# Uncovering edge plasma dynamics via deep learning from partial observations

A. Mathews[1],* M. Francisquez[1,2], J.W. Hughes[1], and D.R. Hatch[3]
[1]*MIT Plasma Science and Fusion Center, Cambridge, Massachusetts 02139, USA*
[2]*Princeton Plasma Physics Laboratory, Princeton, New Jersey 08540, USA and*
[3]*Institute for Fusion Studies, University of Texas, Austin, Texas 78704, USA*
(Dated: September 11, 2020)

One of the most intensely studied aspects of magnetic confinement fusion is edge plasma behaviour, which is critical to reactor performance and operation. Drift-reduced Braginskii two-fluid theory has for decades been widely used to model edge plasmas with varying success. This Letter demonstrates that physics-informed neural networks can accurately learn turbulent field dynamics consistent with the two-fluid theory from just partial observations of a synthetic plasma's electron density and temperature for plasma diagnosis and model validation in challenging thermonuclear environments.

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 edge region is critical in determining the fusion device's overall viability since edge plasma microinstabilities strongly influence a myriad of reactor operations ranging from core fuelling to power output to wall safety. Yet edge plasma modelling is in a nascent state today—comprehensive gyrokinetic codes are still under development and fluid simulations commonly lack essential physics to study collisionless environments. One particular transport theory relevant to boundary plasmas and widely applied to analyze edge turbulence is the drift-reduced Braginskii model [3, 4]. Various adaptations of these equations have been recently taken to investigate several important edge phenomena including pedestal physics [5], blob dynamics [6], neutral effects [7], and heat fluxes impinging plasma-facing components [8]. While broad trends are at times reproduced, precise quantitative agreement between two-fluid theories and experiment is generally lacking on a wide scale. It is also increasingly challenging to validate such models as they incorporate ever larger regions of the hot core [9], where measurements are sparse or missing altogether. To this end, we novelly demonstrate a deep learning technique for diagnosing plasma fluctuations and improving experimental validation of turbulence theories to inform discovery of the equations necessary to model the edge.

In this Letter, we represent the drift-reduced Braginskii model via physics-informed neural networks [10–14]—function approximators trained to solve supervised learning tasks respecting nonlinear partial differential equations—to infer unobserved field dynamics from partial observations of a synthetic plasma. This creates a necessary condition for model validation that can be checked by subsequently measuring this inferred quantity in experiment. Moreover, in upcoming reactors with high neutron fluence environments and limited diagnostics, inferring dynamics with minimal experimental information becomes vital [15]. Fusion plasma diagnostic measurements are noisy and limited in their spatiotemporal scope (e.g. 1- or 2-dimensional profiles of electron density and temperature [16, 17]), but it is imperative to maximally utilize information stored in these observations. Therefore, given limited spatial and temporal measurements of a synthetic plasma's electron density and temperature, we accurately infer the radial electric field consistent with the drift-reduced Braginskii equations as both a diagnostic tool and technique to test the model's viability. This framework for inferring turbulent field dynamics presents advances in systematizing fusion theory validation and is to date among the most complex systems applied in physics-informed deep learning codes.

The synthetic plasma analyzed is numerically simulated by the global drift-ballooning (GDB) finite difference code [18, 19] which solves the two-fluid drift-reduced Braginskii equations in the electrostatic limit relevant to low-$\beta$ conditions. The plasma is assumed to be 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 ($n \approx n_e$), vorticity ($\omega$), parallel electron velocity ($v_{\parallel e}$), parallel ion velocity ($v_{\parallel i}$), electron temperature ($T_e$), and ion temperature ($T_i$) [19]:

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

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

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

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

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

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

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 $\eta_0^s$, $\Omega_s$, 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 background 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, $C_{(f)} = \mathbf{b_0}\times\boldsymbol{\kappa}\cdot\nabla f$, and $\mathbf{b_0} = -\hat{\mathbf{z}}$ follow past convention [19]. The coefficients $\kappa_\parallel^s$ and $\eta_\parallel^s$ correspond to parallel heat diffusivity and conductivity, respectively. Lastly, the electrostatic potential, $\phi$, is solved via the following boundary value problem:

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

The 3-dimensional simulation domain of the synthetic plasma consists of a shearless field-aligned coordinate system where $\hat{\mathbf{x}}$ is the unit vector along the radial direction (i.e. $\hat{\mathbf{R}}$), the 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 electrons and deuterium ions with real electron-ion mass ratio (i.e. $m_i = 3.34\times10^{-27}$ kg and $m_e = 9.11\times10^{-31}$ kg) and on-axis magnetic field of 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 for which there is notable evidence of fluid drift turbulence controlling edge dynamics [20]. This discretized 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 drift-wave turbulence. Partial measurements of $n_e$ and $T_e$ over time only come from a smaller 2-dimensional domain in the interior of the synthetic plasma to emulate experiment (e.g. gas puff imaging [21]) with dimensions of $[L_x = 4.0\text{ cm}, L_y = 3.0\text{ cm}]$ and spatial resolution of $[\Delta x = 0.03\text{ cm}, \Delta y = 0.04\text{ cm}]$ as displayed in Figure 1.

In the GDB code, 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 $\omega$ and $\phi$. The lower limit of the Bohm criterion is imposed as a sheath boundary condition on $v_{\parallel e}$ and $v_{\parallel i}$, and finite conductive heat-fluxes enter the sheaths as outlined in [19]. Explicit hyperdiffusion ($\mathcal{D}$) consisting of fourth-order cross-field and second-order parallel diffusion is applied for numerical stability. Constant Gaussian density ($S_n$) and energy sources ($S_{E,s}$) are also placed at the left wall of the simulation domain while external momentum sources ($S_{\mathcal{M}\parallel s}$) are set to zero. 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. A second-order accurate 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. A complete treatment of the numerical solver along with greater specificity regarding the GDB code can be found in [18, 19].

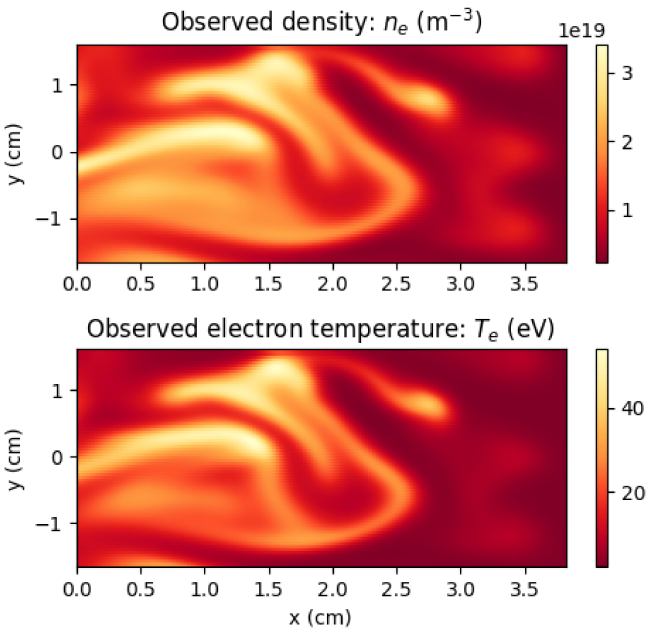


FIG. 1. These 2-dimensional synthetic measurements of $n_e$ and $T_e$ over a finite temporal window are the only variables observed by the physics-informed neural networks from the 3-dimensional simulated plasma exhibiting blob-like filaments.

Every dynamical variable in equations (1)–(6) is represented by its own fully-connected neural network, which is a data-efficient universal function approximator [22]. Each network's architecture consists of 5 hidden layers

with 50 neurons per hidden layer all utilizing Xavier initialization [23]. Physical constraints are learned by the networks by minimizing ascribed loss functions encompassing both the observations of the plasma and model equations. In this Letter, partial observations of the synthetic plasma consist of only $n_e$ and $T_e$ measurements of limited spatial and temporal extent as visualized in Figure 1, but the methods can be trivially generalized to include arbitrary diagnostic measurements of any dimensionality and resolution across open and closed magnetic flux surfaces. All other dynamical variables in the 6-field theory are taken to be unknown, and the framework captures unmeasured quantities consistent with the drift-reduced Braginskii equations. Measured synthetic data are straightforwardly learned by training the electron density and temperature networks against the average $\mathcal{L}_2$-norm of their respective relative errors

$$\mathcal{L}_{n_e} = \frac{1}{N_0}\sum_{i=1}^{N_0}|n_e^*(x_0^i, y_0^i, z_0^i, t_0^i) - n_{e,0}^i|^2, \tag{8}$$

$$\mathcal{L}_{T_e} = \frac{1}{N_0}\sum_{i=1}^{N_0}|T_e^*(x_0^i, y_0^i, z_0^i, t_0^i) - T_{e,0}^i|^2, \tag{9}$$

where $\{x_0^i, y_0^i, z_0^i, t_0^i, n_{e,0}, T_{e,0}\}_{i=1}^{N_0}$ correspond to the set of observed data points given at the simulation resolution 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 [24]. Correspondingly, model loss functions are embedded during training by recasting the evolution equations of (1) and (5) in the following implicit form

$$\begin{aligned} f_{n_e} \coloneqq &-\frac{d^e n}{dt} - \frac{2c}{B}\left[nC_{(\phi)} - \frac{1}{e}C_{(p_e)}\right] \\ &- n\nabla_\parallel v_{\parallel e} + S_n + \mathcal{D}_n, \end{aligned} \tag{10}$$

$$\begin{aligned} f_{T_e} \coloneqq &-\frac{d^e T_e}{dt} + \frac{2T_e}{3n}\left[\frac{d^e n}{dt} + \frac{1}{T_e}\nabla_\parallel \kappa_\parallel^e \nabla_\parallel T_e\right. \\ &+\frac{5n}{m_e\Omega_e}C_{(T_e)} + \eta_\parallel \frac{j_\parallel^2}{T_e} + \frac{0.71}{e}(\nabla_\parallel j_\parallel \\ &\left.-\frac{j_\parallel}{T_e}\nabla_\parallel T_e) + \frac{1}{T_e}S_{E,e}\right] + \mathcal{D}_{T_e}, \end{aligned} \tag{11}$$

and then further normalized into dimensionless form [19]. 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

$$\mathcal{L}_{f_{n_e}} = \frac{1}{N_f}\sum_{i=1}^{N_f}|f_{n_e}^*(x_f^i, y_f^i, z_f^i, t_f^i)|^2, \tag{12}$$

$$\mathcal{L}_{f_{T_e}} = \frac{1}{N_f}\sum_{i=1}^{N_f}|f_{T_e}^*(x_f^i, y_f^i, z_f^i, t_f^i)|^2, \tag{13}$$

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 $f_{T_e}^*$ stand for direct computations by the neural networks of the null partial differential equations prescribed by (10) and (11) in normalized form. The set of collocation points 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 obtained observations. 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) or discretization. This machine learning framework uses a continuous spatiotemporal domain without time-stepping nor finite difference schema in contrast with the numerical code. To handle 2-dimensional data, we essentially assume slow variation of dynamics in the $z$-coordinate and effectively set all parallel derivatives to zero (i.e. $\frac{\partial}{\partial z} \to 0$). Nevertheless, it is necessary to retain parallel flows and Ohmic heating terms in the model to train accurately despite being initially unknown. If measurements in the $z$-direction are observed or more collocation points utilized during training, this procedure may be relaxed—it is simply a trade-off between computational fidelity and training time. Also, 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. Loss functions are optimized with mini-batch sampling using L-BFGS—a quasi-Newton optimization algorithm [25]—and stochastic gradient descent via Adam [26] for 20 hours over 32 cores on Intel Haswell-EP processors.

Remarkably, we find the physics-informed neural network can accurately learn the plasma's electric potential, as displayed in Figure 2. As $\phi$ is a gauge-invariant quantity exact up to an additive constant, it is accordingly learned up to a scalar offset. This difference physically arises because no direct boundary information was enforced upon the deep network when learning $\phi$, although it could be implemented, whereas 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 3 with an average absolute error of 4.194 kV/m while $\mathcal{L}_{f_{n_e}}$ and $\mathcal{L}_{f_{T_e}}$ are $5.173 \times 10^{-2}$ and $3.281 \times 10^{-1}$, respectively. It is notable that despite there being no direct knowledge of $\omega, v_{\parallel e}, v_{\parallel i}$, nor $T_i$ (i.e. multiple unknowns existing in the model equations), the electric field is nonetheless learned consistently with the physical theory encoded in GDB. To emphasize the experimental rel-

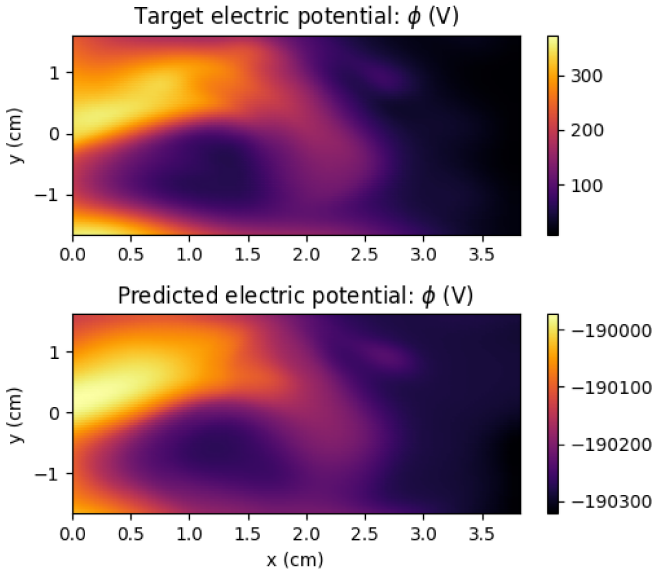


FIG. 2. The synthetic plasma's unobserved electric potential (top) is learned approximately up to an additive constant as predicted by the physics-informed neural network (bottom).

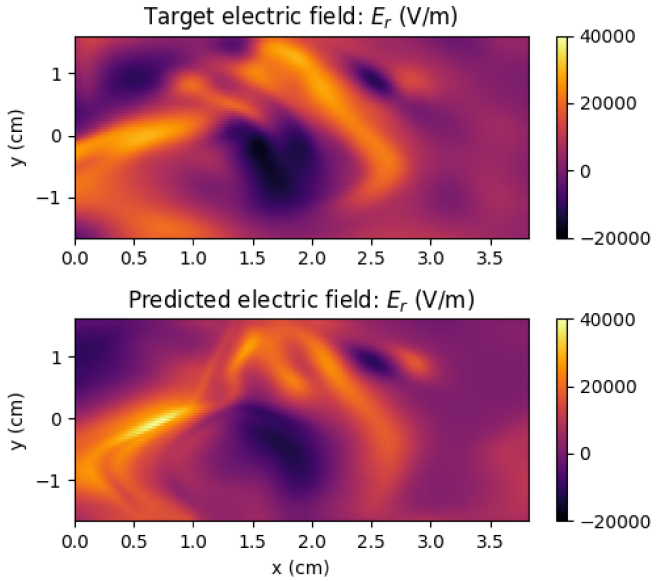


FIG. 3. The learned radial electric field closely agrees with the magnitude and structure of the true radial electric field.

evance of these results and a surprising feature discovered, we note that the radial electric field can actually be mostly learned and reproduced with just partial measurements of $n_e$ without any observations of $T_e$ (which may itself be potentially learned). Therefore, while $T_e$ observations provide stringent constraints on the solution space for quicker convergence, they are not strictly necessary—knowing $T_e$ primarily stabilizes optimization while learning $\phi$. Such analysis of experimental density fluctuation measurements is the focus of continuing work.

All in all, these results illustrate the capability of learning unknown dynamics in a turbulent transport model broadly relevant to magnetized collisional plasmas. This can be used to infer physical quantities that may be difficult to measure or when diagnostic measurements are simply lacking. On the other hand, when inferred field measurements are available, quantitative validation of theory can be expressly addressed. This Letter specifically demonstrates the ability for deep learning to determine unknown turbulent electric field measurements consistent with the drift-reduced Braginskii equations. From a mathematical physics standpoint, it is significant that nonlinear dynamics can be accurately recovered from partial data and theory in a 6-field plasma model. Inferring completely unknown field dynamics from just a 2-dimensional representation of the evolution equations given by (10) and (11) demonstrates a massive reduction in the original multi-field physical model indicating redundancy and the existence of reduced characterizations of turbulence theories for predictive modelling. The computational methods outlined are also transferable across models (e.g. kinetic theory, collisionless fluid closures, electromagnetic, atomic physics) and arbitrarily complex geometries. Furthermore, known limitations and corrections to Braginskii's theory exist [27] which can be introduced in the deep learning framework to automate efficient testing and discovery of reduced turbulence models.

The authors wish to thank P. Perdikaris, S. Wang, N. Mandell, and A.Q. Kuang for insights shared and helpful discussions. All numerical 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. This work is supported by the U.S. Department of Energy (DOE) Office of Science under the Fusion Energy Sciences program by contracts DE-SC0014264, DE-FC02-08ER54966, and DE-FG02-04ER54742, and the Natural Sciences and Engineering Research Council of Canada (NSERC) by the doctoral postgraduate scholarship (PGS D).

* mathewsa@mit.edu

[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] S. I. Braginskii, Reviews of Plasma Physics **1**, 205 (1965).
[4] C.-B. Kim, W. Horton, and S. Hamaguchi, Physics of Fluids B: Plasma Physics **5**, 1516 (1993).
[5] M. Francisquez, B. Zhu, and B. Rogers, Nuclear Fusion **57**, 116049 (2017).
[6] 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).
[7] A. S. Thrysøe, M. Løiten, J. Madsen, V. Naulin, A. H.

Nielsen, and J. J. Rasmussen, Physics of Plasmas **25**, 032307 (2018).
[8] F. Nespoli, I. Furno, F. D. Halpern, B. Labit, J. Loizu, P. Ricci, and F. Riva, Nuclear Materials and Energy **12**, 1205 (2017).
[9] A. Stegmeir, D. Coster, A. Ross, O. Maj, K. Lackner, and E. Poli, Plasma Physics and Controlled Fusion **60**, 035005 (2018).
[10] I. Lagaris, A. Likas, and D. Fotiadis, IEEE Transactions on Neural Networks **9**, 987–1000 (1998).
[11] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations (2017), arXiv:1711.10561 [cs.AI].
[12] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations (2017), arXiv:1711.10566 [cs.AI].
[13] C. Michoski, M. Milosavljević, T. Oliver, and D. R. Hatch, Neurocomputing **399**, 193 (2020).
[14] J. Sirignano and K. Spiliopoulos, Journal of Computational Physics **375**, 1339 (2018).
[15] 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).
[16] A. Mathews and J. W. Hughes, Quantifying experimental edge plasma evolution via multidimensional adaptive gaussian process regression (2020), in preparation.
[17] 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).
[18] B. Zhu, M. Francisquez, and B. N. Rogers, Computer Physics Communications **232** (2018).
[19] M. Francisquez, T. N. Bernard, B. Zhu, A. Hakim, B. N. Rogers, and G. W. Hammett, Physics of Plasmas **27**, 082301 (2020).
[20] B. LaBombard, J. W. Hughes, D. Mossessian, M. Greenwald, B. Lipschultz, J. L. Terry, and t. A. C.-M. Team, Nuclear Fusion **45**, 1658 (2005), IOP Publishing.
[21] S. J. Zweben, J. L. Terry, D. P. Stotler, and R. J. Maqueda, Review of Scientific Instruments **88**, 041101 (2017).
[22] G. Cybenko, Mathematics of Control, Signals and Systems **2**, 303 (1989).
[23] X. Glorot and Y. Bengio, in *JMLR W&CP: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, Vol. 9 (2010) pp. 249–256.
[24] 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.
[25] D. C. Liu and J. Nocedal, Math. Program. **45**, 503–528 (1989).
[26] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2014), arXiv:1412.6980 [cs.LG].
[27] A. N. Simakov and P. J. Catto, AIP Conference Proceedings **871**, 238 (2006).