saldern *et al.* [15], using PINNs in combination with velocity data, to reconstruct the mean flow of a swirling turbulent jet. Pioch *et al.* [16] also uses PINNs, constrained by the RANS equations with the addition of turbulence models, to reconstruct the mean flow over a backward facing step. Additional research by Molnar *et al.* [30] propose a new PINN-based process for background-oriented Schlieren imaging (BOS), to reconstruct the density field reconstruction from experimental data, with greater accuracy than traditional BOS image processing algorithms.

One avenue to increase the capability of PINNs is through the development of novel PINN architectures. The PINN framework has been expanded with Conservative PINNs (Jagtap *et al.* [31]) and Extended PINNs (Jagtap and Karniadakis [32]), with further development by Shukla *et al.* [33]. These methods divide the domain into subdomains, each with its own PINN, increasing the representative capacity of the framework, as discussed in Ref. [34]. Additionally, Mishra and Molinaro [35] and De Rycket *et al.* [36] complete error analysis on the use of (extended) PINNs. Alternatively, one can apply advancements in other data-assimilation techniques to the existing PINN framework, to improve on previous studies, as performed here. This study thus aims to demonstrate that similar strategies as proposed in variational approaches may be developed in the PINN framework, to achieve the reconstruction of turbulent mean flows from sparse mean-velocity measurements, extending the laminar PINN framework of Sliwinski and Rigas [29].

We here introduce a turbulence model-augmented PINN (RANS with SA) following an approach that is inspired by the above-mentioned variational data assimilation study by Franceschini *et al.* [8]: the Reynolds-stress term in the RANS equations is separated into a modeled component and a corrective one. The modeled component involves an eddy viscosity that is assumed to verify the SA equation, while the corrective component forms one of the PINN's output, along with the mean velocity, pressure and eddy viscosity fields. Compared to previous studies investigating PINNs in the context of RANS [13–16], the present approach exploits traditional turbulence modeling, adding

physical constraints on the reconstructed mean flow and facilitating the handling of sparse data as demonstrated in the following, while at the same time not being restricted by modeling deficiencies (e.g., noncompliance to the Boussinesq hypothesis) through the introduction of the correction term in the RANS equations. Moreover, the similarities between the present PINN-based methodology and the variational framework of Franceschini *et al.* [8] allow to perform here detailed comparisons between PINN-based and variational data assimilation, which, to the extent of the authors' knowledge, has never yet been performed in the context of RANS and mean-flow reconstruction, while only very few such comparisons for unsteady flow reconstruction may be found in the literature [37].

The following sections will be set out as follows. Section II will begin with presentation of the mean-flow RANS equations and detail the use of the SA turbulence model, which will be used in this work. This is followed by a description of the data assimilation procedure and its implementation using PINNs or a variational approach. Section III introduces the test case and its numerical setup: a turbulent periodic hill at $Re = 5600$. This is followed by a description of the PINN architecture and training procedure in addition to the numerical setup for the variational data assimilation. Section IV will present the data assimilation results of two parts. The first segment presents results of the PINN mean-flow reconstruction, without use of a turbulence model. This will demonstrate the basic PINN's capabilities and its limitations. The second part will present results, showing the use of turbulence model augmentation (SA) with PINNs. These results will be compared first to the baseline PINN (without turbulence model) and then compared with results from the equivalent variational-based method. Section V contains concluding remarks.

II. FLOW EQUATIONS AND DATA ASSIMILATION METHODOLOGY

In this article, the inverse problem is solved, where a mean (time-averaged) flow field is reconstructed from a set of partial observations. The observations are N_m sparse high-fidelity mean velocity measurements at points $\mathbf{x}_m = (x_m, y_m, z_m)$, i.e., $U_{m,i}$, where U_i is the i th component of the mean velocity and the subscript m denotes the m th sparse measurement. The reconstructed flow field is sought by solving the constrained optimization problem, such that error to measurements, J , is minimized, where

$$J = \sum_{m=1}^{N_m} \sum_{i=1}^3 [U_{m,i} - \hat{U}_i(\mathbf{x}_m)]^2, \quad (1)$$

while simultaneously the reconstructed field satisfies the underdetermined mean-flow governing equations. Here, $\hat{U}_i(\mathbf{x}_m)$ is the reconstructed mean velocity at \mathbf{x}_m . While the equations in this section are presented for general flows, the results demonstrate use for a two dimensional flow.

First, the governing mean-flow equations without (baseline) and with (SA) turbulence model are presented in Sec. II A. Second, in Sec. II B, the details of two data assimilation approaches used to solve the optimization are described—an adjoint-based variational approach and a PINN approach.

A. Formulation of the Reynolds stress closure

By applying the Reynolds decomposition, an unsteady flow field can be separated into a mean and fluctuating component. Through time-averaging of the Navier-Stokes equations, one can obtain the steady RANS equations. The data-driven techniques used in this study will be applied to mean-flow data, which satisfy the following RANS equations:

$$\frac{\partial U_i}{\partial x_i} = 0, \quad (2a)$$

$$U_j \frac{\partial U_i}{\partial x_j} + \frac{1}{\rho} \frac{\partial P}{\partial x_i} - \frac{\partial (2\nu S_{ij})}{\partial x_j} + \frac{\partial u'_i u'_j}{\partial x_j} = 0, \quad (2b)$$

where P is the mean pressure field and $\overline{u_i' u_j'}$ are the Reynolds stress tensor terms. S_{ij} is the mean strain rate tensor equivalent to $\frac{1}{2}(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i})$. For a three-dimensional flow (3D), this results in a system of four equations with 10 unknowns (assuming no spanwise mean flow, this can be reduced down to three equations and 6 unknowns). Given the dependence of the mean-flow solution, (U_i, P) , on the terms of the unknown Reynolds stress tensor, $\overline{u_i' u_j'}$, determination of these terms is critical to the performance and capability of data-assimilation methods for reconstructing the mean-flow field. The remainder of Sec. II A will present formulations of the mean-flow equations, such that data can be used to infer the Reynolds stress (closure) terms.

1. Baseline formulation of mean-flow equations

To solve the inverse problem, an additional constraint is placed on the closure term to improve its inference. First, the divergence of the Reynolds stress tensor [as it appears in the RANS Eq. (2b)] is considered as a forcing vector (Reynolds forcing vector). This reduces the closure term from six individual Reynolds stresses to three individual forcing terms (or two forcing terms from three stresses with no spanwise mean flow). Subsequently, a Helmholtz decomposition is applied, as in Foures *et al.* [17] and Sliwinski and Rigas [29],

$$f_i \equiv -\frac{\partial \overline{u_i' u_j'}}{\partial x_j} = \frac{1}{\rho} \frac{\partial \phi}{\partial x_i} + f_{s,i}, \quad (3)$$

decomposing the forcing into a scalar, potential part (ϕ) and a divergence-free, solenoidal, vectorial component ($f_{s,i}$). This latter condition (divergence-free) provides an additional equation. Introducing decomposition (3) into the RANS equations (2), gives

$$\frac{\partial U_i}{\partial x_i} = 0, \quad (4a)$$

$$U_j \frac{\partial U_i}{\partial x_j} + \frac{1}{\rho} \frac{\partial (P - \phi)}{\partial x_i} - \frac{\partial (2\nu S_{ij})}{\partial x_j} - f_{s,i} = 0, \quad (4b)$$

$$\frac{\partial f_{s,i}}{\partial x_i} = 0. \quad (4c)$$

In addition to the new equation, the Reynolds forcing vector now consists of four terms (three in two dimensions), in the form of a solenoidal forcing vector and a scalar potential field. In this system, there are now five equations and eight unknowns (four equations and six unknowns in 2D). By combining the pressure and potential terms into a single scalar field ($P - \phi$), another unknown can be removed. The resultant flow field, $(U_i, P - \phi, f_{s,i})$ is composed of the mean velocity components, combined pressure and potential forcing term and the solenoidal forcing components. Providing mean velocity data alone, at discrete locations, is sufficient to close the system at those points. However, the $P - \phi$ term can not be separated into its constituent parts and the pressure field cannot be determined.

2. Turbulence-model augmentation of mean-flow equations

While the above approach combines data with a formulation of the governing equations to infer the unclosed divergence of the Reynolds stress tensor, the closure problem still poses a challenge away from data points, since an infinite number of potentially physically unrealizable solutions satisfy the underdetermined governing equations. The following methodology is proposed to augment the aforementioned approach, by adding additional physical constraints.

When performing numerical simulations of the RANS equations, turbulence modeling is utilized to approximate the Reynolds stresses and close the equations. Typically, the Boussinesq approximation is used, where the Reynolds stresses are assumed to be isotropic and depend on the eddy viscosity ν_t . However, neglecting the anisotropic component, as well as other modeling assumptions

made in the calculation of v_t , introduce errors which propagate to the mean-flow solution. Instead, as applied in Franceschini *et al.* [8], one can decompose the Reynolds forcing into a modeled, isotropic part and a corrective component which can include the anisotropic part, in addition to any isotropic component not captured by the modeled part, such that

$$f_i = \frac{\partial(2v_t S_{ij})}{\partial x_j} + f_{c,i} = \frac{\partial(2v_t S_{ij})}{\partial x_j} + \underbrace{\frac{1}{\rho} \frac{\partial \phi}{\partial x_i}}_{f_{c,i}} + f_{s,i}. \quad (5)$$

The first term represents the modeled forcing component and the second term the corrective forcing component, such that the mean-flow solution matches the data (high-fidelity measurements). In this work, the Helmholtz decomposition is also applied to $f_{c,i}$. By augmenting the system with a turbulence model, one can solve for v_t . This work will use the one-equation Spalart-Allmaras turbulence model, which provides a governing equation for a pseudo eddy-viscosity variable \tilde{v} . The actual eddy viscosity v_t may then be obtained from \tilde{v} through an algebraic expression. The SA equations are given in Appendix A 1. The decomposition (5) has a closed (modeled) part, v_t , and unclosed (corrective) part, $f_{c,i}$. Equation (5) can then be substituted into Eq. (2b) to derive the final SA augmented RANS equations,

$$\frac{\partial U_i}{\partial x_i} = 0, \quad (6a)$$

$$U_j \frac{\partial U_i}{\partial x_j} + \frac{1}{\rho} \frac{\partial(P - \phi)}{\partial x_i} - \frac{\partial[2(v + v_t)S_{ij}]}{\partial x_j} - f_{s,i} = 0, \quad (6b)$$

$$\frac{\partial f_{s,i}}{\partial x_i} = 0, \quad (6c)$$

$$U_j \frac{\partial \tilde{v}}{\partial x_j} - S_p - S_{\text{diff}} - S_c - S_d = 0. \quad (6d)$$

Applying the turbulence model allows an isotropic component of Reynolds stresses to be approximated using the SA equation, leaving the anisotropic (and potentially an isotropic) component underdetermined. As a large portion of the Reynolds forcing is now modeled, data-assimilation techniques only need to apply smaller corrections, compared with Eq. (4), to close the solution.

B. Data assimilation techniques

Using data assimilation techniques, to solve the constrained optimization, a solution $\hat{U}_i(\mathbf{x}), \hat{P}(\mathbf{x}) - \hat{\phi}(\mathbf{x}), \hat{f}_{s,i}(\mathbf{x}), \hat{v}(\mathbf{x})$ to the mean-flow equations (4) or (6) is sought, such that J , the measurement error (1), is minimized. In the specific case where turbulence model augmentation is not applied, as in Eq. (4), the eddy viscosity field $\tilde{v}(\mathbf{x}) = 0$. Two techniques are implemented to solve the constrained optimization problem: a neural network approach, specifically the PINN approach from Sliwinski and Rigas [29] and an adjoint-based variational approach, used by Foures *et al.* [17] and Franceschini *et al.* [8].

1. Physics-informed neural network

For a PINN, the solution $(\hat{U}_i, \hat{P} - \hat{\phi}, \hat{f}_{s,i}, \hat{v})(\mathbf{x}; \theta)$ is defined by weights and biases, θ , which define a neural network. Figure 1 shows this PINN schematic. For any neural network, the optimal set of weights and biases, $\hat{\theta}$, is sought by minimizing some loss function C ,

$$\hat{\theta} = \arg \min_{\theta} C(U_i, P - \phi, f_{s,i}, \tilde{v})(\mathbf{x}; \theta). \quad (7)$$

In the case of PINNs, the loss function, C , is used to enforce the governing equations and the boundary conditions. While these two terms are sufficient to define an optimization for the forward problem, when using data assimilation (the inverse problem) the cost function must also include

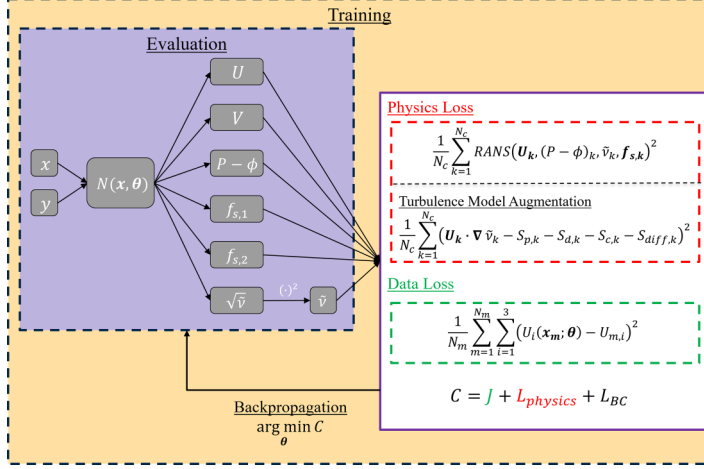


FIG. 1. Structure of the physics-informed neural network. A continuous mapping between spatial coordinates and flow variables is approximated by a deep neural network, denoted by N , defined by a set of weights and biases, θ . High-fidelity coarse data measurements and the (turbulence model augmented) RANS equations constrain the training of the network. k, m subscript the physics and data loss, respectively. i represents the i th component of a vector.

the measurement error term, J . To distinguish the use of PINNs for the inverse problem (as in this paper) compared to the forward problem, it will be referred to as PINN-DA hereon-in to emphasise the use of data. For the mean-flow problem described in Sec. II A, the loss function can be defined as

$$\begin{aligned}
 C = & \frac{\lambda^D}{N_m} J + \frac{\lambda^P}{N_c} \sum_{k=1}^{N_c} \text{RANS}[(U_i, P - \phi, f_{s,i}, \tilde{v})(\mathbf{x}_k; \theta)]^2 + \frac{\lambda^P}{N_c} \sum_{k=1}^{N_c} \left(U_j \frac{\partial \tilde{v}}{\partial x_j} - S_p - S_{\text{diff}} - S_c - S_d \right) \\
 & \times (\mathbf{x}_k; \theta)^2 + \frac{\lambda^B}{N_b} \sum_{b=1}^{N_b} \text{BC}[(U_i, P - \phi, f_{s,i}, \tilde{v})(\mathbf{x}_b; \theta)]^2 + \frac{\lambda^R}{2} \frac{1}{N_c} \sum_{k=1}^{N_c} \sum_{i=1}^3 f_{s,i}(\mathbf{x}_k; \theta)^2, \quad (8)
 \end{aligned}$$

where the governing laws are evaluated at N_c collocation points (subscripted by k), and boundary conditions are imposed at N_b points along the boundary (subscripted by b). The data loss is subscripted by m . RANS is the residual of the mean-flow equations [(4) or (6)], and BC is the error to the imposed boundary conditions. To evaluate the governing laws, the gradients in the RANS equations are calculated accurately to machine-precision using automatic differentiation [leveraging the chain rule to trace the derivatives of all the constituent operations in the mapping $N(\mathbf{x}, \theta)$ from Fig. 1]. The data term, physics term, and boundary condition terms are weighted by factors λ^D , λ^P , and λ^B , respectively. Selection of these weights is discussed in Sec. IV A.

The decomposition of the forcing into a modeled and corrective component (5) is not unique, as discussed in Foures *et al.* [17]. As a result, additional regularization is applied to ensure uniqueness of the distribution between modeled and corrective forcing during optimization. For the PINN-DA, an L_2 regularization is used for its effectiveness and ease of implementation. This regularization term (also evaluated at collocation points), weighted by λ^R , controls the magnitude of the corrective forcing field and subsequently the decomposition between modeled and corrective forcing. By increasing the penalization of forcing magnitude, the importance of the modeled forcing component will grow and the governing equations will tend towards the RANS-SA equations (no corrective forcing). The effect of regularization and importance of λ^R is demonstrated in Appendix B 1. In the case without turbulence-model augmentation [effectively $\tilde{v}(\mathbf{x}) = 0$], $\lambda^R = 0$ and the SA

equation can be removed from the loss function. The total loss C function (7) is minimized using backpropagation by iteratively adjusting the weights and biases based on the gradient of loss with respect to the network parameters, $\partial C/\partial\theta$.

2. Variational assimilation

In the variational approach, the corrective forcing $f_{s,i}$ is adjusted such that the measurement error, J , is minimized, constrained by the RANS equations (6). The solution to this minimization problem is given by the stationary points of the Lagrangian

$$\begin{aligned} \mathcal{L}([U_i, P - \phi, \tilde{v}, f_{s,i}], [U_i^*, P^*, \tilde{v}^*]) \\ = J - \left\langle P^*, \frac{\partial U_i}{\partial x_i} \right\rangle - \left\langle U_i^*, U_j \frac{\partial U_i}{\partial x_j} + \frac{1}{\rho} \frac{\partial (P - \phi)}{\partial x_i} - \frac{\partial [2(\nu + \nu_t)S_{ij}]}{\partial x_j} - f_{s,i} \right\rangle \\ - \left\langle \tilde{v}^*, U_j \frac{\partial \tilde{v}}{\partial x_j} - S_p - S_{\text{diff}} - S_c - S_d \right\rangle. \end{aligned} \quad (9)$$

The inner product is defined as

$$\langle \mathbf{a}, \mathbf{b} \rangle = \int_{\Omega} \mathbf{a} \cdot \mathbf{b} d\Omega, \quad (10)$$

where \mathbf{a}, \mathbf{b} are two spatial fields and Ω is the volume over which the field is defined. U_i^*, P^*, \tilde{v}^* are the Lagrange multipliers (or adjoint variables) enforcing the constraints. By setting the variation of the Lagrangian with respect to the direct flow variables $U_i, P - \phi, \tilde{v}$, one can derive the adjoint equations

$$\frac{\partial U_i^*}{\partial x_i} = 0, \quad (11a)$$

$$U_j^* \frac{\partial U_j}{\partial x_i} - U_j \frac{\partial U_i^*}{\partial x_j} - \frac{\partial P^*}{\partial x_i} - \frac{\partial [2(\nu + \nu_t)S_{ij}^*]}{\partial x_j} + \tilde{v}^* \frac{\partial \tilde{v}}{\partial x_i} + \frac{\partial}{\partial x_j} [\tilde{v}^* \partial_{\nabla U_i} (S_p + S_d)] = \frac{\partial J}{\partial U_i}, \quad (11b)$$

$$-U_j \frac{\partial \tilde{v}^*}{\partial x_j} + 2 \frac{\partial \nu_t}{\partial \tilde{v}} \frac{\partial U_i^*}{\partial x_j} S_{ij} - \tilde{v}^* \frac{\partial (S_p + S_d)}{\partial \tilde{v}} + \frac{\partial}{\partial x_j} [\tilde{v}^* \partial_{\nabla \tilde{v}} (S_c + S_{diff})] = 0. \quad (11c)$$

Furthermore, by calculating the gradient of Lagrangian, \mathcal{L} , with respect to changes in forcing, $\frac{\partial \mathcal{L}}{\partial f_{s,i}}$, one finds $\frac{\partial \mathcal{L}}{\partial f_{s,i}} = \frac{dJ}{df_{s,i}} = U_i^*$. Specifically, this means the gradient $\frac{dJ}{df_{s,i}}$ used to set the minimization direction for optimization is the solution to the adjoint equations (11).

As with the PINN-DA-SA, regularization is required to ensure uniqueness of the corrective forcing. Furthermore, the measurements are sparse, inducing a pointwise forcing of the adjoint momentum equations (11b) according to $\frac{\partial J}{\partial U_i} = 2 \sum_{m=1}^{N_m} \delta(\mathbf{x} - \mathbf{x}_m) [U_i(\mathbf{x}_m) - U_{m,i}]$, which may lead to discontinuities in the adjoint field and in the gradient $\frac{dJ}{df_{s,i}}$, and thus ultimately in the reconstructed forcing and corresponding mean-flow solution. This can be circumvented through H_1 -like regularization of the gradient, which consists in getting a smoothed gradient $\frac{dJ^R}{df_s}$ from the original gradient $\frac{dJ}{df_s}$ through the inversion of the following system:

$$\begin{pmatrix} \frac{1}{1+\beta} (I - \beta \Delta) \circ & \nabla \circ \\ -\nabla \cdot \circ & 0 \end{pmatrix} \begin{pmatrix} \frac{dJ^R}{df_s} \\ \pi \end{pmatrix} = \begin{pmatrix} \frac{dJ}{df_s} \\ 0 \end{pmatrix}, \quad (12)$$

where $\beta = l_r^2$ and l_r may be interpreted as a filter length. In the following, l_r is chosen as the spacing between measurement locations. Equation (12) also involves a pseudopressure field π which is introduced to preserve the divergence-free character of the gradient through the regularization procedure. A complete derivation and details of the variational approach can be found in Refs. [8,17,38].

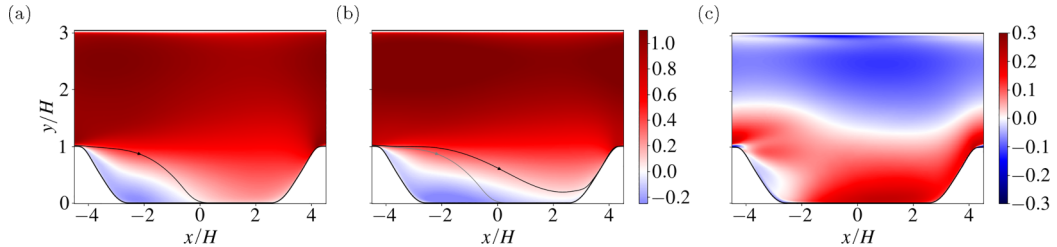


FIG. 2. High-fidelity (DNS) solution of the flow over a periodic hill compared with the RANS solution using an SA model. The dividing streamline between the recirculation region and rest of the flow is shown. (a) Streamwise mean velocity—DNS; (b) streamwise mean velocity—RANS; (c) absolute error of streamwise velocity relative to DNS solution ($U_{\text{err}} = U_{\text{DNS}} - U_{\text{RANS}}$). For comparison, the DNS dividing streamline is shown in gray.

It has to be noted that for high Reynolds numbers turbulent cases, the variational-based data assimilation requires the use of the SA turbulence model augmentation to numerically solve the discretized equations, unlike the PINNs which work (albeit with different accuracy) with and without the SA model. Previously, the laminar studies in Ref. [17] have not required this augmentation. To distinguish this approach it shall be referred to as variational-DA-SA.

3. Comparison: PINNs versus variational assimilation

Equations (8) and (9) highlight the similarities as well as key differences between the variational-DA-SA and PINN-DA approaches. First, while the variational-DA-SA directly adjusts the corrective forcing, through the gradient $dJ/df_{s,i}$, the PINN-DA indirectly adjusts this forcing through adjustment of the network parameters, θ . Second, by the nature of PINNs, a continuous mapping of the flow solution is obtained, which can be queried at any spatial location. Additionally, the governing equations are evaluated and enforced at discrete locations within the domain (collocation points) and the spatial gradients in this mesh-free approach are determined through automatic differentiation. In the variational-DA-SA approach, solution of the direct and adjoint equations introduces mesh dependency, where the conservation laws are enforced across the entire domain, while spatial gradients are calculated through discretization (here finite element method). As a result, discretization errors are introduced, which are not present in PINNs. PINNs softly constrain the conservation laws in their continuous form while these are hard constraints in their discrete form for the variational-DA-SA. Last, the forcing regularization method applied to the two approaches are different. The PINN-DA uses an L_2 -based regularization which was found to be sufficient to produce a smooth forcing solution. However, for the variational-DA-SA, an L_2 approach with pointwise data enforcement leads to unsmooth, pointwise forcing, thus the use of a H_1 -based method, as discussed in Sec. II B 2. Appendix B contains a detailed breakdown on the effect of regularization on PINNs and the effect of different regularisers for the variational-DA-SA approach.

III. NUMERICAL SIMULATIONS AND MODEL SETUP

The PINN and variational approaches described above are applied to a canonical turbulent test case, the flow over a periodic hill. This section will describe the details of the flow case and the numerical setup of the two methods employed.

A. Periodic hill setup

The high-fidelity data for the periodic hill is obtained from the DNS database found in Ref. [39] using the Incompact3D solver [40] and the simulation setup is detailed in Xiao *et al.* [41]. The DNS mean-flow (time-averaged) solution, for streamwise velocity U is shown in Fig. 2(a). The

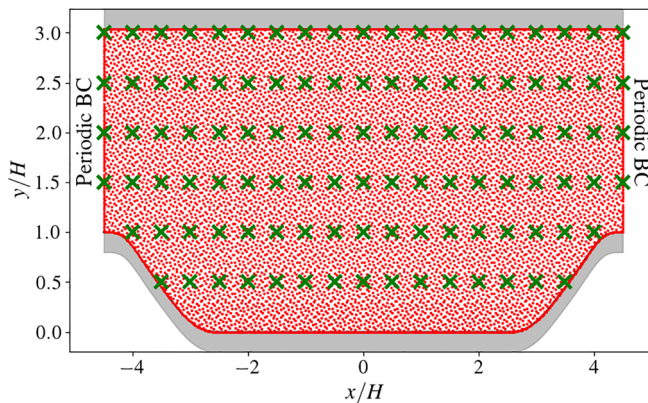


FIG. 3. Turbulent periodic hill domain used for data assimilation, with $\Delta L = 0.5$ data resolution (green crosses) used to compare the variational and PINN approaches. The collocation points, where the PINN evaluates residual error, are in red and were sampled across the domain using a Hammersley distribution.

streamline (solution to the streamfunction) dividing the recirculation region from the rest of the flow is also shown. The periodic hill is simulated at $\text{Re} = U_b H / \nu = 5600$, where H is the hill height, U_b is the bulk velocity over the hill apex and ν the kinematic viscosity. Of the several geometric configurations in the database, this work uses the canonical configuration with the domain length given by $L_x/H = 3.858\alpha + 5.142$ for which the hill stretch factor is $\alpha = 1.0$. At this Reynolds number, separation, recirculation and reattachment are present. The no-slip condition is enforced along the upper and lower walls, while periodicity is enforced in the streamwise direction. To maintain the required massflow and thus Reynolds number, body forcing is applied in the periodic streamwise direction. This is determined at each iteration, such that the Reynolds number matches the preset Reynolds number.

To analyze the effect of data resolution on mean-flow reconstruction accuracy, the high-fidelity DNS data is spatially sampled on a square grid at nine data resolutions— $0.05 \leq \Delta L \leq 1.0$, where $\Delta L = \Delta x/H = \Delta y/H$. All results and figures from here on will be compared to the results from the $\Delta L = 0.5$ resolution, unless otherwise stated. The measurement locations are shown in Fig. 3. The choice of a sparse square sampling grid represents a velocity field as measured using methods such as laser Doppler velocimetry or PIV.

The low-fidelity RANS solution with the SA turbulence model is shown in Fig. 2(b). While the DNS reattachment occurs at $x/H = 0$, the RANS solver massively overpredicts the size of the recirculation region, as shown. The error field likewise, in Fig. 2(c), shows high error across the entire domain.

B. PINN setup

The PINNs are implemented using the DeepXDE Python library by Lu *et al.* [42], as in Ref. [29]. This toolbox simplifies the process of building PINNs (with the Raissi *et al.* [22] paradigm). The code and data to run the PINN cases from this paper are available at Ref. [43]. All PINNs use the same architecture. The neural network was constructed as a multilayer perceptron with seven fully connected hidden layers, each with 50 nodes. The input layer consists of two nodes, representing spatial coordinates, x/H and y/H . The output layer consists of varying number of nodes depending on the test case. The PINN initially contains five output nodes with two mean velocity components, the combined mean pressure and potential forcing field, $P - \phi$, and two solenoidal forcing nodes. When the SA turbulence model augmentation was used, an additional output node is required for \bar{v} . Each node used a tanh activation function for its smoothness and its second-order differentiability. This architecture was deemed to perform well across a range of data resolutions. A grid search

method was employed to find optimal hyperparameters (number of nodes and layers, activation function and learning rate). The effect of hyperparameters on reconstruction accuracy are included in Appendix C. For PINNs, the weights and biases defining the network are randomly initialized using the Glorot uniform algorithm. This contrasts the variational-DA-SA approach, in which an “initial guess” (the RANS-SA solution) was used, around which the system is initially linearized.

For PINN optimization, collocation points must be specified, to define where the governing laws are evaluated. In this work, the PINNs use 10 000 collocations points which are distributed using a Hammersely distribution, such that there are sufficient points across all regions of the domain. This number of collocation points for the PINNs was selected after the reconstruction performance showed little sensitivity to further increases of the number of points. An additional 1000 points were distributed to the walls and periodic boundaries to enforce the boundary conditions. The collocation points (and domain) are shown in Fig. 3. As in Ref. [29], a two-step optimization was used for the PINNs. Initially an ADAM optimization is applied, followed by an L-BFGS-B phase. L-BFGS-B is a quasi-Newton method and thus its convergence depends on the initial guess. By using an ADAM optimiser first, one can reach an acceptable loss level to provide a good initial guess for the L-BFGS-B optimiser. Details on PINN convergence are discussed further in Sec. IV A. The PINNs were trained on one NVIDIA RTX 6000 GPU.

C. Modification to PINNs for Spalart-Allmaras augmentation

Using PINNs in combination with the SA turbulence model introduces additional complexity, which require modifications to the neural network architecture and the problem formulation to achieve convergence.

First, the SA transport equation contains $\frac{1}{d^2}$ terms in the production and destruction terms, where d is the true wall distance, resulting in singularities at the wall, where $d = 0$. This is not a problem for the finite element method (RANS and variational) where the residual is evaluated at a cell center, which is not on the wall. For the PINN paradigm used, however, the governing equations are evaluated exactly at the wall boundaries and so this leads to singularities and subsequently failure to convergence. Consequently, the equation was reformulated by multiplying the SA equation with d^2 to eliminate the singularity. This new formulation can be found in Appendix A 2. Problems arising from singularities when used with PINNs have also been mentioned in Ref. [30].

The second change is to eliminate the effect of clipping negative values of $\tilde{\nu}$. The negative SA equation formulation used in the variational approach has distinct equations for positive and negative $\tilde{\nu}$ (detailed in Appendix A 1) to ensure the transport equation is continuous across both positive and negative $\tilde{\nu}$. The final eddy viscosity, ν_t , is then clipped at negative values of $\tilde{\nu}$ and set to zero, resulting in only positive values of ν_t . Convergence of the PINNs is improved when the complexity caused by the discontinuity from clipping ν_t is removed. Thus, to enforce positive $\tilde{\nu}$, the output of the neural network is defined as $\sqrt{\tilde{\nu}}$. A transformation is then applied to the output, squaring the value, to produce a positive $\tilde{\nu}$. This is shown in Fig. 1.

D. Variational assimilation setup

The solutions of the RANS-SA system (6) and its adjoint (11) are obtained through a finite-element method (FEM) spatial discretization implemented in FreeFEM++ [44]. Contrary to the presentation in Sec. II B 2, the adjoint model is obtained following a discrete adjoint approach, where the Jacobian matrix that is associated to the RANS-SA equations is transposed. This Jacobian matrix is also used to invert the nonlinear RANS-SA equations based on a Newton method. Piecewise-linear functions that are enriched by bubble functions are used for velocity and pseudoturbulent viscosity variables, while piecewise-linear functions are used for pressure. The FEM is known to be unstable at high Reynolds numbers. Accordingly, both streamline-upwind Petrov-Galerkin (SUPG) [8,45] and grad-div [46] formulations are here employed to stabilise the method. A low-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) [47] algorithm is used

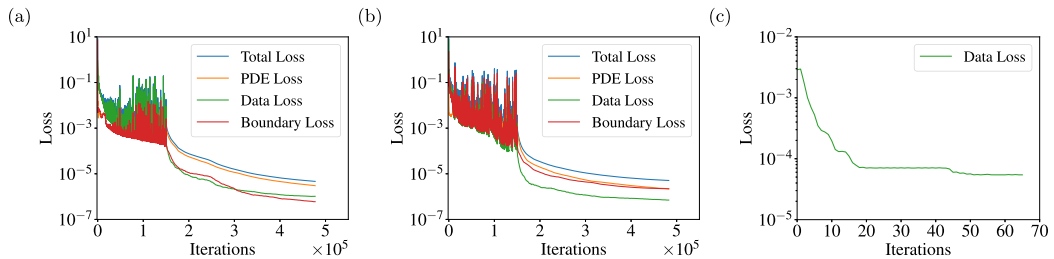


FIG. 4. Residual loss during optimization for (a) PINN-DA-Baseline, (b) PINN-DA-SA, and (c) Variational-DA-SA. For PINN optimization (training), there are two distinct phases: the ADAM (stochastic gradient descent based) optimization for 150 000 iterations followed by the L-BFGS-B phase.

to exploit the gradient $\frac{dJ^R}{d\mathbf{f}_s}$ from Eq. (12) to perform the minimization of the cost function J in Eq. (1). As mentioned above, the first-guess for the optimization procedure is $\mathbf{f}_s = \mathbf{0}$, i.e., the baseline solution of the RANS-SA equations. The full optimization procedure was carried out on 28 cores Intel Xeon Broadwell E5-2680v4, 2.4 GHz. The total number of degrees of freedom in the RANS and adjoint simulations is around 2×10^5 . Further details on the implementation of the variational-DA-SA can be found in Refs. [8,38].

IV. RESULTS

We compare the mean-flow reconstruction from sparse mean velocity measurements for the turbulent periodic hill case using three approaches: PINN-DA-Baseline (without turbulence model), PINN-DA-SA (with SA), and variational-DA-SA (with SA). First, an evaluation of the PINN-DA-Baseline approach is presented. Second, we will show that significant reduction in reconstruction error is achieved by augmenting the PINN-DA-Baseline approach with the SA turbulence model (PINN-DA-SA). Third, we compare it directly to the variational-DA-SA showing the increased accuracy of the PINN-DA-SA relative to the variational-DA-SA.

To evaluate the mean-flow reconstruction accuracy, a volume weighted L_2 error will be used and is calculated as

$$\varepsilon_2 = \sqrt{\frac{1}{\Omega} \sum_{n=1}^N \delta\Omega_n \sum_{i=1}^2 [U_{n,i} - \hat{U}_i(\mathbf{x}_n)]^2}, \quad (13)$$

where Ω is the volume of the domain, $\delta\Omega_n$ is the cell volume of each point used to determine the error (as defined by the DNS grid) and N is the number of points across the domain. $U_{n,i}$ is the high-fidelity, DNS mean velocity at \mathbf{x}_n and $\hat{U}_i(\mathbf{x}_n)$ is the corresponding reconstructed mean velocity. The variational solution is interpolated onto the DNS grid using a linear interpolation method. Additionally, the employed finite-element mesh corresponds to typical mesh sizes that are close to those in DNS.

A. Optimization

Figure 4 shows the optimization convergence of the PINN-DAs (a,b) and the variational-DA-SA approach (c). The PINN-DA loss convergence curve is decomposed into three loss components—PDE residual, data error and boundary condition error. For PINN-DA-SA the additional loss from the SA transport equation contributes also to the PDE loss. Each individual loss component appearing in Eq. (8) was given initial weight values λ based on the relative magnitudes of the flow variables. Thereafter, the weights were manually adjusted to find the combination with the lowest loss, as compiled in Table I. The weight sensitivities are presented in Appendix C. During the first phase of the training, ADAM optimization is performed for 150 000 iterations. The second

TABLE I. Loss function weights used in PINN optimization. For PINN-DA-Baseline, the $\tilde{\nu}$ and λ^R weights can be ignored. Additionally, this regularization weight is only valid for the $\Delta L = 0.5$ data resolution. For other resolutions, see Fig. 13.

Weight	λ^P	λ^D	$\lambda^B(U_i)$ -Wall	$\lambda^B(\tilde{\nu}, f_{s,i})$ -Wall	λ^B -Periodic	λ^R
Value	1	10	2.5	10	1	0.01

phase applies L-BFGS-B optimization for 300 000 iterations or until the model has sufficiently converged to the floating point tolerance. Figures 4(a) and 4(b) shows that during optimization (for both PINN-DA methodologies), the ADAM phase reduces loss from the high value caused by the initial randomized state to approximately 10^{-3} . During the ADAM optimization phase, one can observe frequent jumps in the loss. This is an effect of the adaptive step size used in ADAM. The step size is inversely proportional to the exponential average of squared (loss) gradient (as described in Ref. [48]). As the loss flattens out, the adaptive step size becomes very large, causing the model to “jump” too far to a higher loss level. The L-BFGS-B phase is then used to fine tune the optimization, reducing total loss to below 10^{-5} .

The convergence of the variational-DA-SA, seen in Fig. 4(c) shows the data loss, corresponding to J/N_m , where J is defined in Eq. (1). This data loss is equivalent to the corresponding curves in Figs. 4(a) and 4(b). While convergent, the data loss is an order of magnitude higher for the variational-DA-SA compared with the PINN-DAs. While there is a vast difference in number of iterations between the PINN and variational approaches, an iteration is not equivalent across both methods. For PINNs, a single iteration involves evaluating the PINN at all collocation and data points, calculating the cost function and back-propagating the loss to update the model weights (and thus indirectly the flow solution). For the variational approach, in each iteration the direct and adjoint equations are solved after which the forcing solution is directly updated. An average PINN iteration takes approximately 0.84 s/iteration, while for the variational approach this is approximately 55 s/iteration. This difference in computational cost can be attributed to computational complexity, more mesh points in the variational method (than PINN collocation points), reduced number of degrees of freedom in the neural network (updating only weights and biases instead of the actual solution across the domain) and hardware.

B. PINN-DA-Baseline

Here, the PINN-DA-Baseline approach is applied to reconstruct the mean-flow velocity field when coarse mean-flow velocity data is available with spacing $\Delta L = 0.5$. This is compared against the DNS and RANS-SA velocity profiles in Fig. 5. The data assimilation of the high-fidelity measurements has improved the prediction of the velocity field compared to RANS-SA. The total

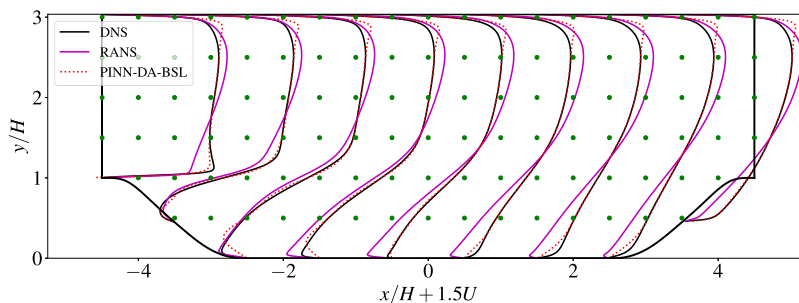


FIG. 5. Streamwise velocity profiles for DNS, RANS-SA and PINN-DA-Baseline.

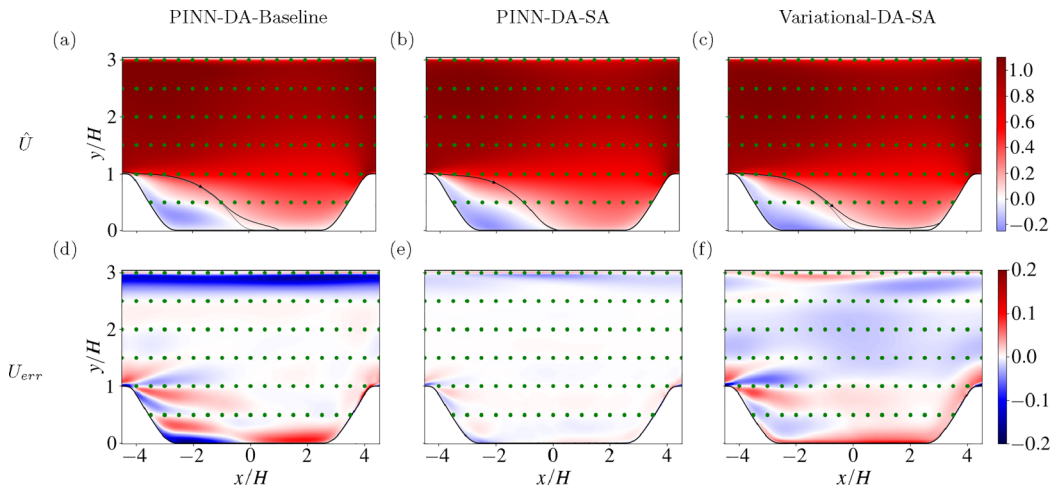


FIG. 6. Reconstructed U velocity component for the PINN-DA-Baseline (left), PINN-DA-SA (middle), variational-DA-SA (right) approaches. (a)–(c) U reconstruction. (d)–(f) Absolute U error. PINN-DA-SA performs best, significantly reducing the separation error seen in other approaches, while also reducing the near wall error seen in PINN-DA-Baseline. The diving streamline (separating the recirculation region) is shown in panels (a)–(c) and compared with the DNS equivalent (in gray). While the PINN-DA-SA recirculation region is indistinguishable, there is a small but noticeable difference for the PINN-DA-Baseline and variational-DA-SA solutions.

mean velocity L_2 error is 3.60×10^{-2} instead of the 8.99×10^{-2} error from RANS-SA. High accuracy of reconstruction is observed in the bulk flow domain. However, high mean velocity reconstruction error is observed in areas with high velocity gradients, such as near the walls and at separation. This is most apparent at the upper wall boundary layer. This near wall error accounts for over 40% of the total L_2 error as the PINN-DA-Baseline fails to accurately capture high-gradient flow features. These trends are highlighted in the streamwise velocity contours in Figs. 6(a) and 6(d). At data points, the PINN-DA-Baseline reproduces the measurements accurately. Furthermore, in spite of the higher near-wall reconstruction errors, Fig. 6(a) shows improved reconstruction of the recirculation bubble, even with limited data in this area. This is most noticeable when comparing the PINN-DA-Baseline bubble from Fig. 6(a) with the RANS bubble in Fig. 2(b). The provision of sparse high-fidelity datapoints in the PINN-DA-Baseline approach has enabled better prediction of the shape of the recirculation region. However, the error field makes the limitation of the PINN-DA-Baseline approach more apparent, with the near wall errors particularly noticeable.

The results of PINN-DA-Baseline show that the methodology used in Ref. [29] for laminar flow can also be used for turbulent flows. In Ref. [29], the base formulation of the RANS equations [as in Eq. (2)] was also tested. It was shown that by providing data for both first-order (mean velocity) and second-order (Reynolds stresses) statistics, the PINN-DA-Baseline was able to reconstruct the pressure field for the laminar case. This approach has been demonstrated in Appendix D. With similar mean velocity reconstruction accuracy, this second approach was also able to infer the pressure field for this turbulent case.

C. PINN-DA-SA

This section contains results from the augmentation of the SA model to PINNs. First, a comparison between the PINN-DA-Baseline and PINN-DA-SA results will be presented, followed by a discussion on the improved reconstruction when the SA model is used. Finally, the results of the variational data assimilation using SA (variational-DA-SA) are also presented, followed by a parametric analysis of the data resolution.

1. PINN-DA-SA versus PINN-DA-Baseline

Figures 6(a) and 6(d) show the PINN-DA-Baseline and Figs. 6(b) and 6(e) the PINN-DA-SA reconstruction results. Streamwise velocity and error contours relative to the DNS solution are shown, with similar trends holding for the vertical velocity component (not shown here). While the error field topologies are qualitatively similar, the magnitude error is significantly reduced. For PINN-DA-SA, the recirculation bubble is almost indistinguishable from the DNS one. As a consequence of introducing the Spalart-Allmaras model, reconstruction error has reduced by 63% ($\varepsilon_2 = 1.35 \times 10^{-2}$). For the bulk flow region above the hills and away from the walls (between $y/H = 1.5$ and $y/H = 2.5$) and at datapoints, the PINN-DA-SA and PINN-DA-Baseline show similarly low reconstruction errors. This is expected in this region given the lack of mean shear and production of turbulent kinetic energy.

Comparing the PINN-DA-Baseline error against the PINN-DA-SA highlights the effect of augmenting the PINN with the SA turbulence model. The high mean-flow error along the upper wall has largely reduced indicating that use of the SA model has enabled a better reconstruction of the near-wall region. There are also similarly significant improvements along the lower wall both at separation but also after reattachment. The improvements of PINN-DA-SA compared to PINN-DA-Baseline corroborate well with the original design purpose and performance of the SA turbulence model, which shows improved performance for turbulent boundary layers in adverse pressure gradients.

2. Why PINN-DA-SA?

To understand the difference in performance between PINN-DA-Baseline and PINN-DA-SA, we examine the individual terms of the cost function, C , for each approach. This consists of the measurement error term (data loss) and the enforcement of the governing equations (PDE loss).

First, we examine the data loss component. Inspection of Fig. 6(d) reveals that the PINN-DA-Baseline reconstructs the mean velocity measurements accurately. This is comparable to the performance of the PINN-DA-SA, with the convergence plots in Figs. 4(a) and 4(b) showing similar converged data loss values. Away from the data points, the reconstruction error increases substantially for the PINN-DA-Baseline. This is most apparent along the upper wall where a high error region exists above the data at $y/H = 2.5$. Given that the reconstruction error is low at the data points for PINN-DA-Baseline (and also similar to PINN-DA-SA), further decrease of the error to measurements, J , does not reduce the mean velocity error away from datapoints, where most of the reconstruction error lies.

Second, one can consider the other component of the cost function, the residual PDE error. Figure 7(a) shows the PINN-DA-Baseline momentum residual error (for x - and y -momentum equations). The PDE residual error contour shows that there is a limited correlation between mean velocity reconstruction error and residual error from the conservation laws—most apparent along the upper and lower walls. A similar analysis for the PINN-DA-SA corroborates this trend [Fig. 7(b)]. The total residual PDE loss from both PINN-DA-Baseline and PINN-DA-SA (seen in Fig. 4) are of similar magnitudes (approx. 10^{-5}) and the residual PDE fields are comparable, despite the significant differences in reconstruction accuracy between the two PINN-DA approaches. This indicates that further reduction in the residual PDE error does not lead to increase in reduction in reconstruction error and thus cannot be attributed as the cause of the difference between PINN-DA-Baseline and PINN-DA-SA.

The difference between PINN-DA-Baseline and PINN-DA-SA can be consequently attributed to the effect of the turbulence model augmentation. Due to the unclosed (underdetermined) nature of the RANS equations, there are infinite solutions to the forcing (divergence of Reynolds stress tensor) in the mean-flow equations. Providing high-fidelity data measurements “anchors” the PINN-DA solution at these discrete data points. However, away from them, there are still many candidate solutions to the mean flow, as the governing equations are still underdetermined. These solutions all satisfy the governing equations and thus all have low residual PDE error. There are consequently

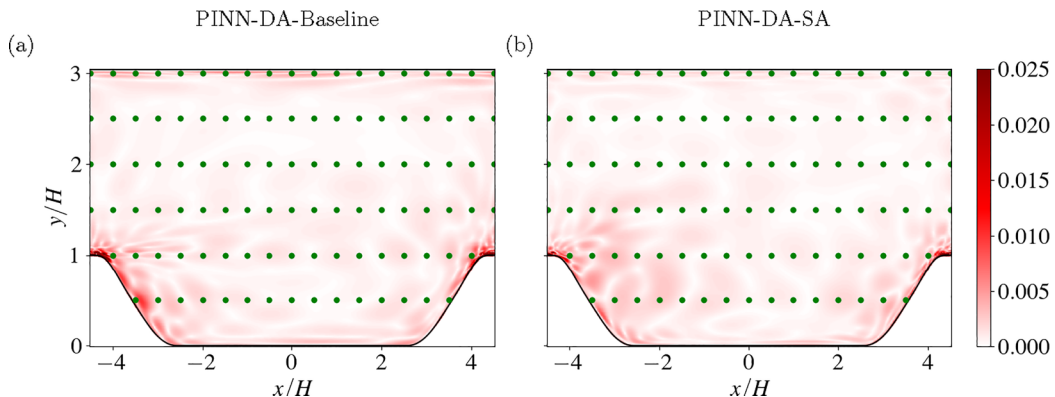


FIG. 7. PDE residual error (x -momentum and y -momentum). (a) PINN-DA-Baseline and (b) PINN-DA-SA. The residual error between approaches are comparable despite the considerable improvements in mean velocity reconstruction error demonstrated by the PINN-DA-SA.

no terms within the optimized cost function [Eq. (8)] that can be minimized further, to reduce the mean velocity reconstruction error across the domain for PINN-DA-Baseline. The current optimization for PINN-DA-Baseline is converged towards a solution that obeys the underdetermined RANS equations. This is also briefly mentioned in Foures *et al.* [17]. The SA model adds physics constraints that limit further the infinite number of admissible solutions far from the measurement points and as a direct consequence lead to improved reconstruction.

3. Variational-DA-SA results and comparison with PINNs

The PINN-DA-SA can be directly compared to the variational-DA-SA, which also uses SA augmentation, since both methods use the same data and physics constraints. Figures 6(c) and 6(f) show the reconstructed mean streamwise velocity and the respective absolute error for the variational-DA-SA. As with all the data assimilation results presented, the key flow features (separation, recirculation bubble and reattachment) are well reconstructed. As with both PINN-DA methodologies, the variational-DA-SA reconstructs the shape of the recirculation bubble more accurately than the RANS-SA, despite the limited measurements in this region. The reconstruction error matches the features seen in the PINN-DA results as well as in Ref. [17], with high error between data points, near walls, and at the initial separation.

The key feature seen in the comparison between variational-DA-SA and PINN-DA-SA is that the latter has demonstrably lower error across all regions of flow. The highest reduction in error in the PINN-DA-SA is observed in the region between hills ($y/H \leq 1.0$), specifically the recirculation and separation. While all approaches have topologically similar error fields (such as the initial separation), the PINN-DA-SA demonstrates a vastly reduced error. At data points, the reconstruction error is much lower for PINN-DA approaches than the variational-DA-SA. Figure 4 shows that the PINN-DA's have measurement error of the order of 10^{-7} while the variational-DA-SA is 10^{-5} . The relative performance of the variational-DA-SA and PINN-DA-SA compared with both RANS-SA and the DNS flow can be summarized in the velocity profiles in Fig. 8. While both data assimilation methods represent a significant improvement over a RANS-SA simulation, the PINN-DA-SA is almost identical to the DNS results, while small discrepancies are still more apparent for the variational-DA-SA.

Comparing both PINN-DA-SA and variational-DA-SA to PINN-DA-Baseline show that application of SA has been successfully used to reduce error for near-wall gradients, as a result of the additional physical description provided by the turbulence model. The specific choice of turbulence

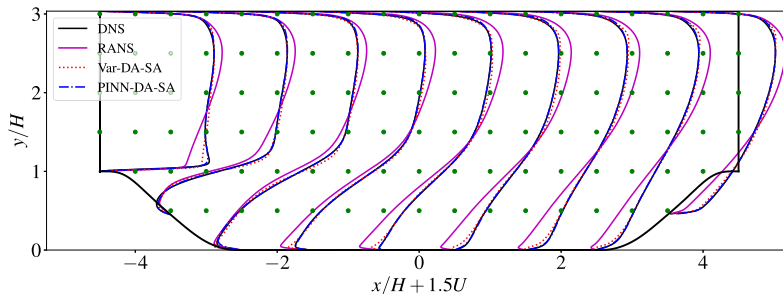


FIG. 8. Mean-flow reconstruction of streamwise velocity profiles for DNS, RANS-SA, PINN-DA-SA, and variational-DA-SA.

model (SA) is also a contributing factor, given the aforementioned good performance in boundary layers with adverse pressure gradients.

The lower reconstruction error of the PINN-DA-SA approach compared to the variational-DA-SA approach has been confirmed also for the laminar case, in the absence of the SA model. The test case for this was the unsteady cylinder flow at low Reynolds numbers. Appendix E includes details on the laminar flow case, followed by a comparison of results.

4. Forcing fields

The inferred forcing given by Eq. (5) consists of solenoidal and modeled eddy viscosity (SA contribution) components. For PINN-DA-Baseline, the modeled eddy viscosity component is zero. A further analysis of the forcing terms allows isolation of the effect of SA in the accurate reconstruction of the mean-flow field.

The PINN-DA-Baseline and PINN-DA-SA forcing fields shown in Fig. 9 are qualitatively similar in the uniform flow region between $y/H = 1.5$ and $y/H = 2.5$, where turbulent production is low. However, the regions of the flow where PINN-DA-Baseline shows high levels of error [as seen in Fig. 6(d)] result in larger forcing changes in the PINN-DA-SA solution. Introduction of a modeled component (via SA) has led to significant changes in the corrective forcing, most notably at the separation but also near the walls. These changes underpin the differences in reconstruction error between PINN-DA-Baseline and PINN-DA-SA. Constraining the solution with additional physical definition (via SA) has provided additional modeled forcing in regions where data-assimilation methods with the underdetermined RANS had the highest errors. While the PINN-DA-Baseline had converged to a poor solution, the PINN-DA-SA has reduced the “amount” of underdetermined forcing resulting in lower reconstruction errors. However, the addition of the corrective term, which is still unclosed, has allowed the data-assimilation approach to correct the SA model.

The PINN-DA-SA forcing is different to the variational-DA-SA forcing, in accordance with the differences in mean-flow reconstruction. In general, the PINN-DA-SA forcing is closer to the variational-DA-SA forcing than the PINN-DA-Baseline forcing. However, there are several differences in forcing, appearing in regions with the largest differences in reconstruction. This is most apparent in the significantly less smooth forcing around the separation in the variational-DA-SA forcing compared to PINN-DA-SA. The variational-DA-SA also exhibits a high forcing concentration in f_2 forcing along the rear hill, centered around a datapoint. This datapoint (in the $\Delta L = 0.5$ case) has shown to be particularly sensitive. While the gradient regularization procedure employed in the variational approach ensures that each update of forcing is smooth, it does not guarantee that the accumulation of updates and thus the final forcing will also be smooth. While other regularization strategies were investigated (see Appendix B 2), they led to poorer reconstructed mean velocity fields.

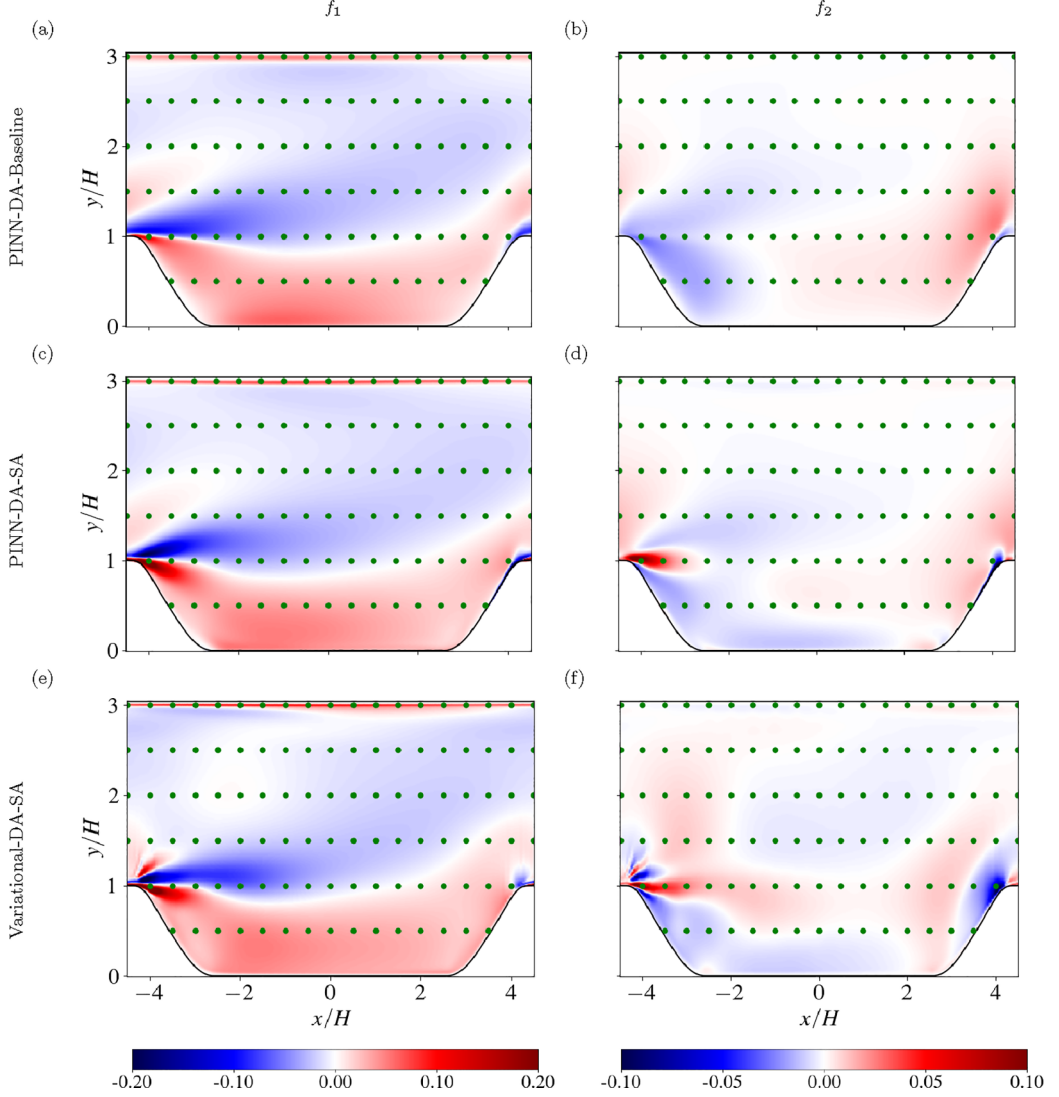


FIG. 9. A breakdown of the Reynolds forcing (excluding the potential field) f_1 (left) and f_2 (right) from the different methodologies: $f_i = \frac{\partial(2v_t S_{ij})}{\partial x_j} + f_{s,i}$. For PINN-DA-Baseline, $v_t = 0$. These are (a), (b) PINN-DA-Baseline, (c), (d) PINN-DA-SA, (e), (f) Variational-DA-SA.

To evaluate the accuracy of the data assimilation methods, the forcing is also compared with the true DNS forcing. As the potential forcing field is absorbed into the pressure term and thus is inseparable, we instead compare the curl of forcing, to remove the contribution of the potential forcing (since $\nabla \times \nabla \phi = 0$),

$$(\nabla \times \mathbf{f})_3 = \frac{\partial f_{s,2}}{\partial x} - \frac{\partial f_{s,1}}{\partial y} + \left(\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} \right) 2v_t S_{12} + \frac{\partial^2}{\partial x \partial y} [2v_t (S_{22} - S_{11})]. \quad (14)$$

A direct comparison can be performed then against the curl of the DNS forcing,

$$(\nabla \times \mathbf{f})_3 = \frac{\partial f_2}{\partial x} - \frac{\partial f_1}{\partial y} = \left(\frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} \right) \overline{u'v'} + \frac{\partial^2}{\partial x \partial y} (\overline{v'v'} - \overline{u'u'}). \quad (15)$$

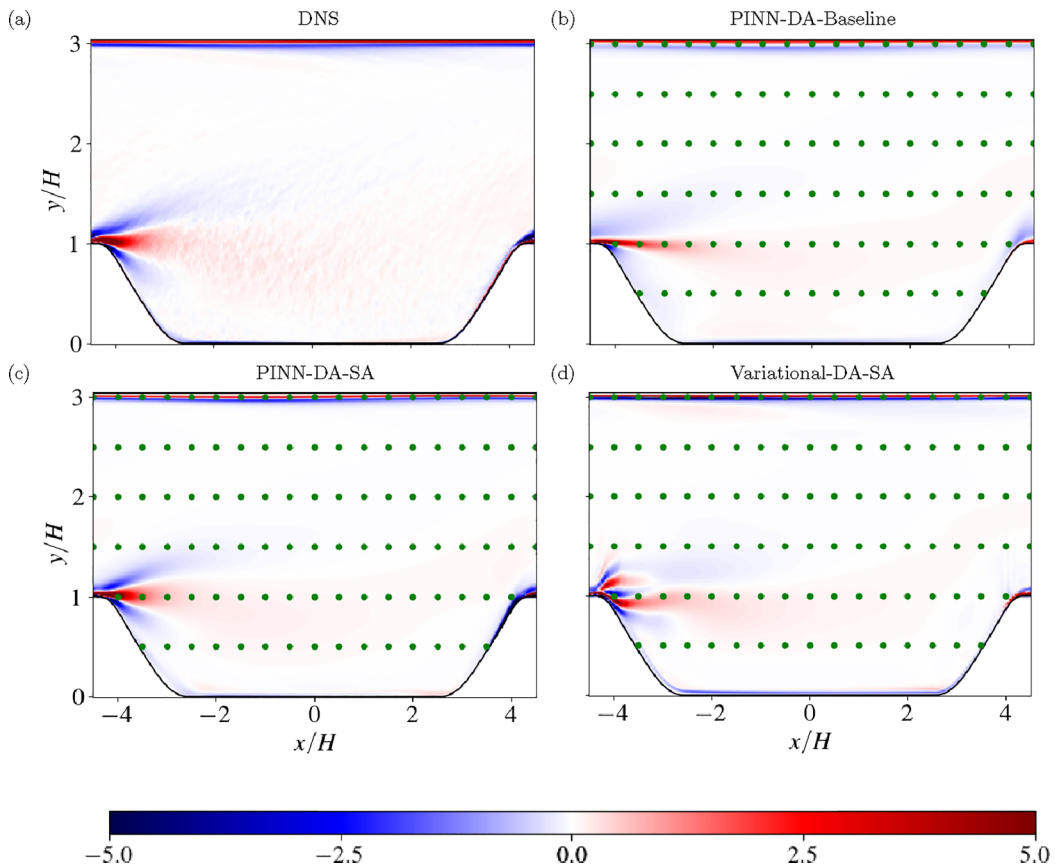


FIG. 10. $\nabla \times \mathbf{f}$. Comparison of curl (\mathbf{f}) between the ground truth (DNS) and the different data assimilation techniques presented in this paper. (a) DNS, (b) PINN-DA-Baseline, (c) PINN-DA-SA, (d) Variational-DA-SA.

The curl of the inferred forcing for all three reconstruction methods employed here is shown in Fig. 10. Best agreement with the DNS is achieved for PINN-DA-SA, which aligns with the most accurate mean-flow reconstruction. The PINN-DA-Baseline differs from DNS in several key regions - the initial separation, the recirculation, the rearward hill apex and along the walls.

Differences between the variational-DA-SA and PINN-DA-SA forcing (and curl) correlate with differences in the error fields. These are most apparent at the separation region and along the upper and lower walls.

5. Data assimilation with varying data resolution

Here, we evaluate the dependence of the reconstruction error on the data resolution of the available high-fidelity measurements. In Fig. 11, we compare variational-DA-SA, PINN-DA-Baseline and PINN-DA-SA approaches, for a range of data-resolutions, ranging from $\Delta L = 0.05$ to $\Delta L = 1.0$. The reconstruction error increases monotonically as the (equidistant/square) spacing between data points increases. All data assimilation methods improve the RANS-SA solution, irrespective of resolution. For data spacing $\Delta L < 1$, the turbulence model augmented PINN-DA-SA method achieves the lowest reconstruction error against all the other methods. Additionally, while the $\Delta L = 0.8$ has not converged as well for the variational-DA-SA approach, the reconstruction error remains close to the neighboring points.

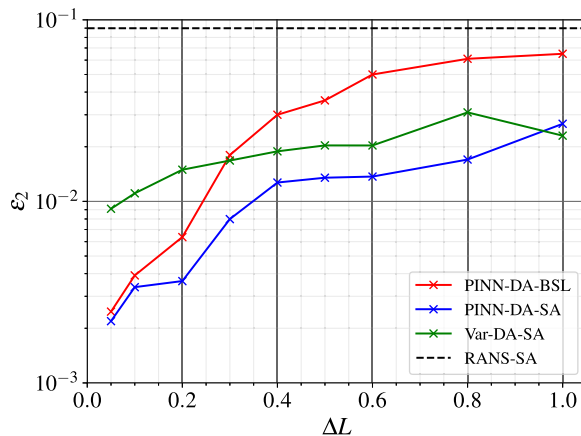


FIG. 11. Volume-weighted L_2 mean velocity reconstruction error [as defined in Eq. (13)] versus data resolution for the three reconstruction methodologies (solid lines with symbols). All three approaches improve the RANS-SA predictions (dashed line).

Comparing the PINN-DA-Baseline to the PINN-DA-SA results shows the significant improvement achieved through the turbulence model augmentation. The gap between the two can mostly be attributed to the error near the walls and in the separation. Additionally, the rate of error increase with respect to data resolution is steeper for PINN-DA-Baseline. This suggests adding a modeled (closed) forcing component (i.e., PINN-DA-SA) reduces the effect coarsening data resolution has on reconstruction accuracy compared to PINN-DA-Baseline. As data resolution becomes finer, the two PINN-DA approaches converge. This is expected—as more data is provided, there are more points at which the PINN-DA-Baseline can close the solution and thus the advantage the turbulence model augmentation provides is diminished.

Comparing the PINN-DA-SA versus the variational-DA-SA solution shows that, for most of the resolution range, the PINN-DA-SA also demonstrates consistently lower error at equivalent resolutions. In fact, the PINN-DA-SA at $\Delta L = 0.8$ matches the reconstruction accuracy of the variational-DA-SA at $\Delta L = 0.3$. However, at the extreme end of coarse data ($\Delta L = 1.0$), while the PINN-DA-SA is still a significant improvement over the PINN-DA-Baseline (and also RANS-SA), the variational-DA-SA approach has a lower reconstruction error. This suggests that the PINNs have a greater dependence on data for good performance. As data resolution reduces (spacing increases), the optimization problem converges to solving the direct RANS-SA problem, for which the numerical framework within the variational-DA-SA method is more robust, i.e., PINNs are less effective forward solvers. Algorithmic improvements for PINNs, including adaptive weighting algorithms, can potentially provide further improvements of the PINN approach (as demonstrated by Ref. [49]), which is currently under investigation. On the opposite end, for fine data resolutions, the variational-DA-SA is less accurate than the PINN-DA-Baseline.

One may attribute the difference between the reconstruction accuracy of the PINN-DA-SA and variational-DA-SA to the differences in the way the RANS equations are enforced in the data assimilation procedure. Concerning the variational-DA-SA approach, it may be first noticed that the latter solves for only the weak form of the RANS equations, following the finite-element approach. In addition, as mentioned in Sec. III D, stabilization terms are added in the governing equations. Then there is the process of spatial discretization through, among others, the choice of shape functions, which will determine the order of spatial accuracy. In contrast, PINNs enforce the RANS equations in their original, strong formulation, and in a continuous way. Moreover, the effect of applying discrete pointwise forcing at measurement locations poses a greater challenge for the variational approach than PINNs and thus the method of regularization has a greater

impact on reconstruction performance. Appendix B 2 contains analysis of regularization choice. By avoiding both these sources of error, PINN-DA-SA achieves a lower reconstruction error than the variational-DA-SA. This is evidenced by the reconstruction error for finer data resolutions. Even as data resolution increases, the aforementioned discretization error remains. Consequently, the reconstruction error of the variational-DA-SA reduces at a much slower rate than both PINN-DAs.

V. CONCLUSION

Three approaches to turbulent mean-flow reconstruction from sparse data measurements have been presented and compared. First, it was shown that for all methods, inferring the closure of the underdetermined RANS equations or the corrective closure of the RANS-SA equations by assimilating high-fidelity sparse data improved the mean-flow reconstruction compared to RANS-SA predictions.

Second, the use of Spalart-Allmaras turbulence model, as used in Ref. [8] (in the variational-DA-SA approach) was proposed for use in PINNs, particularly in the context of turbulent flows. The SA turbulence model augmented PINN (PINN-DA-SA) proved to be the most accurate flow reconstruction method followed by the equivalent approach albeit with a variational formulation. Comparing the PINN-DA-Baseline (PINN without turbulence SA model) with PINN-DA-SA showed the isolated effect of turbulence model augmentation. The PINN-DA-SA performs significantly better in regions with complex flow features, such as near walls and at separation. Injection of additional physical constraints in the form of a turbulence model provides an additional modeled component to the underdetermined Reynolds forcing, reducing the complexity of the optimization and narrowing the solution space for the optimization. A corrective forcing term is then tuned to correct the rest of the closure.

Third, the variational-DA-SA approach, while an improvement over the PINN-DA-Baseline, has higher errors compared to the PINN-DA-SA, most notably off the separation. These observations for mean-flow reconstruction accuracy are also associated with equivalent error in Reynolds forcing. The difference in variational-DA-SA and PINN-DA-SA can be attributed to the effect of discretization, which is avoided by PINNs. These observations are consistent across different data resolutions.

Given that PINNs is a growing field of research, there are potentially many avenues to improve the reconstruction capability further. The most apparent example is hyperparameter tuning, such as loss weighting. In this study, a static weighting method was used with manual iterations to tune them. This is a laborious exercise, particularly as systems become more complex with more boundary conditions and less uniform topologies. Adaptive weighting algorithms [49–51] or novel architectures, such as Competitive PINNs [52], would both simplify this problem and also generalize the process to selecting weights. This has been demonstrated in Appendix C.

The code and data to run the PINN cases presented here are available at Ref. [43].

APPENDIX A: SPALART-ALLMARAS EQUATIONS

The full definition of the SA transport equations, as mentioned in Eq. (6), is described here.

1. Formulation of Spalart-Allmaras equations

The Spalart-Allmaras equations are solved for $\tilde{\nu}$. Then, the eddy viscosity, ν_t , is obtained as

$$\nu_t = \begin{cases} 0 & \text{if } \tilde{\nu} < 0, \\ f_{\nu 1} \tilde{\nu} & \text{if } \tilde{\nu} \geq 0, \end{cases} \quad (\text{A1})$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{v}}{v}. \quad (\text{A2})$$

The SA transport equation takes the form

$$U_j \frac{\partial \tilde{v}}{\partial x_j} - S_p - S_{\text{diff}} - S_c - S_d = 0, \quad (\text{A3})$$

where S_p , S_{diff} , S_c , S_d are production, diffusion, cross-diffusion, and destruction terms, respectively. These are defined as

$$S_p = \begin{cases} c_{b1} \tilde{S} \tilde{v} g_n & \text{if } \tilde{v} < 0, \\ c_{b1} \tilde{S} \tilde{v} & \text{if } \tilde{v} \geq 0, \end{cases} \quad S_d = \begin{cases} c_{w1} \frac{\tilde{v}^2}{d^2} & \text{if } \tilde{v} < 0, \\ -c_{w1} f_w \frac{\tilde{v}^2}{d^2} & \text{if } \tilde{v} \geq 0, \end{cases}$$

$$S_{\text{diff}} = \frac{1}{\sigma} \nabla \cdot (\eta \nabla \tilde{v}) = \frac{1}{\sigma} \left(\eta \frac{\partial^2 \tilde{v}}{\partial x_j^2} + \frac{\partial \eta}{\partial x_j} \frac{\partial \tilde{v}}{\partial x_j} \right), \quad S_c = \frac{c_{b2}}{\sigma} \|\nabla \tilde{v}\|^2 = \frac{c_{b2}}{\sigma} \left(\frac{\partial \tilde{v}}{\partial x_j} \right)^2, \quad (\text{A4})$$

where the auxiliary functions and constants are defined as

$$\eta = \begin{cases} \nu(1 + \chi + \frac{1}{2}\chi^2) & \text{if } \tilde{v} < 0, \\ \nu(1 + \chi) & \text{if } \tilde{v} \geq 0, \end{cases} \quad S' = \|\nabla \times \mathbf{U}\|, \quad S = \sqrt{S'^2 + M^2} - M,$$

$$\bar{S} = \frac{\tilde{v} f_{v2}}{\kappa^2 d^2}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad \tilde{S} = \max(10^{-10}, S + \bar{S}),$$

$$r' = \frac{\tilde{v}}{\tilde{S} \kappa^2 d^2}, \quad r = \begin{cases} r' & \text{if } 0 \geq r' \geq 10, \\ 0 & \text{otherwise,} \end{cases} \quad g = r + c_{w2}(r^6 - r),$$

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{\frac{1}{6}}, \quad g_n = 1 - \frac{1000 \chi^2}{1 + \chi^2}. \quad (\text{A5})$$

The constants in the SA transport equations are $c_{v1} = 7.1$, $c_{v2} = 0.7$, $c_{v1} = 0.9$, $\kappa = 0.41$, $\sigma = \frac{2}{3}$, $c_{b1} = 0.1355$, $c_{b2} = 0.622$, $c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1+c_{b2}}{\sigma}$, $c_{w2} = 0.3$, $c_{w3} = 2$, $M = 10^{-5}$.

2. Wall-distance multiplied Spalart-Allmaras equations

To remove singularities, which cause problems for the PINN-DA-SA formulation at points evaluated at (or near) the wall (as discussed in Sec. III C), the modified SA equations take the form

$$d^2 \left(U_j \frac{\partial \tilde{v}}{\partial x_j} - S_{\text{diff}} - S_c \right) - S_p^N - S_d^N = 0, \quad (\text{A6})$$

where S_p^N , S_d^N are new modified production and destruction terms, respectively. These are defined as

$$S_p^N = d^2 S_p = \begin{cases} c_{b1} S^N \tilde{v} g_n & \text{if } \tilde{v} < 0, \\ c_{b1} \tilde{S}^N \tilde{v} & \text{if } \tilde{v} \geq 0, \end{cases} \quad S_d^N = d^2 S_d = \begin{cases} c_{w1} \tilde{v}^2 & \text{if } \tilde{v} < 0, \\ -c_{w1} f_w \tilde{v}^2 & \text{if } \tilde{v} \geq 0. \end{cases} \quad (\text{A7})$$

The modified auxiliary functions are thus

$$S^N = d^2 S = d^2 (\sqrt{S'^2 + M^2} - M), \quad \bar{S}^N = d^2 \bar{S} = \frac{\tilde{v} f_{v2}}{\kappa^2},$$

$$\tilde{S}^N = d^2 \tilde{S} = \max(10^{-10}, d^2(S + \bar{S})) = \max(10^{-10}, S^N + \bar{S}^N), \quad r' = \frac{\tilde{v}}{\tilde{S}^N \kappa^2}. \quad (\text{A8})$$

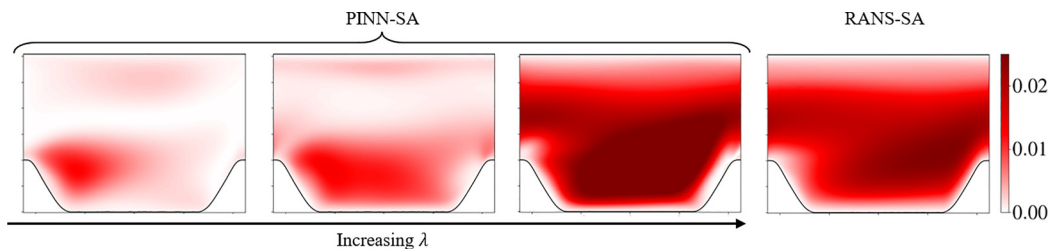


FIG. 12. Effect of regularization parameter (solenoidal forcing magnitude penalization), λ^R on PINN-SA solution for \tilde{v} . \tilde{v} for increasing λ^R compared with the RANS-SA \tilde{v} shows that increasing regularization causes eddy viscosity to converge towards the RANS solution. The presented values of λ^R are 10^{-8} , 10^{-4} , 10^0 left to right.

APPENDIX B: EFFECT OF REGULARIZATION ON DATA ASSIMILATION TECHNIQUES

As mentioned in Sec. II B, the forcing decomposition between modeled and corrective parts is not unique and thus the regularization of corrective forcing is critical. This section will show the effect of regularization for both PINN-DA and variational approaches and highlight its importance.

1. Effect of solenoidal forcing regularization on PINNs

For PINN-DA-SA, an L_2 penalization of the solenoidal forcing magnitude was applied. During PINN-DA-SA hyperparameter tuning, it was found that the final solution was strongly influenced by the regularization of solenoidal forcing. Clearly as regularization weight, λ^R , increases, the magnitude of solenoidal forcing is penalized more and thus there is a greater dependence on the modeled eddy viscosity forcing component. In effect, the greater the value of λ , the closer the governing equations tend towards a pure RANS-SA problem. However, for the PINN-DA, enforcement of data loss at measurement points results in differences. This is demonstrated in Fig. 12. As solenoidal forcing magnitude is penalized more and more, the solution tends towards RANS-SA. However, this is an imperfect solution as the PINN-DA-SA is attempting to fit high-fidelity (mean) DNS data to the RANS-SA equations. This is most notable at the point around the rear hill apex. As a result, the residual PDE loss begins to increase at very high λ^R . However, without penalization ($\lambda^R = 0$), the PINN-DA-SA solution tends towards the PINN-DA-Baseline solution with a negligible contribution from the eddy viscosity component ($\tilde{v} \approx 0$).

As the data coarsens, and thus the dependence on data reduces, the requirement for a modeled component becomes more important, as the corrective Reynolds forcing can be closed at fewer points using data. Less data means the solution will move closer to a RANS-SA solution. As a result, the coarser the data, the greater the value for the optimal λ^* , as the modeled forcing component becomes a greater component of the forcing. This is seen in Fig. 13.

2. Importance of regularization for variational DA

For the variational DA, the choice of regularization (of solenoidal forcing $f_{s,i}$) was more important. Three approaches were trialed: no regularization; L_2 regularization; H_1 regularization.

First, without regularization, while data is well enforced [as seen in Fig. 14(a)] even compared with the final H_1 based approach, it results in a nonphysical solenoidal forcing that appears strongly pointwise, clearly seen in Fig. 14(d), as large corrections are applied at the data locations. The final reconstruction is thus less physical. Additionally, regularization is needed to enforce convergence to a unique decomposition.

For the L_2 regularization approach, error is higher than either H_1 or no regularization methods. This is due to the inefficiency of the approach in the variational framework. To achieve sufficient smoothing of the solution, avoiding the problems of the none regularized approach, very large

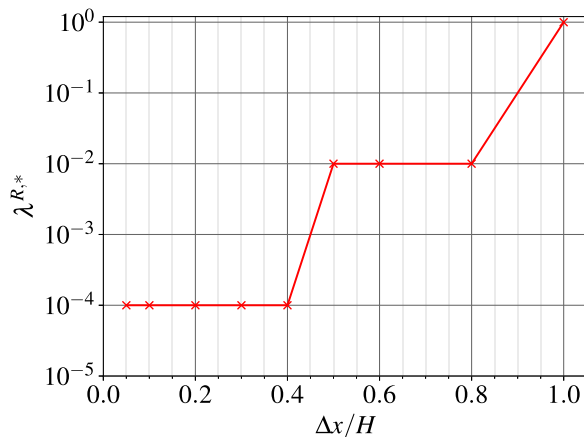


FIG. 13. The optimized regularization parameter, $\lambda^{R,*}$, with respect to data resolution.

penalization weighting is needed. However, this comes at the cost of poor data enforcement, as seen in Fig. 14(b).

For the variational-DA-SA, H_1 regularization was selected to smooth out the solution. As demonstrated, for the variational approach, H_1 regularization is a much more suitable approach. It allows much better matching of the data (as compared with L_2 regularization), while also giving a more physical solution. As a result this was selected as the final regularization method.

For PINN-DA-SA, good data enforcement and a smooth nonpointwise solenoidal forcing was always ensured. Regularization is exclusively used to control the distribution between modeled and corrective forcing. Given its computational simplicity, L_2 was selected for PINN-DA-SA. This was not the case for variational-DA-SA and regularization was needed to smooth the solution in addition to the distribution of modeled and corrective forcing.

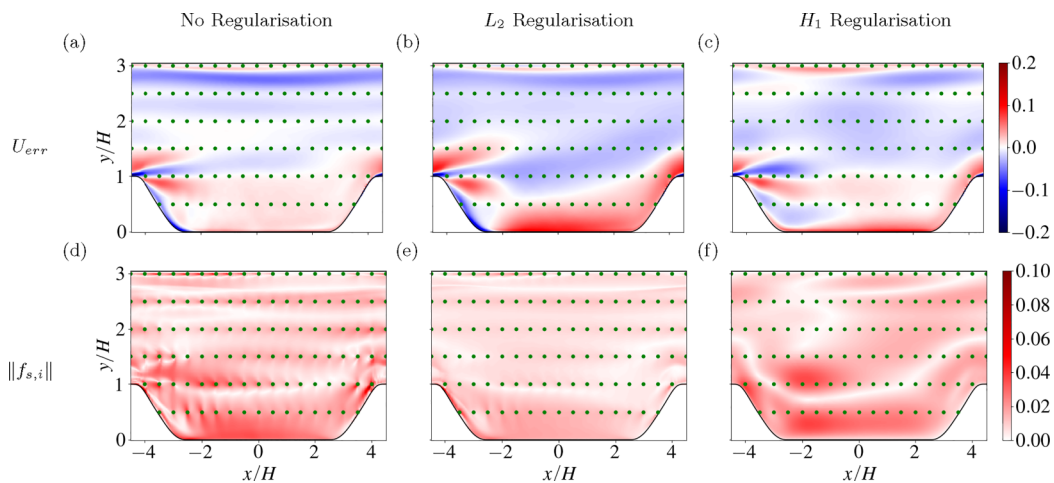


FIG. 14. Absolute U error (a)–(c) and corrective solenoidal forcing, $\|f_{s,i}\|$, (d)–(f) for variational-DA-SA method using different regularization approaches: No regularization (left), L_2 regularization (middle) and H_1 regularization (right).

TABLE II. Sensitivity of final model loss function, C , to activation functions and learning rate on a model with seven layers and 50 nodes/layer relative to final choice (tanh and 5×10^{-4} learning rate).

Activation functions					Learning rate (with tanh activation)		
tanh	ReLU	sin	ELU	L-LAAF tanh	1×10^{-3}	5×10^{-4}	1×10^{-4}
–	+55.4%	+9.1%	+69.3%	+5.1%	–1.3%	–	+3.6%

APPENDIX C: HYPERPARAMETER AND LOSS WEIGHT SELECTION

1. Tuning the model with a grid search approach

To select hyperparameters (model architecture, activation function and learning rate) and loss weighting, a grid search method was used. Tables II and III, show the sensitivity of final model loss to hyperparameters. From these results, the architecture, as discussed in Sec. III B, was selected. A similar analysis was completed for the loss weighting. A summary of the key results are shown in Table IV. Repeats of the final model were found to converge within 3.0% of the loss presented in Sec. IV A (after three repetitions).

A tanh activation was selected due to the lowest value of cost function. While an initial learning rate of 1×10^{-3} demonstrated a slightly lower final loss value, this model was less repeatable than the chosen learning rate (5×10^{-4}). Similarly, the model architecture (nodes and layers) was selected for demonstrating the lowest final loss value. While adding more layers and nodes performs similarly, the additional model complexity increases computational cost.

2. Alternative tuning methodologies

While a grid search approach was used to tune hyperparameters in this work, there have been many papers investigating the use of more robust methods. In this section, two approaches are investigated: adaptive activation functions and adaptive weights.

Adaptive activation functions, such as the layerwise locally adaptive activation function (L-LAAF) proposed in Ref. [53], have shown promise at both accelerating convergence of PINNs, as well as improving model accuracy. For every layer, L-LAAF introduces a learnable tuning parameter, a into the activation function to scale its gradient:

$$\sigma^*(x) = \sigma(ax), \quad (C1)$$

where σ^* , σ are the adaptive and base activation functions, respectively, and x is some input into the activation function. The effect of applying L-LAAF (with tanh) to the periodic hill problem is presented in Table II and Fig. 15(a). For this problem, there was a minimal effect on the converged loss value. There does appear to be some improvement in the ‘spikiness’ seen in the initial ADAM phase. However, during the second, long L-BFGS-B phase, this is no longer seen and the convergence speed is almost identical.

 TABLE III. Sensitivity on final model loss function, C , to architecture parameters (number of layers and nodes) relative to final architecture (7 layers, 50 nodes).

		Number of layers				
		5	6	7	8	9
Number of nodes/layer	25	+19.7%	+15.6%	+9.9%	+7.5%	+5.2%
	50	+7.4%	+5.7%	–	+2.8%	+2.6%
	75	+6.3%	+5.5%	+2.8%	+4.7%	+6.3%

TABLE IV. Sensitivity on final model loss function, C , to perturbations on weights from Table I.

λ^P		λ^D		λ^B	
$\times 10$	$\times 0.1$	$\times 10$	$\times 0.1$	$\times 10$	$\times 0.1$
+4.7%	+8.6%	+3.8%	+9.3%	+1.6%	+7.2%

Adaptive weighting algorithms have been an increasingly promising area of research to make robust and reduce the difficulty of model weight tuning process. This is a particular challenge for PINNs, where there are often many loss terms in the equations. Figure 15(b) demonstrates the advantage of using the adaptive loss weighting algorithm, as implemented in Ref. [29]. While the difference in the final loss between the model with optimized weights (from Table I) and the model with adaptive weighting is minimal, the latter of these avoids all the intermediate grid-search steps used to find and select the optimal weights. This massively reduces the computational cost of finding the correct weighting and is thus the recommended approach moving forward.

APPENDIX D: RECONSTRUCTING PRESSURE WITH PINN-DA-BASELINE

Instead of applying the Helmholtz decomposition (4) with the PINN-DA-Baseline, as in Sec. IV B, one can alternatively enforce the base RANS equations as presented in Eq. (2), using both high-fidelity first-order statistics (mean-flow) and second-order statistics (Reynolds stress) ($U_i, \overline{u'_i u'_j}$) at discrete locations to reconstruct the flow field. This approach enables reconstruction of both mean velocity and Reynolds stresses fields but also allows the inference of the pressure field. The weights used to train this model are compiled in Table V.

Using the data spacing $\Delta L = 0.5$, shows that this alternate PINN-Baseline approach produces comparable mean velocity reconstruction as the PINN-DA-Baseline, as seen in Fig. 16(a). The L_2 error is 3.37×10^{-2} for this second approach compared with 3.60×10^{-2} for the former approach. Likewise the key trends discussed in Sec. IV B all still hold. However, this alternate formulation has more output variables and fewer equations when compared to the approach used in the main paper.

Although both approaches achieve similar reconstruction error, this second approach requires second-order statistics (Reynolds stress measurements) along with first-order ones (mean-flow field). However, this formulation can be used to reconstruct the pressure field, as shown before for laminar cases [29]. This is demonstrated in the turbulent case, as seen in Figs. 16(b) and 16(c). This has a high potential for practical applications to obtain the pressure field without performing intrusive measurements. Examination of the RANS equations (2) shows how provision of mean

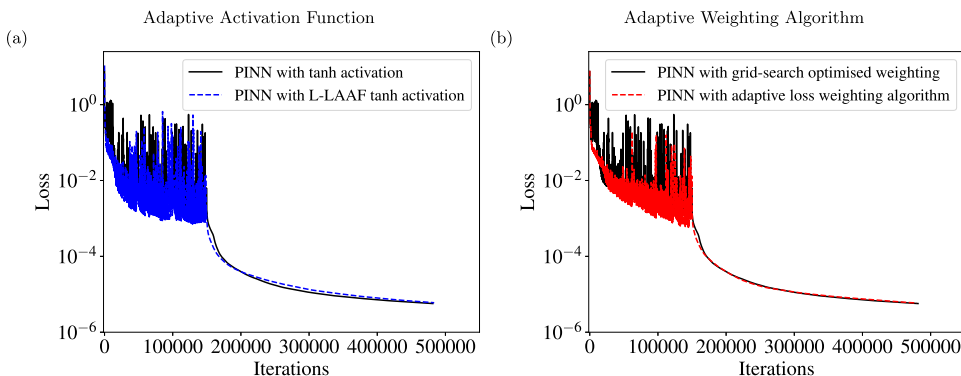


FIG. 15. Comparing the effect of convergence for PINN-DA-SA by applying (a) layerwise locally adaptive activation functions (L-LAAF) (b) an adaptive loss weighting algorithm.

TABLE V. Weights used in PINN optimization when both mean velocity and Reynolds stress data are provided.

Weight	λ^P	$\lambda^D(U_i)$	$\lambda^D(\overline{u'_i u'_j})$	$\lambda^B(U_i)$ -Wall	$\lambda^B(\overline{u'_i u'_j})$ -Wall	λ^B -Periodic
Value	1	10	100	2.5	10	1

velocity and Reynolds stress data at discrete points defines all terms in the equations except for the pressure term. At these measurement locations, the equations are closed and the resultant neural network can infer a pressure field, both at data points but also away from them as well. This resultant pressure field has highest reconstruction error in the regions with highest pressure gradients in the flow, matching the trend seen with mean velocity error. The pressure reconstruction error is less correlated with the location of data points, than the mean velocity. It should be highlighted that the neural network pressure output must also include an integration constant (as the RANS equations only contain derivatives of pressure). To determine this constant and recover the actual, unshifted pressure field, one needs to fix the pressure at one point. This can be done during training (by providing pressure data at a single point) or as in this case, the pressure field is shifted in post processing. Taking the pressure data at a singular point, one can find a constant, such that by adding it to the model pressure at that point, the predicted pressure exactly matches the measurement. This constant can then be applied to the full solution.

APPENDIX E: LAMINAR CYLINDER FLOW

Both the variational-DA [17] and PINN-DA-Baseline [29] techniques (without the use of turbulence model augmentation) have already been successfully demonstrated on a laminar circular

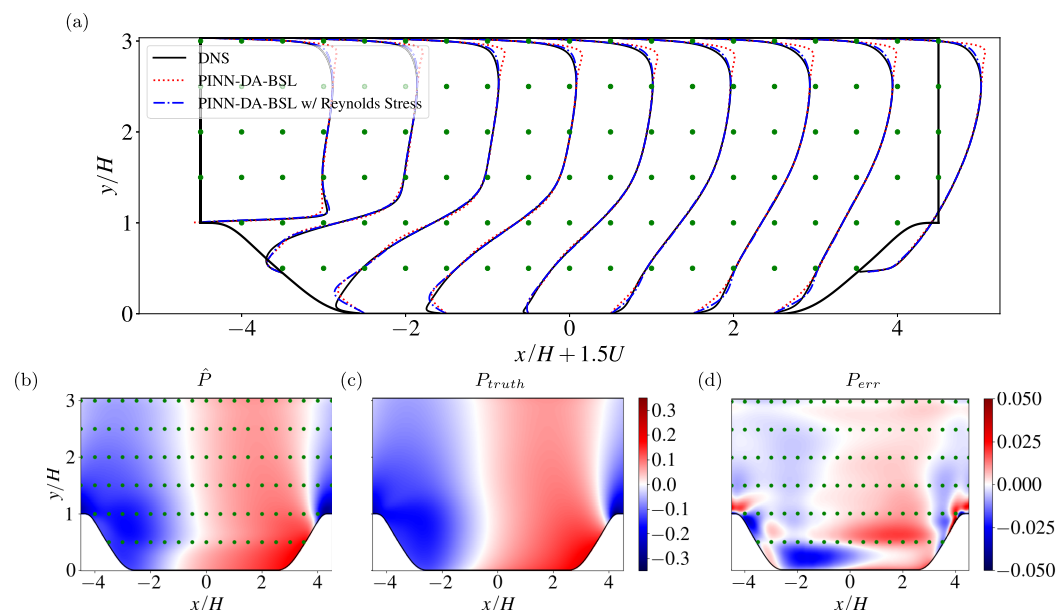


FIG. 16. PINN-DA-Baseline mean-flow reconstruction. (a) Comparison between the DNS, PINN-DA-Baseline and PINN-DA-Baseline (with Reynolds stress data) velocity profiles for the U component. (b) Pressure field inferred from the PINN-DA-Baseline approach from sparse mean velocity and Reynolds stress data. (c) Absolute mean pressure error from PINN-DA-Baseline pressure reconstruction. (d) Absolute mean pressure error from PINN-DA-Baseline pressure reconstruction.

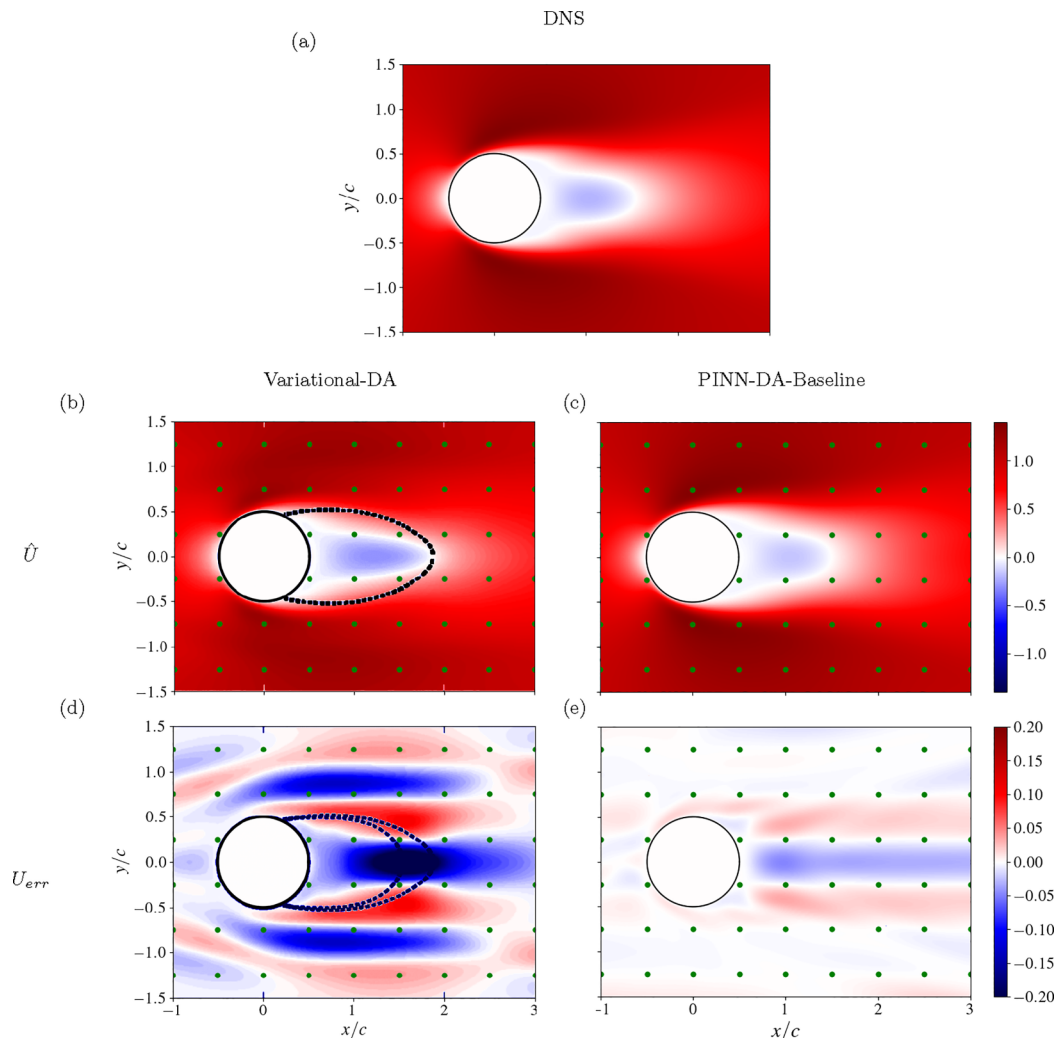


FIG. 17. Comparison of the U (x -component) mean velocity component reconstructions for $d = 0.5$ data resolution with DNS (top). The variational-DA approach (left) and the PINN-DA-Baseline (right) figures are taken from [17,29], respectively. (a) U truth. (b), (c) U prediction. (d), (e) Absolute U error.

cylinder case. However, a quantitative comparison has not been completed, which is presented in this section.

A direct numerical simulation of a 2D circular cylinder flow at $Re = 150$ is performed to generate the high-fidelity database, which is used to extract sparse mean velocity measurements for use in the above data-driven techniques. The details of the simulation setup and methodologies can be found in Refs. [17,29] for the variational and PINN case, respectively. Figure 17(a) shows the time-averaged mean U field for the circular cylinder. For this work, the data spacing, $d = 0.5$, on a rectangular grid is used to compare approaches as a comparison case. The aforementioned papers also contain results of differing data resolutions.

This direct comparison of both variational-DA and PINN-DA approaches was used as a benchmark for what the turbulent applications can achieve (as shown in the main paper). These results will demonstrate that for laminar flow, the PINN-DA approach flow reconstruction can match and improve the variational-DA reconstructions.

Figure 17 shows the reconstructed U field (b, c) and the prediction error (d, e). This highlights several trends, which are consistent across all velocity components and were also seen in the turbulent cases. First, the reconstructed mean velocity fields are very similar to the true fields and accurately capture the expected physical features, such as the symmetrical wake and the stagnation at the front of the cylinder. Additionally, a key pattern emerged from this result that was consistent across both the laminar and turbulent cases. Both the variational-DA and PINN-DA approaches have almost identical reconstruction error fields. These occur between datapoints, in the wake and recirculation regions and near the wall. Finally, as one moves away from the cylinder (in y), the error reduces as the flow becomes more uniform.

Whilst the reconstruction error fields are comparable, it is clear that the absolute error for the PINN-DA-Baseline is much smaller in magnitude. The variational-DA has a maximum error magnitude of 0.3 (for absolute U error), whereas for PINN-DA it is around a third of this at 0.11. This comparison is important as it suggests, all else equal for a laminar cylinder flow, the PINN-DA produces better results. This is most apparent when comparing the length of the wake. While the variational approach predicts the wake recirculation region closes at $x = 1.85$, the PINN approach closes at $x = 1.55$ (compared with $x = 1.55$ for the DNS flow field). Furthermore, the decay in error, observed as one moves away from the body, is much more apparent in the PINN-DA results. This may indicate the fundamental nature of PINNs and their construction, allows them to better generalize these simple uniform regions.

-
- [1] P. Spalart and S. Allmaras, A one-equation turbulence model for aerodynamic flows, *Proceedings of the 30th Aerospace Sciences Meeting and Exhibit* (AIAA, Reno, Nevada, USA, 1992), p. 439
 - [2] D. C. Wilcox, Reassessment of the scale-determining equation for advanced turbulence models, *AIAA J.* **26**, 1299 (1988).
 - [3] B. E. Launder and B. I. Sharma, Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc, *Lett. Heat Mass Transfer* **1**, 131 (1974).
 - [4] H. Xiao and P. Cinnella, Quantification of model uncertainty in RANS simulations: A review, *Prog. Aerosp. Sci.* **108**, 1 (2019).
 - [5] K. Duraisamy, G. Iaccarino, and H. Xiao, Turbulence modeling in the age of data, *Annu. Rev. Fluid Mech.* **51**, 357 (2019).
 - [6] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, Machine learning for fluid mechanics, *Annu. Rev. Fluid Mech.* **52**, 477 (2020).
 - [7] A. P. Singh and K. Duraisamy, Using field inversion to quantify functional errors in turbulence closures, *Phys. Fluids* **28**, 045110 (2016).
 - [8] L. Franceschini, D. Sipp, and O. Marquet, Mean-flow data assimilation based on minimal correction of turbulence models: Application to turbulent high Reynolds number backward-facing step, *Phys. Rev. Fluids* **5**, 094603 (2020).
 - [9] W. N. Edeling, P. Cinnella, and R. P. Dwight, Predictive RANS simulations via Bayesian model-scenario averaging, *J. Comput. Phys.* **275**, 65 (2014).
 - [10] J. L. Wu, J. X. Wang, and H. Xiao, A Bayesian calibration–prediction method for reducing model-form uncertainties with application in RANS simulations, *Flow, Turbul. Combust.* **97**, 761 (2016).
 - [11] O. Saredi, N. T. Ramesh, A. Sciacchitano, and F. Scarano, State observer data assimilation for RANS with time-averaged 3D-PIV data, *Comput. Fluids* **218**, 104827 (2021).
 - [12] O. Marquet, V. Mons, M. Zauner, and B. Leclaire, Turbulent mean flow estimation with state observer assimilation of velocity measurements in RANS models, in *Proceedings of the 12th International Symposium on Turbulence and Shear Flow Phenomena (TSFP12)* (HAL CCSD, Osaka, Japan, 2022).

- [13] S. Luo, M. Vellakal, S. Koric, V. Kindratenko, and J. Cui, Parameter identification of RANS turbulence model using physics-embedded neural network, in *Proceedings of the International Conference on High Performance Computing* (Springer, Berlin, 2020), pp. 137–149.
- [14] H. Eivazi, M. Tahani, P. Schlatter, and R. Vinuesa, Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations, *Phys. Fluids* **34**, 075117 (2022).
- [15] J. G. R. von Saldern, J. M. Reumschüssel, T. L. Kaiser, M. Sieber, and K. Oberleithner, Mean flow data assimilation based on physics-informed neural networks, *Phys. Fluids* **34**, 115129 (2022).
- [16] F. Pioch, J. H. Harmening, A. M. Müller, F. Peitzmann, D. Schramm, and O. el Moctar, Turbulence modeling for physics-informed neural networks: Comparison of different RANS models for the backward-facing step flow, *Fluids* **8**, 43 (2023).
- [17] D. P. G. Foures, N. Dovetta, D. Sipp, and P. J. Schmid, A data-assimilation method for Reynolds-averaged Navier–Stokes-driven mean flow reconstruction, *J. Fluid Mech.* **759**, 404 (2014).
- [18] S. Symon, N. Dovetta, B. J. McKeon, D. Sipp, and P. J. Schmid, Data assimilation of mean velocity from 2D PIV measurements of flow over an idealized airfoil, *Exp. Fluids* **58**, 61 (2017).
- [19] A. P. Singh, A framework to improve turbulence models using full-field inversion and machine learning, Ph.D. thesis, The University of Michigan, 2018.
- [20] J. R. Holland, J. D. Baeder, and K. Duraisamy, Towards integrated field inversion and machine learning with embedded neural networks for RANS modeling, in *Proceedings of the AIAA Scitech Forum* (AIAA, San Diego, California, USA, 2019), p. 1884.
- [21] P. S. Volpiani, M. Meyer, L. Franceschini, J. Dandois, F. Renac, E. Martin, O. Marquet, and D. Sipp, Machine learning-augmented turbulence modeling for RANS simulations of massively separated flows, *Phys. Rev. Fluids* **6**, 064607 (2021).
- [22] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* **378**, 686 (2019).
- [23] A. D. Jagtap, D. Mitsotakis, and G. E. Karniadakis, Deep learning of inverse water waves problems using multi-fidelity data: Application to Serre–Green–Naghdi equations, *Ocean Eng.* **248**, 110775 (2022).
- [24] 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).
- [25] A. D. Jagtap, Z. Mao, N. Adams, and G. E. Karniadakis, Physics-informed neural networks for inverse problems in supersonic flows, *J. Comput. Phys.* **466**, 111402 (2022).
- [26] H. Gao, L. Sun, and J. X. Wang, Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels, *Phys. Fluids* **33**, 073603 (2021).
- [27] D. Kelshaw, G. Rigas, and L. Magri, Physics-informed CNNs for super-resolution of sparse observations on dynamical systems, [arXiv:2210.17319](https://arxiv.org/abs/2210.17319).
- [28] G. Hasanuzzaman, H. Eivazi, S. Merbold, C. Egbers, and R. Vinuesa, Enhancement of PIV measurements via physics-informed neural networks, *Meas. Sci. Technol.* **34**, 044002 (2023).
- [29] L. Sliwinski and G. Rigas, Mean flow reconstruction of unsteady flows using physics-informed neural networks, *Data-Centric Eng.* **4**, e4 (2023).
- [30] J. P. Molnar, L. Venkatakrishnan, B. E. Schmidt, T. A. Sipkens, and S. J. Grauer, Estimating density, velocity, and pressure fields in supersonic flows using physics-informed BOS, *Exp. Fluids* **64**, 14 (2023).
- [31] A. D. Jagtap, E. Kharazmi, and G. E. Karniadakis, Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems, *Comput. Methods Appl. Mech. Eng.* **365**, 113028 (2020).
- [32] A. D. Jagtap and G. E. Karniadakis, Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations, *Commun. Comput. Phys.* **28**, 2002 (2020).
- [33] K. Shukla, A. D. Jagtap, and G. E. Karniadakis, Parallel physics-informed neural networks via domain decomposition, *J. Comput. Phys.* **447**, 110683 (2021).
- [34] Z. Hu, A. D. Jagtap, G. E. Karniadakis, and K. Kawaguchi, When do extended physics-informed neural networks (XPINNs) improve generalization? *SIAM J. Sci. Comput.* **44**, A3158 (2022).

- [35] S. Mishra and R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs, *IMA J. Numer. Anal.* **42**, 981 (2022).
- [36] T. De Ryck, A. D. Jagtap, and S. Mishra, Error estimates for physics-informed neural networks approximating the Navier–Stokes equations, *IMA J. Numer. Anal.* **44**, 83 (2023).
- [37] Y. Du, M. Wang, and T. A. Zaki, State estimation in minimal turbulent channel flow: A comparative study of 4DVar and PINN, *Int. J. Heat Fluid Flow* **99**, 109073 (2023).
- [38] V. Mons, O. Marquet, B. Leclaire, P. Cornic, and F. Champagnat, Dense velocity, pressure and Eulerian acceleration fields from single-instant scattered velocities through Navier–Stokes-based data assimilation, *Meas. Sci. Technol.* **33**, 124004 (2022).
- [39] The data is available from <https://github.com/xiaoh/para-database-for-PIML>.
- [40] S. Laizet and N. Li, Incompact3d: A powerful tool to tackle turbulence problems with up to O(10⁵) computational cores, *Int. J. Numer. Methods Fluids* **67**, 1735 (2011).
- [41] H. Xiao, J. L. Wu, S. Laizet, and L. Duan, Flows over periodic hills of parameterized geometries: A dataset for data-driven turbulence modeling from direct simulations, *Comput. Fluids* **200**, 104431 (2020).
- [42] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, DeepXDE: A deep learning library for solving differential equations, *SIAM Rev.* **63**, 208 (2021).
- [43] https://github.com/RigasLab/PINN_SA.
- [44] F. Hecht, New development in FreeFem++, *J. Numer. Math.* **20**, 251 (2012).
- [45] A. N. Brooks and T. J. R. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, *Comput. Methods Appl. Mech. Eng.* **32**, 199 (1982).
- [46] M. Olshanskii, G. Lube, T. Heister, and J. Löwe, Grad–div stabilization and subgrid pressure models for the incompressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Eng.* **198**, 3975 (2009).
- [47] J. Nocedal, Updating quasi-Newton matrices with limited storage, *Math. Comp.* **35**, 773 (1980).
- [48] M. J. Kochenderfer and T. A. Wheeler, *Algorithms for Optimization* (MIT Press, Cambridge, MA, 2019).
- [49] 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).
- [50] S. Wang, Y. Teng, and P. Perdikaris, Understanding and mitigating gradient flow pathologies in physics-informed neural networks, *SIAM J. Sci. Comput.* **43**, A3055 (2021).
- [51] R. van der Meer, C. W. Oosterlee, and A. Borovykh, Optimally weighted loss functions for solving pdes with neural networks, *J. Comput. Appl. Math.* **405**, 113887 (2022).
- [52] Q. Zeng, S. H. Bryngelson, and F. Schäfer, Competitive physics informed networks, [arXiv:2204.11144](https://arxiv.org/abs/2204.11144).
- [53] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks, *Proc. Math. Phys. Eng. Sci.* **476**, 20200334 (2020).