Uncovering turbulent plasma dynamics via deep learning from partial observations

A. Mathews[1],* M. Francisquez[1,2], J.W. Hughes[1], D.R. Hatch[3], B. Zhu[4], and B.N. Rogers[5]
[1]*MIT Plasma Science and Fusion Center, Cambridge, Massachusetts 02139, USA*
[2]*Princeton Plasma Physics Laboratory, Princeton, New Jersey 08540, USA*
[3]*Institute for Fusion Studies, University of Texas, Austin, Texas 78704, USA*
[4]*Lawrence Livermore National Laboratory, Livermore, California 94550, USA and*
[5]*Department of Physics and Astronomy, Dartmouth College, Hanover, New Hampshire 03755, USA*
(Dated: August 13, 2021)

One of the most intensely studied aspects of magnetic confinement fusion is edge plasma turbulence which is critical to reactor performance and operation. Drift-reduced Braginskii two-fluid theory has for decades been widely applied to model boundary plasmas with varying success. Towards better understanding edge turbulence in both theory and experiment, we demonstrate that a novel multi-network physics-informed deep learning framework constrained by partial differential equations can accurately learn turbulent fields consistent with the two-fluid theory from partial observations of electron pressure which is not otherwise possible using conventional equilibrium models. This technique presents a novel paradigm for the advanced design of plasma diagnostics and validation of magnetized plasma turbulence theories in challenging thermonuclear environments.

I. INTRODUCTION

Predicting turbulent transport in the edge of magnetic confinement fusion devices is a longstanding goal spanning several decades currently presenting significant uncertainties in the particle and energy confinement of fusion power plants [1, 2]. The boundary region is critical in determining the fusion device's overall viability since edge plasma conditions strongly influence a myriad of reactor operations ranging from core fuelling to power output to wall safety [3–5]. Yet edge modelling continues to need improvement—comprehensive gyrokinetic codes suitable for the boundary of tokamaks are still under development and fluid simulations commonly lack essential physics necessary to study collisionless systems. One particular transport theory relevant to boundary plasmas and widely applied to analyze edge turbulence is the drift-reduced Braginskii model [6–8]. Various adaptations of these equations have been recently taken to investigate several important edge phenomena including pedestal physics [9], blob dynamics [10], neutral effects [11], and heat fluxes impinging plasma-facing components [12]. And while various trends are at times reproduced in these works, direct quantitative agreement between the two-fluid turbulence theory and observations is generally lacking on a wide scale due to difficulty in aligning global simulations with intricate plasma experiments where relevant measurements may be sparse or missing altogether. Yet this is a crucially important endeavour since gathering sufficient information to effectively test reduced turbulent transport models is vital to developing predictive capability for future fusion devices. These machines will access novel burning plasma regimes and operate with some of the largest temperature gradients in the universe, but our existing models may be inaccurate and standard diagnostics incapable of surviving such harsh thermonuclear environments [13]. Fusion plasma diagnostic measurements accordingly require suitable analysis techniques as they are inherently noisy and limited in their spatiotemporal scope such as 1- or 2-dimensional profiles of electron density and temperature [14–17]. Furthermore, it is essential to know the turbulence-generated edge electric field since $\mathbf{E} \times \mathbf{B}$ drifts, which strongly affect perpendicular transport, influence plasma stability and structure across the edge [18–20]. Resultant downstream interactions from turbulent particle and heat fluxes striking surfaces therein pose significant risk to safe operation, e.g. the control of radiative regimes such as detached divertor schemes, in upcoming fusion reactors [21, 22]. To this end, we demonstrate a novel physics-informed deep learning technique capable of diagnosing unknown turbulent field fluctuations directly consistent with drift-reduced Braginskii theory from limited electron pressure observations.

We represent the drift-reduced Braginskii model via physics-informed neural networks (PINNs) [23–26]—highly expressive function approximators trained to solve supervised learning tasks while respecting nonlinear partial differential equations—to infer unobserved field dynamics from partial measurements of a synthetic plasma. As illustrated through a readily adaptable multi-network machine learning framework, this paradigm is transferable to the broad study of quasineutral plasmas in magnetized collisional environments and presents novel pathways for the AI-assisted design of plasma diagnostics. In ways previously inaccessible with classical analytic methods, this framework has the ability to improve the direct testing of reduced turbulence models in both experiment and simulation to inform discovery of the equations necessary to model the edge. The overall computational technique introduces significant advances in systematizing the quantitative validation of plasma turbulence theories and is to date among the most complex nonlinear systems applied in physics-informed deep learning codes.

* mathewsa@mit.edu

To demonstrate this framework, we proceed with a description of drift-reduced Braginskii theory and computational modelling of a synthetic plasma in Section II, outline a novel physics-informed machine learning architecture suited for the analysis of multi-field plasma turbulence in Section III, present results in the robust learning of unknown turbulent fields in section IV, and conclude with a summary and future outlook in Section V.

II. SYNTHETIC PLASMA CONSTRUCTION

The synthetic plasma analyzed is numerically simulated by the global drift-ballooning (GDB) finite difference code [27, 28] which solves the two-fluid drift-reduced Braginskii equations in the electrostatic limit relevant to low-β conditions. This is a full-f [29–31] fluid model in the sense that the evolution of the equilibrium and fluctuating components of the solution are not separated and relative perturbation amplitudes can be of order unity as found in experiments [32]. The plasma is magnetized and quasineutral with the perpendicular fluid velocity given by $\mathbf{E}\times\mathbf{B}$, diamagnetic, and ion polarization drifts. After neglecting collisional drifts, as well as terms of order m_e/m_i, one arrives at the following set of equations (in Gaussian units) governing the evolution of the synthetic plasma's density ($n \approx n_e \approx n_i$), vorticity ($\omega$), parallel electron velocity ($v_{\parallel e}$), parallel ion velocity ($v_{\parallel i}$), electron temperature (T_e), and ion temperature (T_i) [28]

$$\frac{d^e n}{dt} = -\frac{2c}{B}\left[nC_{(\phi)} - \frac{1}{e}C_{(p_e)}\right] - n\nabla_\parallel v_{\parallel e} + S_n + \mathcal{D}_n \quad (1)$$

$$\begin{aligned}\frac{\partial\omega}{\partial t} &= \frac{2c}{eB}\left[C_{(p_e)} + C_{(p_i)}\right] - \frac{1}{em_i\Omega_i}C_{(G_i)} \\ &+ \frac{1}{e}\nabla_\parallel j_\parallel - \nabla\cdot\left\{\frac{nc^2}{\Omega_i B^2}\left[\phi, \nabla_\perp\phi + \frac{\nabla_\perp p_i}{en}\right]\right. \\ &\left. + \frac{nc}{\Omega_i B}v_{\parallel i}\nabla_\parallel\left(\nabla_\perp\phi + \frac{\nabla_\perp p_i}{en}\right)\right\} + \mathcal{D}_\omega\end{aligned} \quad (2)$$

$$\begin{aligned}\frac{d^e v_{\parallel e}}{dt} &= \frac{1}{m_e}\left(e\nabla_\parallel\phi - \frac{\nabla_\parallel p_e}{n} - 0.71\nabla_\parallel T_e + e\eta_\parallel j_\parallel\right) \\ &+ \frac{2}{3}\frac{\nabla_\parallel G_e}{n} + \frac{2cT_e}{eB}C_{(v_{\parallel e})} + S_{\mathcal{M}\parallel e} + \mathcal{D}_{v_{\parallel e}}\end{aligned} \quad (3)$$

$$\begin{aligned}\frac{d^i v_{\parallel i}}{dt} &= \frac{1}{m_i}\left(-e\nabla_\parallel\phi - \frac{\nabla_\parallel p_i}{n} + 0.71\nabla_\parallel T_e - e\eta_\parallel j_\parallel\right) \\ &+ \frac{2T_e}{3n}\frac{\nabla_\parallel G_i}{n} - \frac{2cT_i}{eB}C_{(v_{\parallel i})} + S_{\mathcal{M}\parallel i} + \mathcal{D}_{v_{\parallel i}}\end{aligned} \quad (4)$$

$$\begin{aligned}\frac{d^e T_e}{dt} &= \frac{2T_e}{3n}\left[\frac{d^e n}{dt} + \frac{1}{T_e}\nabla_\parallel\kappa_\parallel^e\nabla_\parallel T_e + \frac{5n}{m_e\Omega_e}C_{(T_e)}\right. \\ &\left. + \eta_\parallel\frac{j_\parallel^2}{T_e} + \frac{0.71}{e}(\nabla_\parallel j_\parallel - \frac{j_\parallel}{T_e}\nabla_\parallel T_e) + \frac{1}{T_e}S_{E,e}\right] + \mathcal{D}_{T_e},\end{aligned} \quad (5)$$

$$\begin{aligned}\frac{d^i T_i}{dt} &= \frac{2T_i}{3n}\left[\frac{d^i n}{dt} + \frac{1}{T_i}\nabla_\parallel\kappa_\parallel^i\nabla_\parallel T_i\right. \\ &\left. - \frac{5n}{m_i\Omega_i}C_{(T_i)} + \frac{1}{T_i}S_{E,i}\right] + \mathcal{D}_{T_i}\end{aligned} \quad (6)$$

whereby the field-aligned electric current density is $j_\parallel = en\left(v_{\parallel i} - v_{\parallel e}\right)$, the stress tensor's gyroviscous terms contain $G_s = \eta_0^s\left\{2\nabla_\parallel v_{\parallel s} + c\left[C_{(\phi)} + C_{(p_s)}/(q_s n)\right]\right\}$, and η_0^s, Ω_s, and q_s are the species ($s = \{e, i\}$) viscosity, cyclotron frequency, and electric charge, respectively. The convective derivatives are $d^s f/dt = \partial_t f + (c/B)\left[\phi, f\right] + v_{\parallel s}\nabla_\parallel f$ with $[F, G] = \mathbf{b_0}\times\nabla F\cdot\nabla G$ and $\mathbf{b_0}$ representing the unit vector parallel to the magnetic field. The field's magnitude, B, decreases over the major radius of the torus ($B \propto 1/R$), and its curvature is $\boldsymbol{\kappa} = -\hat{\mathbf{R}}/R$. The curvature operator, $C_{(f)} = \mathbf{b_0}\times\boldsymbol{\kappa}\cdot\nabla f$, $\nabla_\parallel = -\partial/\partial z$, and $\mathbf{b_0} = -\hat{\mathbf{z}}$ follow past convention [28]. The coefficients $\kappa_\parallel^s$ and $\eta_\parallel^s$ correspond to parallel thermal conductivity and electrical resistivity, respectively. Time-independent Gaussian density (S_n) and energy sources ($S_{E,s}$) are placed at the left wall while zero external momentum ($S_{\mathcal{M}\parallel s}$) is explicitly forced upon the system. Explicit hyperdiffusion consisting of both fourth-order cross-field and second-order parallel diffusion is applied for numerical stability in the form of $\mathcal{D}_f = \chi_x\frac{\partial f}{\partial x^4} + \chi_y\frac{\partial f}{\partial y^4} + \chi_z\frac{\partial f}{\partial z^2}$. Under quasineutrality, electric fields arise not by local imbalance of charged particles but by the requirement that the electric current density is divergence free [33, 34]. Accordingly, the electrostatic potential, ϕ, is numerically solved via the following boundary value problem:

$$\nabla\cdot\frac{nc}{\Omega_i B}\left(\nabla_\perp\phi + \frac{\nabla_\perp p_i}{en}\right) = \omega. \quad (7)$$

The 3-dimensional simulation domain implemented is a shearless field-aligned coordinate system where $\hat{\mathbf{x}}$ is the unit vector along the radial direction (i.e. $\hat{\mathbf{R}}$), the helical magnetic field is oriented along $\hat{\mathbf{z}}$, and $\hat{\mathbf{y}}$ is perpendicular to both $\hat{\mathbf{x}}$ and $\hat{\mathbf{z}}$. The synthetic plasma consists of deuterium ions and electrons with real masses (i.e. $m_i = 3.34\times10^{-27}$ kg and $m_e = 9.11\times10^{-31}$ kg) and on-axis magnetic field of $B_{axis} = 5.0$ T with minor and major radius of $a_0 = 0.22$ m and $R_0 = 0.68$ m, respectively, consistent with characteristics of discharges in the high-field Alcator C-Mod tokamak [35] for which there is notable evidence of fluid drift turbulence controlling edge dynamics [36]. Moreover, drift-reduced fluid models, where the ion gyration frequency is considered to be faster than the evolving turbulent plasma fluctuations (i.e. $\Omega_i \gg \frac{\partial}{\partial t}$), are generally good approximations to full velocity models when studying edge turbulence [37].

This discretized toroidal geometry is a flux-tube-like domain on the outboard side (i.e. strictly bad-curvature) of the tokamak with field lines of constant helicity wrapping around the torus and terminating on walls producing both resistive interchange and toroidal drift-wave turbulence. Transport is primarily along blobby field-aligned structures with increased pressure propagating

due to perpendicular drifts which polarize the blob and yield outward $\mathbf{E}\times\mathbf{B}$ drift of the filament. This is related to the Poynting vector representing the directional energy flux density of the electromagnetic field [33, 38]. The physical dimensions of the entire simulation domain are $[L_x = 7.7\text{ cm}, L_y = 5.5\text{ cm}, L_z = 1800.0\text{ cm}]$ with spatiotemporal resolution of $[\Delta x = 0.03\text{ cm}, \Delta y = 0.04\text{ cm}, \Delta z = 56.25\text{ cm}, \Delta t = 4.55\times10^{-11}\text{ s}]$. Periodic boundary conditions are employed in the binormal direction for all quantities. Homogeneous Neumann conditions are set in the radial coordinate for n, $v_{\parallel e}$, $v_{\parallel i}$, T_e, and T_i while homogeneous Dirichlet conditions are used for ω and ϕ. By constraining $\phi = 0$ along the walls, this in principle enforces radial $\mathbf{E}\times\mathbf{B}$ flows to go to zero on the boundaries of the simulation domain. The lower limit of the Bohm criterion is imposed as a sheath condition,

$$v_{\parallel i}(z = \pm\frac{L_z}{2}) = \mp c_s = -\sqrt{\frac{T_i + T_e}{m_i}}, \tag{8}$$

$$v_{\parallel e}(z = \frac{\pm L_z}{2}) = \begin{cases} \mp c_s \exp(\Lambda - \frac{e\phi}{T_e}) & \text{if } \phi > 0 \\ \mp c_s \exp(\Lambda) & \text{if } \phi \leq 0 \end{cases}, \tag{9}$$

where $\Lambda = \log\sqrt{m_i/[2\pi m_e(1+\frac{T_i}{T_e})]}$. Since the direction of the flows at the sheath entrance are known, ghost cells in the z-direction are filled such that an upwind stencil ensues to evolve n, ω, T_e, and T_i [28]. For T_e and T_i specifically, finite conductive heat fluxes entering the sheaths are applied according to $q_{\parallel,s} = -\kappa_\parallel^s \nabla_\parallel T_s = \pm\gamma_s n v_{\parallel,s} T_s$, where the upper (lower) sign corresponds to the top (bottom) sheath and γ_s is the sheath transmission coefficient. Its value for ions and electrons is taken to be $\gamma_i = 5T_i/2T_e$ and $\gamma_e = 2+|e\phi|/T_e$, respectively [28]. Collisional coefficients and diffusivities are kept constant in the direct numerical simulation as they can be unphysically large at high temperatures due to the lack of kinetic effects and generally require closures going beyond Chapman-Enskog. To start the numerical simulation, electrons and ions are initialized with zero parallel velocity and vorticity fields along with truncated Gaussian density and temperature profiles. A second-order trapezoidal leap-frog time-stepping scheme evolves the system of equations forward with subcycling of parabolic terms (e.g. $\nabla_\parallel \kappa_\parallel^s \nabla_\parallel T_s$) due to the low frequency turbulence structure changing slowly over the thermal diffusion timescale. The commonly applied Boussinesq approximation [39] in Braginskii solvers is also used when evolving the generalized vorticity, ω. The normalizations applied to solve these partial differential equations in both the finite difference code and deep learning framework are sketched in the Appendix. A complete treatment of the numerical solver and greater specificity regarding the turbulence simulations can be found in [27, 28].

III. MACHINE LEARNING PLASMA THEORY

Neural networks are operationally computational programs composed of elementary arithmetic operations (e.g. addition, multiplication) and functions (e.g. exp, sin, log) which can be differentiated to arbitrary order up to machine precision via application of chain rule [40, 41]. While biases are presently inevitable [42], these regression models are in theory constructed without necessarily committing to a designated class of basis functions (e.g. polynomial, trigonometric). Automatic differentiation in conjunction with this adaptive capacity of neural networks permits them to effectively address nonlinear optimization problems in physics and engineering by training upon both partial differential equations and observational data via multi-task learning [24]. Constraining classically underdetermined systems by physical laws and experimental measurements in this way presents an emerging technique in computational mechanics which we extend to the deduction of unknown turbulent plasma dynamics. In this deep learning framework, every dynamical variable in equations (1)–(6) is approximated by its own fully-connected neural network, which is commonly known as a data-efficient universal function approximator [43], since its high plasticity enables it to be molded given sufficient training.

FIG. 1. These 2-dimensional measurements of turbulent electron density and temperature over a short temporal window are the only observed variables from the 3-dimensional synthetic plasma exhibiting blob-like filaments.

For analysis in the multi-network framework, partial measurements of n_e and T_e over time only come from a smaller 2-dimensional field-aligned domain in the interior of the synthetic plasma to emulate experiment (e.g. gas puff imaging [44]) with dimensions of $[L_x^* =$

3.8 cm, $L_y^* = 3.8$ cm] and spatiotemporal resolution of [$\Delta^* x = 0.03$ cm, $\Delta^* y = 0.04$ cm, $\Delta^* t = 7.27 \times 10^{-7}$ s] as depicted by a snapshot in Figure 1. Each network consequently takes the local spatiotemporal points (x, y, t) from the reduced domain for measurements as the only inputs to the initial layer while the dynamical variable being represented is the sole output. In the middle of the architecture, every network consists of 5 hidden layers with 50 neurons per hidden layer and hyperbolic tangent activation functions (σ) using Xavier initialization [45].

FIG. 2. A local time trace of the turbulent n_e over 200 μs from the simulated plasma at [$x = 1.0$ cm, $y = 0.0$ cm, $z = -28.1$ cm]. The observed synthetic data analyzed in the machine learning framework only comes from the small temporal window (green) which corresponds to just 4 points in time.

In actuality, the repeated differentiation and summation of networks to construct every single term in the partial differential equations trained upon subsequently constructs a far larger resultant computation graph representing the collective model beyond the 5 hidden layers in each dynamical variable's individual network. The cumulative graph is therefore a truly deep approximation of the turbulent plasma theory. Physical constraints are learned by the networks via minimization of ascribed loss functions encompassing both limited measurements of the plasma and two-fluid turbulence model. In particular, partial observations of the simulated plasma consist of only n_e and T_e measurements of 2-dimensional spatial extent as visualized in Figure 1 over just 4 separate time slices (i.e. 2.9 μs). For reference, the synthetic plasma's fluctuations have an approximate autocorrelation time of 1.5 μs and radial autocorrelation length of 0.4 cm. The narrow temporal extent of the strongly fluctuating n_e observations at a local spatial point is further visualized in Figure 2. Properties of all other dynamical variables in the 6-field turbulence theory are taken to be unknown, and the networks are simultaneously optimized against the drift-reduced Braginskii equations and observed data to better approximate the unmeasured quantities. To be precise, partial synthetic measurements are learned by training the n_e and T_e networks against the average $\mathcal{L}_2$-norm of their respective relative errors

$$\mathcal{L}_{n_e} = \frac{1}{N_0} \sum_{i=1}^{N_0} |n_e^*(x_0^i, y_0^i, z_0^i, t_0^i) - n_{e,0}^i|^2, \tag{10}$$

$$\mathcal{L}_{T_e} = \frac{1}{N_0} \sum_{i=1}^{N_0} |T_e^*(x_0^i, y_0^i, z_0^i, t_0^i) - T_{e,0}^i|^2, \tag{11}$$

where $\{x_0^i, y_0^i, z_0^i, t_0^i, n_{e,0}^i, T_{e,0}^i\}_{i=1}^{N_0}$ correspond to the set of observed data and the variables n_e^* and T_e^* symbolize predicted electron density and temperature, respectively, by the networks. The theory enforcing physical constraints in the deep learning framework is expressed by evaluating the individual terms in the model by differentiating the neural networks with respect to input spatiotemporal coordinates via application of chain rule through automatic differentiation [46]. Correspondingly, model loss functions are embedded during training by recasting the evolution equations of (1) and (5) in the following implicit form

$$\begin{aligned} f_{n_e} \coloneqq & -\frac{d^e n}{dt} - \frac{2c}{B}\left[nC_{(\phi)} - \frac{1}{e}C_{(p_e)}\right] \\ & - n\nabla_\parallel v_{\parallel e} + S_n + \mathcal{D}_n, \end{aligned} \tag{12}$$

$$\begin{aligned} f_{T_e} \coloneqq & -\frac{d^e T_e}{dt} + \frac{2T_e}{3n}\left[\frac{d^e n}{dt} + \frac{1}{T_e}\nabla_\parallel \kappa_\parallel^e \nabla_\parallel T_e \right. \\ & + \frac{5n}{m_e \Omega_e} C_{(T_e)} + \eta_\parallel \frac{j_\parallel^2}{T_e} + \frac{0.71}{e}(\nabla_\parallel j_\parallel \\ & \left. - \frac{j_\parallel}{T_e}\nabla_\parallel T_e) + \frac{1}{T_e} S_{E,e}\right] + \mathcal{D}_{T_e}, \end{aligned} \tag{13}$$

and then further normalized into dimensionless form matching the numerical code as in (A4) and (A8) [28]. This normalized implicit formulation is vital to learning via optimization since all physical terms collectively sum to zero when the equations are ideally satisfied. These physical constraints provided by the unitless evolution equations of n_e and T_e from the two-fluid model are jointly optimized using loss functions defined by

$$\mathcal{L}_{f_{n_e}} = \frac{1}{N_f} \sum_{i=1}^{N_f} |f_{n_e}^*(x_f^i, y_f^i, z_f^i, t_f^i)|^2, \tag{14}$$

$$\mathcal{L}_{f_{T_e}} = \frac{1}{N_f} \sum_{i=1}^{N_f} |f_{T_e}^*(x_f^i, y_f^i, z_f^i, t_f^i)|^2, \tag{15}$$

where $\{x_f^i, y_f^i, z_f^i, t_f^i\}_{i=1}^{N_f}$ denote the set of collocation points, and $f_{n_e}^*$ and $f_{T_e}^*$ are the null partial differential equations prescribed by (12) and (13) in normalized

FIG. 3. Visualization of the physics-informed framework with individual networks being selectively trained against loss functions comprising both partial observations, $\mathcal{L}_{n_e}$ and $\mathcal{L}_{T_e}$, and reduced theory, $\mathcal{L}_{f_{n_e}}$ and $\mathcal{L}_{f_{T_e}}$, to infer unobserved turbulent dynamics. All spatial gradients and time derivatives in f_{n_e} and f_{T_e} are represented using automatic differentiation of each individual variable's network which in practice extends the size of the computation graph being evaluated. To handle reduced 2-dimensional data from the 3-dimensional synthetic plasma, the z-coordinate is removed from the networks for simplicity and as a test for determining the minimal information necessary to learn ϕ. If noisy data is observed, then θ_{n_e} (and θ_{T_e} if T_e measurements are available) should be additionally trained against $\mathcal{L}_{f_{n_e}}$ (and $\mathcal{L}_{f_{T_e}}$).

form directly evaluated by the neural networks. Optimization against the applied plasma theory is central to the methodology and enforces physical constraints in the deep learning framework by ensuring each sub-network respects the multi-field turbulence model's constraints as visualized in Figure 3. This enables fine tuning of each neural networks' weights and biases (θ) by adjusting them in this generalized regression model to satisfy the physical laws governing the nonlinear connection sought between the sub-networks. The set of collocation points over which the partial differential equations are evaluated can be arbitrarily large and span any extent over the physical domain, but are taken in this example to correspond to the positions of the synthetic measurements being trained upon, i.e. $\{x_0^i, y_0^i, z_0^i, t_0^i\}_{i=1}^{N_0} = \{x_f^i, y_f^i, z_f^i, t_f^i\}_{i=1}^{N_f}$. It should be once again noted that the only observed dynamical quantities in these equations are 2-dimensional views of n_e and T_e without any explicit information about boundary conditions nor initializations. All analytic terms encoded in these equations including high-order operators are computed exactly by the neural networks without any approximation (e.g. linearization) nor discretization. This machine learning framework with a collocation grid of arbitrarily high resolution uses a continuous spatiotemporal domain without time-stepping nor finite difference schema in contrast with the numerical code. To handle 2-dimensional data, we assume slow variation of dynamics in the z-coordinate and effectively set all parallel derivatives to zero (i.e. $\frac{\partial}{\partial z} \to 0$). Notwithstanding, parallel flows and Ohmic heating terms in the model are still kept. If measurements in the z-direction are available or more collocation points utilized during training with observational data of reduced dimensionality, this procedure may be relaxed—it is partly a trade-off between computational fidelity and stability. It is noteworthy that the temporal resolution of the data observed by the neural networks is several orders of magnitude lower than the time steps taken by the finite difference solver as required for numerical stability, i.e. $\Delta^* t \gg \Delta t$. Also, if sought, training on data sets viewed at oblique angles in 3-dimensional space over long macroscopic timescales can be easily performed via segmentation of the domain and parallelization, but a limited spatial view with reduced dimensionality is taken to emulate experimental conditions for field-aligned observations [44] and theoretically test what information is indispensable to learn unobserved turbulent dynamics.

Loss functions are optimized with mini-batch sampling where $N_0 = N_f = 500$ using stochastic gradient descent via Adam [47] and the limited-memory Broy-

den–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm—a quasi-Newton optimization algorithm [48]— for 20 hours over 32 cores on Intel Haswell-EP processors which corresponds to approximately 8694 full iterations over both optimizers. If observing noisy data, we find that expanding to larger sample sizes with $N_0 = N_f = 2500$ and training solely via L-BFGS is optimal for learning. Removing $\mathcal{L}_{f_{n_e}}$ and $\mathcal{L}_{f_{T_e}}$ from the optimization process (i.e. setting $N_f = 0$) would correspond to training of classical neural networks without any knowledge of the underlying governing equations which would then be incapable of learning turbulent field fluctuations. Setting $N_0 = 0$ instead while providing initial and boundary conditions for all dynamical variables would alternatively correspond to regularly solving the equations directly via neural networks. Overall, priming networks by firstly training in stages on observed data or prior constraints is useful to enhance stability and convergence in this multi-objective task. Additionally, encoding domain expertise such as subsonic bounds on parallel flows or regularizing temperature to be strictly positive via suitable output activation functions can assist training by constraining the admissible landscape of solutions. Networks constructed in this way can intrinsically abide by physical laws which is especially useful to uncover unknowns like $v_{\|e}$ and T_i.

A fundamental goal in computational plasma modelling is determining the minimum complexity necessary (and no less) to develop sufficiently predictive tokamak simulations. With sparse availability of measurements in fusion experiments, designing diagnostic techniques for uncovering such information is crucial. On this point, we emphasize that training is from scratch over just a single synthetic plasma discharge with no extraneous validation nor testing sets required since overfitting is technically not encountered in this physics-informed paradigm. The multi-network deep learning framework simply utilizes a single set of n_e and T_e measurements over a period of microseconds which corresponds to the small data regime of machine learning. Merging partial observational data of n_e and T_e along with physical laws in the form of partial differential equations governing the time-dependent evolution of n_e and T_e sufficiently constrains the set of admissible solutions for the previously unknown nonlinear mappings the neural networks ultimately learn. It is also quite general: due to quasineutrality, no significant adjustments are necessary to generalize the technique when multiple ions and impurities may be present in boundary plasmas beyond the inclusion of appropriate collisional drifts and sources in multi-species plasmas [49]. This deep learning technique for diagnosing turbulent fields is hence easily transferable which permits its systematic application across magnetic confinement fusion experiments whereby the underlying physical model fundamental to the turbulent transport is consistent. The framework sketched can also be readily extended to different settings in the interdisciplinary study (both numerical and experimental) of magnetized collisional plasmas in propulsion engines and astrophysical environments.

IV. NUMERICAL EXPERIMENTS

Accurate turbulent edge electric field fluctuation characterization is particularly significant to magnetic confinement fusion devices. By constraining the deep learning framework with the two-fluid turbulence theory and strikingly little empirical information in the form of partial 2-dimensional observations of n_e and T_e, we find that physics-informed neural networks can accurately learn the plasma's turbulent electric potential without the physics-informed computational framework ever having observed it, as displayed in Figures 4 and 5.

FIG. 4. The synthetic plasma's unobserved electric potential (top) is learned approximately up to an additive constant as predicted by the physics-informed neural network (bottom).

FIG. 5. 1-dimensional radial profile of the true and predicted ϕ at $[y = 0.0$ cm$, z = -28.1$ cm$, t = 229.9\ \mu$s$]$ corresponds to a slice of Figure 4. The ordinates have identical ranges spanning exactly 410 V with equivalent spacing for direct comparison.

It is notable that despite there being no knowledge of $\omega, v_{\|e}, v_{\|i}$, nor T_i (i.e. multiple unknowns existing

in the partial differential equations and (7) never even being directly invoked), the electric field is nonetheless learned consistently with the physical theory encoded by the plasma turbulence model. Since ϕ is a gauge-invariant quantity exact up to an additive constant, it is accordingly uncovered up to a scalar offset which varies randomly due to the stochastic nature of the optimization employed in the machine learning framework. This difference physically arises because no direct boundary information was enforced upon the neural network when learning ϕ, although it could be straightforwardly implemented. By contrast, the numerical code imposed zero potential on the outer walls of the simulation domain. General agreement in both magnitude and structure in the learned radial electric field is evident in Figure 6 with an average absolute error of 26.19 V/cm while $\mathcal{L}_{f_{n_e}}$ and $\mathcal{L}_{f_{T_e}}$ are 3.101×10^{-1} and 3.743×10^{-1}, respectively, after 8694 full iterations. In physical units, $\mathcal{L}_{f_{n_e}}$ and $\mathcal{L}_{f_{T_e}}$ would be 3.472×10^{17} cm^{-3}/s and 1.273×10^{6} eV/s.

FIG. 6. The learned turbulent E_r (bottom) closely agrees with the magnitude and structure of the true E_r (top) despite ω, $v_{\|e}$, $v_{\|i}$, and T_i being initially unknown.

To emphasize the experimental practicality of these results based upon our synthetic plasma and a surprising feature discovered, we note that the framework is robust to very noisy measurements. For example, if only observing density fluctuations with strong Gaussian noise, one could still largely recover the unmeasured radial and poloidal electric fields and even resolve the true partially observed variables. Namely, given just highly noisy n_e measurements as in Figure 7, one can learn the true turbulent density in this physics-informed deep learning framework to subsequently infer the unmeasured electric field. If E_r was already known, this technique could then precisely check the validity of the reduced turbulence theory against observations from experiment or kinetic simulations [50]. But, if using a standard feed-forward neural network, one must be careful with convergence since the objective of simply minimizing $\mathcal{L}_{n_e}$ without sufficient regularization, as innately provided by $\mathcal{L}_{f_{n_e}}$, can result in overfitting of noisy data.

For comparison, classical and oft-employed models for calculating the electric potential with adiabatic electrons such as the Boltzmann relation [51] fail when computing perpendicular turbulent field fluctuations. Alternative approximations of E_r from simple ion pressure balance as expected neoclassically [52] would yield highly incorrect estimates of the turbulent electric field, too. Such methods ordinarily used in magnetic confinement fusion are only applicable to discerning equilibrium fields and dynamics parallel to the magnetic field in steady-state scenarios, but are erroneous in the analysis of microturbulence in nonquiescent plasmas as markedly observed when comparing Figure 8 to the true ϕ and E_r as plotted in Figures 4 and 6, respectively. Our deep learning technique based upon drift-reduced Braginskii theory therefore provides a novel way to accurately measure the turbulent electric field in edge plasmas from just the electron pressure. As a further point of contrast compared to classical techniques, it is important to note that the inverse learning scenario is not possible. In particular, given observations of ϕ and T_e, one cannot simply infer the turbulent n_e fluctuations with the machine learning framework outlined. This one-way nature in learning indicates a division exists between the two pathways when attempting to constrain the admissible solutions of (14) and (15) to uncover unknown nonequilibrium dynamics. Training is thus unidirectional and representative of asymmetries extant in the partial data and turbulence theory being learnt via optimization in the physics-informed model.

Observational and physical model errors				
M	$\mathcal{L}_{n_e}$ (10)	$\mathcal{L}_{T_e}$ (11)	$\mathcal{L}_{f_{n_e}}$ (14)	$\mathcal{L}_{f_{T_e}}$ (15)
1	1.498×10^{-2}	7.258×10^{-2}	8.316×10^{-2}	1.476×10^{-1}
10	7.673×10^{-3}	8.525×10^{-3}	2.049×10^{-1}	1.302×10^{-0}
100	9.334×10^{-4}	6.592×10^{-4}	5.746×10^{-1}	1.895×10^{-0}
1000	6.489×10^{-5}	3.203×10^{-5}	4.123×10^{-1}	9.171×10^{-1}
8694	4.715×10^{-6}	3.906×10^{-6}	3.101×10^{-1}	3.743×10^{-1}

TABLE I. Each normalized loss function optimized in the machine learning framework is tabulated after M full iterations, where $M = 8694$ corresponds to the final iteration after 20 hours of training against both the partial observations of n_e and T_e and their implicit evolution equations.

To better interpret the learning process, we tabulate the normalized loss functions being trained upon after M full iterations by the optimizers in Table I. After 1 iteration, (14) and (15) are relatively small in magnitude, and this would correspond to a trivial result satisfying the partial differential equations given the nonuniqueness of its solutions. As training progresses, observational data is better captured in the deep learning framework and the neural networks proceed to converge toward the

FIG. 7. The physics-informed deep learning framework is capable of recovering the true n_e despite strong Gaussian noise ($\sigma = 0.25$) present. The classical solution corresponds to a standard feed-forward neural network where $N_f = 0$.

FIG. 8. Estimates of the turbulent ϕ and E_r as expected by the Boltzmann model, $n_e(\phi) = n_e(\phi_0)e^{e(\phi-\phi_0)/T_e}$ [51], or neoclassical estimates, $\nabla\phi = -\nabla p_i/Zn_ie$ [52], where $Z = 1$ for deuterium ions, yield markedly errant predictions in both structure and magnitude when compared to the true values displayed at the top of Figures 4 and 6.

sought solution as opposed to trivial ones. A difference in the rates of learning for n_e, T_e, and ϕ also exist since the electric field is learned implicitly via the model instead of being trained upon directly. Each individual loss function being optimized therefore does not necessarily decrease perfectly monotonically, but it is instead the collective training against partial differential equations in conjunction with observational data that is key. Namely, while there are many potential solutions to (12) and (13)—and while they may be more easily satisfied by trivial solutions—the limited n_e and T_e measurements compel the optimizer towards the physical solution of the partially observed plasma. In scenarios where inconsistencies in the true and learned model E_r exist, one can repurpose this machine learning framework to iteratively test and thereby discover the *correct* partial differential equations altogether by quantitatively examining the proposed model's consistency with observations as in Table I. For example, the analytic form of reduced source models in fluid theories [11, 38] can be inserted in the physics-informed deep learning framework to account for local turbulent ionization and inelastic collisions with kinetic neutrals by observing such measurements of n_e, T_e, and ϕ in global simulations [53] and experiments [17].

V. CONCLUSION

These results illustrate a custom physics-informed deep learning paradigm with the novel capacity to learn unknown nonequilibrium dynamics in a multi-field turbulent transport model broadly relevant to magnetized collisional plasmas. We specifically demonstrate the ability to determine unobserved turbulent electric fields consistent with the drift-reduced Braginskii equations from partial electron pressure observations in contrast with with standard analytic techniques. This can be applied to infer field fluctuations that may be difficult to measure or when sought plasma diagnostics are simply lacking. On the other hand, if experimental electric field measurements exist, then the quantitative validity of the plasma turbulence model embedded in the neural networks can be expressly assessed. This technique is also quite robust since, due to quasineutrality, it can be used to study

ionized gases in magnetized environments with multiple ions and impurities present as commonly found in astrophysical settings and fusion energy and space propulsion systems. From a mathematical physics standpoint, it is significant that nonlinear dynamics can be accurately recovered from partial data and theory in a 6-field model. Inferring completely unknown turbulent fields from just 2-dimensional measurements and representations of the evolution equations given by (12) and (13) demonstrates a massive reduction in the original 3-dimensional turbulence model indicating redundancy and the existence of reduced theory characterizations. Going forward, this framework has the capability to be generalized (e.g. to learn T_e, T_i, $v_{\parallel e}$, and $v_{\parallel i}$ in addition to ϕ using just 1-dimensional n_e measurements) and transform how turbulence theories are systematically and quantitatively validated in both plasma simulations and experiments. The interpretable physics-informed methodology outlined is also transferable across models (e.g. collisionless fluids, gyrokinetic, electromagnetic, atomic physics) and complex geometries. Furthermore, known limitations and *unknown* corrections to Braginskii's theory exist [54] which can be introduced in the deep learning framework to automate efficient testing and discovery of reduced plasma turbulence models when high fidelity data is observed. These extensions in theory and computing will be focused upon in future works.

Appendix A: Normalization of plasma theory

Converting the drift-reduced Braginskii equations from physical units to a normalized form is essential to numerically solve the model in both finite difference solvers and physics-informed machine learning codes. For completeness, the full normalization procedure is carried out whereby the physical variables and all associated quantities are transformed according to [28]

$$\begin{aligned} n &\leftarrow n/n_0, & T_s &\leftarrow T_s/T_{s0}, \\ \phi &\leftarrow \phi/\phi_0, & v_{\parallel s} &\leftarrow v_{\parallel s}/c_{s0}, \end{aligned} \tag{A1}$$

where $n_0 = 5 \times 10^{19}\ \mathrm{m}^{-3}$, $T_{s0} = 25$ eV, $c_{s0}^2 = T_{s0}/m_i$, $\phi_0 = B_0 a^2/ct_0$, and $t_0 = \sqrt{a_0 R_c/2}/c_{e0}$ is the interchange-like reference timescale. To match our simulation with experimental edge parameters of the Alcator C-Mod tokamak, $B_0 = B_{axis}R_0/(R_0 + a_0)$ and $R_c = R_0 + a_0$. This in turn defines the following dimensionless constants

$$\begin{aligned} \epsilon_R &= \frac{2a}{R_c}, & \kappa^i &= 3.9\frac{2}{3}\frac{T_{i0}t_0}{m_i R_c^2 \nu_{i0}}, & \alpha_d &= \frac{T_{e0}ct_0}{eB_0a^2}, \\ \epsilon_v &= \frac{c_{e0}t_0}{R_c}, & \eta &= 0.51\nu_{e0}t_0, & \epsilon_G &= \frac{0.08\tau}{\nu_{i0}t_0}, \\ \tau &= \frac{T_{i0}}{T_{e0}}, & \kappa^e &= 3.2\frac{2}{3}\frac{T_{e0}t_0}{\nu_{e0}m_e R_c^2}, & \epsilon_{Ge} &= \frac{0.73}{12\nu_{e0}t_0} \end{aligned} \tag{A2}$$

where c and ν_{s0} denote the speed of light and collision rate [55], respectively. The spatiotemporal coordinates are normalized by the following conversions

$$\begin{aligned} x &\leftarrow x/a_0, & y &\leftarrow y/a_0, \\ z &\leftarrow z/R_0, & t &\leftarrow t/t_0. \end{aligned} \tag{A3}$$

Under these set of transformations, the normalized two-fluid equations numerically solved are

$$\begin{aligned} \frac{d^e \ln n}{dt} = &-\epsilon_R\left[C_{(\phi)} - \alpha_d\frac{C_{(p_e)}}{n}\right] \\ &- \epsilon_v\nabla_\parallel v_{\parallel e} + \frac{1}{n}S_n + \mathcal{D}_{\ln n} \end{aligned} \tag{A4}$$

$$\begin{aligned} \frac{\partial\omega}{\partial t} = &C_{(p_e)} + \tau C_{(p_i)} + \frac{\epsilon_v}{\alpha_d\epsilon_R}\nabla_\parallel j_\parallel - \epsilon_G C_{(G_i)} \\ &- \nabla\cdot\{\frac{n}{B^3}\left[\phi, \nabla_\perp\phi + \tau\alpha_d\frac{\nabla_\perp p_i}{n}\right] + \\ &\sqrt{\tau}\epsilon_v\frac{n}{B^2}v_{\parallel i}\nabla_\parallel\left(\nabla_\perp\phi + \tau\alpha_d\frac{\nabla_\perp p_i}{n}\right)\} + \mathcal{D}_\omega \end{aligned} \tag{A5}$$

$$\begin{aligned} \frac{d^e v_{\parallel e}}{dt} = &\frac{m_i}{m_e}\epsilon_v\left(\frac{1}{\alpha_d}\nabla_\parallel\phi - \frac{\nabla_\parallel p_e}{n} - 0.71\nabla_\parallel T_e\right) \\ &+ 4\epsilon_v\epsilon_{Ge}\frac{m_i}{m_e}\frac{\nabla_\parallel G_e}{n} + \epsilon_R\alpha_d T_e C_{(v_{\parallel e})} \\ &+ \eta\frac{j_\parallel}{T_e^{3/2}} + S_{\mathcal{M}\parallel e} + \mathcal{D}_{v_{\parallel i}} \end{aligned} \tag{A6}$$

$$\begin{aligned} \frac{d^i v_{\parallel i}}{dt} = &-\frac{\epsilon_v}{\sqrt{\tau}}\left(\frac{1}{\alpha_d}\nabla_\parallel\phi + \tau\frac{\nabla_\parallel p_i}{n} - 0.71\nabla_\parallel T_e\right) \\ &+ \frac{4\epsilon_v\epsilon_G}{\sqrt{\tau}}\frac{\nabla_\parallel G_i}{n} - \epsilon_R\tau\alpha_d T_i C_{(v_{\parallel i})} \\ &- \frac{m_e}{m_i}\frac{\eta}{\sqrt{\tau}}\frac{j_\parallel}{T_e^{3/2}} + S_{\mathcal{M}\parallel i} + \mathcal{D}_{v_{\parallel i}} \end{aligned} \tag{A7}$$

$$\begin{aligned} \frac{d^e \ln T_e}{dt} = &\frac{5}{3}\epsilon_R\alpha_d C_{(T_e)} + \frac{\kappa^e}{p_e}\nabla_\parallel T_e^{7/2}\nabla_\parallel \ln T_e \\ &+\frac{2}{3}\left\{-\epsilon_R\left[C_{(\phi)} - \alpha_d\frac{C_{(p_e)}}{n}\right] - \epsilon_v\nabla_\parallel v_{\parallel e}\right. \\ &+\frac{1}{n}\left[0.71\epsilon_v\left(\nabla_\parallel j_\parallel - j_\parallel\nabla_\parallel\ln T_e\right)\right. \\ &\left.\left.+\frac{m_e}{m_i}\eta\frac{j_\parallel^2}{T_e^{5/2}}\right]\right\} + \frac{2}{3}\frac{1}{p_e}S_{E,e} + \mathcal{D}_{\ln T_e} \end{aligned} \tag{A8}$$

$$\begin{aligned} \frac{d^i \ln T_i}{dt} = &-\frac{5}{3}\tau\epsilon_R\alpha_d C_{(T_i)} + \frac{\kappa^i}{p_i}\nabla_\parallel T_i^{7/2}\nabla_\parallel \ln T_i \\ &+\frac{2}{3}\left\{-\epsilon_R\left[C_{(\phi)} - \alpha_d\frac{C_{(p_e)}}{n}\right] - \sqrt{\tau}\epsilon_v\nabla_\parallel v_{\parallel i}\right. \\ &\left.+\epsilon_v\frac{\nabla_\parallel j_\parallel}{n}\right\} + \frac{2}{3}\frac{1}{p_i}S_{E,i} + \mathcal{D}_{\ln T_i}, \end{aligned} \tag{A9}$$

where the normalized diffusivities applied for all dynamical variables are $\chi_x = -4.54 \times 10^{-10}$, $\chi_y = -1.89 \times 10^{-9}$, and $\chi_z = -8.91 \times 10^{-3}$. The normalized evolution equations given by (A4) and (A8) are the physical model constraints learnt in the machine learning framework as detailed above. A few subtle yet importance differences exist between the physical theory posed and the construction of the synthetic plasma. One deviation between the theorized plasma and the one produced computationally is that the numerical code actually evolves the logarithmic form of n, T_e, and T_i to enforce positivity and the high order diffusion operators act on these logarithmic quantities, too. While equivalent analytically, this choice numerically forces the drift-reduced Braginskii equations to be posed and solved in non-conservative form by the finite difference solver. Consequent errors due to numerical approximation can manifest as unexpected artificial sources or sinks in the simulation domain [28]. In addition, simulation boundaries applied in practice only approximately satisfy the zero flux conditions when employing even- and odd-symmetry conditions on a cell-centered discretized grid [28]. These computational discrepancies can cause potential misalignment between inferred dynamics using idealized theory and numerical modelling of the synthetic plasma's turbulent fields. Physics-informed deep learning can overcome these numerical limitations when representing plasma theory since positivity can be intrinsically encoded in the network plus it employs a continuous spatiotemporal domain and the nonlinear continuum equations represented by (12) and (13) are consequently evaluated exactly up to computer precision [40]. Unphysical numerical dissipation in observational data can therefore present deviations from reflecting the sought plasma theory, but reasonable agreement is nevertheless found when analyzing the synthetic measurements with the partial differential equations embedded in the physics-informed machine learning framework.

ACKNOWLEDGMENTS

We wish to thank P. Perdikaris, S. Wang, and A.Q. Kuang for insights shared and helpful discussions. All simulations presented and codes run are performed using MIT's Engaging cluster and we are grateful for the team's maintenance and assistance with computing resources. The work is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) by the doctoral postgraduate scholarship (PGS D), Manson Benedict Fellowship, and the U.S. Department of Energy (DOE) Office of Science under the Fusion Energy Sciences program by contracts DE-SC0014264, DE-FC02-08ER54966, DE-FG02-04ER54742, and DE-AC52-07NA27344. Relevant data input files and codes can be found on Github (github.com/AbhilashMathews/PlasmaPINNs) and the repository will become public upon publication.

[1] B. A. Carreras, IEEE Transactions on Plasma Science **25**, 1281 (1997).
[2] I. T. Chapman and A. W. Morris, Philosophical transactions. Series A, Mathematical, physical, and engineering sciences **377** (2019).
[3] G. Federici, C. Skinner, J. Brooks, J. Coad, C. Grisolia, A. Haasz, A. Hassanein, V. Philipps, C. Pitcher, J. Roth, W. Wampler, and D. Whyte, Nuclear Fusion **41**, 1967 (2001).
[4] N. M. Li, X. Q. Xu, J. W. Hughes, J. L. Terry, J. Z. Sun, and D. Z. Wang, AIP Advances **10**, 015222 (2020).
[5] C. S. Chang, S. Ku, R. Hager, R. M. Churchill, J. Hughes, F. Köchl, A. Loarte, V. Parail, and R. A. Pitts, Physics of Plasmas **28**, 022501 (2021).
[6] S. I. Braginskii, Reviews of Plasma Physics **1**, 205 (1965).
[7] C.-B. Kim, W. Horton, and S. Hamaguchi, Physics of Fluids B: Plasma Physics **5**, 1516 (1993).
[8] M. Francisquez, *Global Braginskii modeling of magnetically confined boundary plasmas*, Ph.D. thesis, Dartmouth College (2018).
[9] M. Francisquez, B. Zhu, and B. Rogers, Nuclear Fusion **57**, 116049 (2017).
[10] P. Ricci, F. D. Halpern, S. Jolliet, J. Loizu, A. Mosetto, A. Fasoli, I. Furno, and C. Theiler, Plasma Physics and Controlled Fusion **54**, 124047 (2012).
[11] A. S. Thrysøe, M. Løiten, J. Madsen, V. Naulin, A. H. Nielsen, and J. J. Rasmussen, Physics of Plasmas **25**, 032307 (2018).
[12] F. Nespoli, I. Furno, F. D. Halpern, B. Labit, J. Loizu, P. Ricci, and F. Riva, Nuclear Materials and Energy **12**, 1205 (2017).
[13] W. Biel, R. Albanese, R. Ambrosino, M. Ariola, M. V. Berkel, I. Bolshakova, K. J. Brunner, R. Cavazzana, M. Cecconello, S. Conroy, A. Dinklage, I. Duran, R. Dux, T. Eade, S. Entler, G. Ericsson, E. Fable, D. Farina, L. Figini, C. Finotti, T. Franke, L. Giacomelli, L. Giannone, W. Gonzalez, A. Hjalmarsson, M. Hron, F. Janky, A. Kallenbach, J. Kogoj, R. König, O. Kudlacek, R. Luis, A. Malaquias, O. Marchuk, G. Marchiori, M. Mattei, F. Maviglia, G. D. Masi, D. Mazon, H. Meister, K. Meyer, D. Micheletti, S. Nowak, C. Piron, A. Pironti, N. Rispoli, V. Rohde, G. Sergienko, S. E. Shawish, M. Siccinio, A. Silva, F. d. Silva, C. Sozzi, M. Tardocchi, M. Tokar, W. Treutterer, and H. Zohm, Fusion Engineering and Design **146**, 465 (2019).
[14] M. Griener, E. Wolfrum, G. Birkenmeier, M. Faitsch, R. Fischer, G. Fuchert, L. Gil, G. F. Harrer, P. Manz, D. Wendler, and U. Stroth, Nuclear Materials and Energy **25**, 100854 (2020).
[15] C. Bowman, J. R. Harrison, B. Lipschultz, S. Orchard, K. J. Gibson, M. Carr, K. Verhaegh, and O. Myatra, Plasma Physics and Controlled Fusion **62**, 045014 (2020).
[16] A. Mathews and J. Hughes, Quantifying experimental edge plasma evolution via multidimensional adaptive gaussian process regression (2021), arXiv:2103.01305 [physics.plasm-ph].

[17] A. Mathews, J. W. Hughes, J. Terry, M. Goto, D. Stotler, D. Reiter, S.-G. Baek, and S. Zweben, Deep collisional-radiative modelling of edge turbulent electron density and temperature fluctuations (2021), in preparation.

[18] A. N. Simakov and P. J. Catto, Physics of Plasmas **10**, 4744 (2003).

[19] H. Q. Wang, H. Y. Guo, G. S. Xu, A. W. Leonard, X. Q. Wu, M. Groth, A. E. Jaervinen, J. G. Watkins, T. H. Osborne, D. M. Thomas, D. Eldon, P. C. Stangeby, F. Turco, J. C. Xu, L. Wang, Y. F. Wang, and J. B. Liu, Physical Review Letters **124** (2020).

[20] Y. Zhang, Z. B. Guo, and P. H. Diamond, Physical Review Letters **125** (2020).

[21] A. Q. Kuang, S. Ballinger, D. Brunner, J. Canik, A. J. Creely, T. Gray, M. Greenwald, J. W. Hughes, J. Irby, B. LaBombard, and et al., Journal of Plasma Physics **86**, 865860505 (2020).

[22] A. Q. Kuang, N. M. Cao, A. J. Creely, C. A. Dennett, J. Hecla, B. LaBombard, R. A. Tinguely, E. A. Tolman, H. Hoffman, M. Major, J. R. Ruiz, D. Brunner, P. Grover, C. Laughman, B. N. Sorbom, and D. G. Whyte, Fusion Engineering and Design **137**, 221 (2018).

[23] I. Lagaris, A. Likas, and D. Fotiadis, IEEE Transactions on Neural Networks **9**, 987–1000 (1998).

[24] M. Raissi, P. Perdikaris, and G. Karniadakis, Journal of Computational Physics **378**, 686 (2019).

[25] C. Michoski, M. Milosavljević, T. Oliver, and D. R. Hatch, Neurocomputing **399**, 193 (2020).

[26] J. Sirignano and K. Spiliopoulos, Journal of Computational Physics **375**, 1339 (2018).

[27] B. Zhu, M. Francisquez, and B. N. Rogers, Computer Physics Communications **232** (2018).

[28] M. Francisquez, T. N. Bernard, B. Zhu, A. Hakim, B. N. Rogers, and G. W. Hammett, Physics of Plasmas **27**, 082301 (2020).

[29] E. A. Belli and J. Candy, Plasma Physics and Controlled Fusion **50**, 095010 (2008).

[30] J. Madsen, Physics of Plasmas **20**, 072301 (2013).

[31] M. Held, M. Wiesenberger, and A. Kendl, Nuclear Fusion **60**, 066014 (2020).

[32] O. E. Garcia, S. M. Fritzner, R. Kube, I. Cziegler, B. LaBombard, and J. L. Terry, Physics of Plasmas **20**, 055901 (2013).

[33] B. Scott, Physics of Plasmas **10**, 963 (2003).

[34] W. Zholobenko, T. Body, P. Manz, A. Stegmeir, B. Zhu, M. Griener, G. D. Conway, D. Coster, and F. Jenko, Plasma Physics and Controlled Fusion **63**, 034001 (2021).

[35] M. Greenwald, A. Bader, S. Baek, M. Bakhtiari, H. Barnard, W. Beck, W. Bergerson, I. Bespamyatnov, P. Bonoli, D. Brower, D. Brunner, W. Burke, J. Candy, M. Churchill, I. Cziegler, A. Diallo, A. Dominguez, B. Duval, E. Edlund, P. Ennever, D. Ernst, I. Faust, C. Fiore, T. Fredian, O. Garcia, C. Gao, J. Goetz, T. Golfinopoulos, R. Granetz, O. Grulke, Z. Hartwig, S. Horne, N. Howard, A. Hubbard, J. Hughes, I. Hutchinson, J. Irby, V. Izzo, C. Kessel, B. LaBombard, C. Lau, C. Li, Y. Lin, B. Lipschultz, A. Loarte, E. Marmar, A. Mazurenko, G. McCracken, R. McDermott, O. Meneghini, D. Mikkelsen, D. Mossessian, R. Mumgaard, J. Myra, E. Nelson-Melby, R. Ochoukov, G. Olynyk, R. Parker, S. Pitcher, Y. Podpaly, M. Porkolab, M. Reinke, J. Rice, W. Rowan, A. Schmidt, S. Scott, S. Shiraiwa, J. Sierchio, N. Smick, J. A. Snipes, P. Snyder, B. Sorbom, J. Stillerman, C. Sung, Y. Takase, V. Tang, J. Terry, D. Terry, C. Theiler, A. Tronchin-James, N. Tsujii, R. Vieira, J. Walk, G. Wallace, A. White, D. Whyte, J. Wilson, S. Wolfe, G. Wright, J. Wright, S. Wukitch, and S. Zweben, Physics of Plasmas **21**, 110501 (2014).

[36] B. LaBombard, J. W. Hughes, D. Mossessian, M. Greenwald, B. Lipschultz, and J. L. Terry, Nuclear Fusion **45**, 1658 (2005), iOP Publishing.

[37] J. Leddy, B. Dudson, and M. R. and, Plasma Physics and Controlled Fusion **57**, 125016 (2015).

[38] A. S. Thrysøe, V. Naulin, A. H. Nielsen, and J. Juul Rasmussen, Physics of Plasmas **27**, 052302 (2020).

[39] A. Stegmeir, A. Ross, T. Body, M. Francisquez, W. Zholobenko, D. Coster, O. Maj, P. Manz, F. Jenko, B. N. Rogers, and K. S. Kang, Physics of Plasmas **26**, 052517 (2019).

[40] M. Raissi, Journal of Machine Learning Research **19**, 1 (2018).

[41] V. Vassilev, A. Efremov, and O. Shadura, EPJ Web Conf. **245**, 02015 (2020).

[42] S. Wang, H. Wang, and P. Perdikaris, On the eigenvector bias of fourier feature networks: From regression to solving multi-scale pdes with physics-informed neural networks (2020), arXiv:2012.10047 [cs.LG].

[43] G. Cybenko, Mathematics of Control, Signals and Systems **2**, 303 (1989).

[44] S. J. Zweben, J. L. Terry, D. P. Stotler, and R. J. Maqueda, Review of Scientific Instruments **88**, 041101 (2017).

[45] X. Glorot and Y. Bengio, in *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Vol. 9 (2010) pp. 249–256.

[46] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.

[47] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2014), arXiv:1412.6980 [cs.LG].

[48] D. C. Liu and J. Nocedal, Math. Program. **45**, 503–528 (1989).

[49] A. Poulsen, J. J. Rasmussen, M. Wiesenberger, and V. Naulin, Physics of Plasmas **27**, 032305 (2020).

[50] A. Mathews, N. Mandell, M. Francisquez, J. Hughes, and A. Hakim, Turbulent field fluctuations in gyrokinetic and fluid plasmas (2021), arXiv:2107.09744 [physics.plasm-ph].

[51] J. D. Callen, *Fundamentals of Plasma Physics*, draft ed. (University of Wisconsin-Madison, Madison, Wisconsin, 2006) Chap. Collective Plasma Phenomena.

[52] E. Viezzer, T. Pütterich, C. Angioni, A. Bergmann, R. Dux, E. Fable, R. McDermott, U. Stroth, and E. W. and, Nuclear Fusion **54**, 012003 (2013).

[53] C. Wersal and P. Ricci, Nuclear Fusion **55**, 123014 (2015).

[54] A. N. Simakov and P. J. Catto, AIP Conference Proceedings **871**, 238 (2006).

[55] J. D. Huba, *Plasma Physics* (Naval Research Laboratory, Washington, DC, 2013) pp. 1–71.