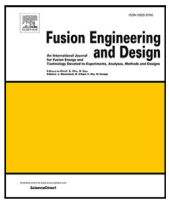
Fusion Engineering and Design 217 (2025) 115010

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes



Shadow masks predictions in SPARC tokamak plasma-facing components using HEAT code and machine learning methods

D. Corona [a,*], M. Scotto d'Abusco [a], M. Churchill [a], S. Munaretto [a], A. Kleiner [a], A. Wingen [c], T. Looby [b]

[a] Princeton Plasma Physics Laboratory, Princeton, NJ 08540, United States of America
[b] Commonwealth Fusion Systems, Devens, MA 01434, United States of America
[c] Oak Ridge National Laboratory, Oak Ridge, TN 37381, United States of America

ARTICLE INFO

Keywords:
Shadow mask
CAD
Neuronal Network
Heat flux
Equilibrium
Surrogate model
Plasma facing component

ABSTRACT

This work uses machine learning (ML) to complement HEAT (Heat flux Engineering Analysis Toolkit) by developing 3-D footprint surrogate models for fast and accurate heat load calculations in the divertor of the SPARC tokamak. The focus is on shadowed regions, or magnetic shadows, caused by the 3-D geometry of plasma-facing components (PFCs). ML classifiers are employed to create a surrogate model for HEAT generated shadow masks, predicting these shadow masks and divertor heat flux profiles based on a diverse range of equilibria and only the plasma current, safety factor(q95) at the edge, and magnetic flux angles as input parameters. The ultimate goal is to integrate the model for real-time control and future operational decisions.

1. Introduction

The handling of power exhaust continues to be a critical challenge for the next generation of fusion devices, which requires innovative solutions in divertor design and operation. Recently significant advances in the application of Machine Learning (ML) for heat flux estimation and real-time control have happened. For instance, the WEST team has employed physics-constrained deep neural networks to extract divertor hot-spot features in real time [1,2], and similar ML approaches have been successfully applied at Wendelstein 7-X to reconstruct heat load patterns on plasma-facing components [3–5]. Moreover, ML-driven real-time control has been demonstrated in experiments like TCV [6] and DIII-D [7], stressing out the potential of these techniques for plasma stabilization and divertor protection.

High-power systems like SPARC [8] demand precise and efficient numerical tools to simulate heat fluxes on complex plasma-facing component (PFC) geometries, both for design and future operations. Codes to simulate divertor heat loads can be compute intensive (e.g. the XGC code [9] runs for hours on massively parallel HPC systems). The HEAT code [10] was developed to provide rapid calculations of divertor heat loads for engineering design considerations. While much faster than other alternatives, a typical HEAT simulation requires on the order of 10 min for a single simulation, with the field-line tracing and ray-triangle intersection checking to determine shadowed regions ("shadow mask") of the divertor being the bottleneck.

This work presents the development, results, and applications of a machine learning-based surrogate model for the HEAT code, specifically tailored for SPARC. Through collaboration among research groups and the use of high-performance computing (HPC), a robust database of magnetic equilibrium and corresponding HEAT simulations was constructed to train the machine learning model. This surrogate significantly accelerates HEAT computations, from tens of minutes to sub-seconds, enabling the possibility of real-time or between-discharge applications for divertor protection and control actions.

2. SPARC tokamak and its divertor

The SPARC tokamak is a device under construction and will be operational with its first plasma scheduled for 2026. It is designed as a high field (B_0 = 12.2 T), compact (R_0 = 1.85 m), superconducting, D-T tokamak. The SPARC divertor is toroidally continuous and tightly baffled to contain neutral particles in the divertor volume. SPARC is being designed to withstand the divertor heat flux in a full-power discharge (10 s flat top) with a single null plasma (although double-null operation is also planned) via impurity seeding and strike point sweeping [8].

The divertor of any fusion reactor must be designed to exhaust the heating power with acceptable loads on the plasma facing components

* Corresponding author.
E-mail address: lcoronar@pppl.gov (D. Corona).

https://doi.org/10.1016/j.fusengdes.2025.115010
Received 21 December 2024; Received in revised form 11 March 2025; Accepted 26 March 2025
Available online 2 May 2025

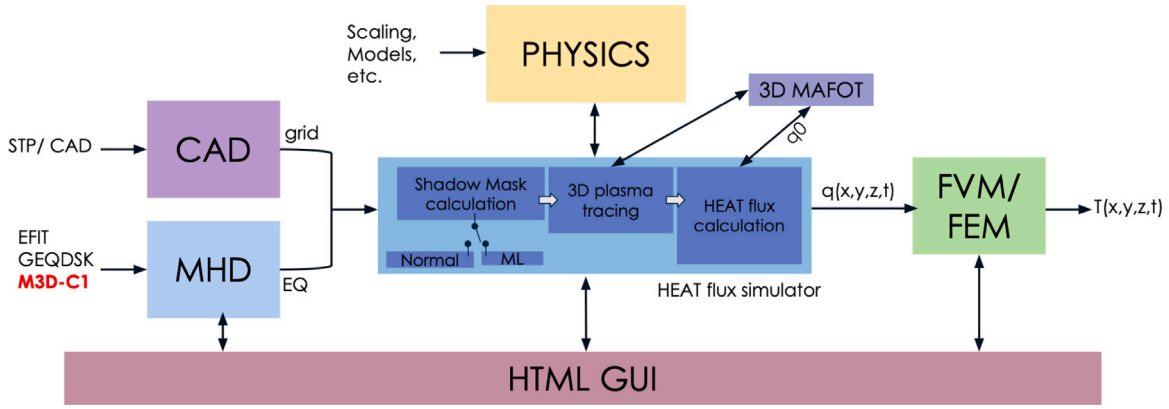


Fig. 1. HEAT code schematic illustrating the Shadow Mask calculation in the Heat Flux Simulator block, now featuring options to execute in either Normal or ML mode.

(PFC's) [11]. Studies in the last decades have shown that the thermal power loads relevant to a fusion reactor are about an order of magnitude more than the power handling capabilities of the standard divertor as the transition from closed to open field lines concentrates the heat flux [12]. From the point of view of power exhaust SPARC is characterized by the expected divertor parallel heat flux peaks that could be as big as $10\,\mathrm{GW/m^2}$ [13–15].

3. The HEAT code and database creation

The HEAT code is a software package designed for the analysis of PFCs by calculating surface heat fluxes with high precision in 3D geometry [10]. It integrates magnetohydrodynamic (MHD) equilibrium and divertor physics with engineering computer-aided design (CAD) models to solve for surface heat fluxes and temperatures across the PFCs. A key functionality of HEAT is the identification of magnetic shadows—regions shielded from the incident heat flux due to the intricate 3D geometry of the PFCs.

The identification of magnetic shadows is essential for accurately modeling heat flux distribution across PFC's. To represent these shielded regions, HEAT generates a shadow mask, a binary classification of surface points that cannot receive incident heat flux due to upstream obstructions in the 3D geometry. This process, known as "intersection checking", traces magnetic field lines from PFC surfaces to determine where they intersect other components, marking these shadowed regions. Techniques such as fish scaling leverage this understanding by designing upstream tiles to strategically shadow downstream edges, as a result mitigating heat flux exposure [16]. A schematic representing the structure of how the HEAT code works is shown in Fig. 1, the submodule for the Shadow Mask calculation using ML will be explained in detail in next sections.

To begin the development of a ML version of the HEAT code, a database of approximately 1000 HEAT simulations was generated, 80% of the database was used for training and 20% as test cases. These runs were based on a set of diverted equilibriums and a divertor CAD model representing a 20-degree section of the SPARC tokamak. The focus of each simulation was the prediction for a specific "carrier", defined as a section of 15 tiles within the region highlighted in Fig. 2 also known as the region "Tile-4" of the divertor. The power input was specified at 20 MW, which leads to an attached divertor peak perpendicular heat flux of $\sim 50\,\mathrm{MW/m^2}$ (actual divertor operation will be detached, with impurity seeding to radiate power to reach material limits of $\sim 10\,\mathrm{MW/m^2}$). The HEAT code was set up to run with a parallel heat flux width $\lambda_q = 1.0$ mm and heat flux spreading parameter $S = 1.0$ mm. It is important to highlight that the shadow mask, being a product of the CAD geometry and the magnetic field configuration, is inherently independent of the heat flux spreading parameter (S) and the parallel heat flux width (λ_q). These parameters primarily influence the quantitative magnitude of the heat flux distribution rather than the qualitative determination of shadowed regions. Although variations in S and λ_q can affect the overall heat flux error metrics, the focus of this work is to isolate the effects of the geometric and magnetic field configurations on the shadow mask.

A single HEAT simulation for this geometry took nominally 47 min with a trace length equal to 10 degrees and a trace step size of 0.05 degrees. The database creation process was parallelized using asynchronous MPI on 8 compute nodes with 90 CPUs per node, reducing the total runtime for the 1000 HEAT simulations to a few hours. The carrier's mesh resolution in these simulations was set at 1 mm, resulting in a Shadow-Mask output of approximately 500,000 geometric points for the whole carrier. Note that this same mesh was used for all of the simulations, such that the ML model had a fixed mesh to predict for. To filter the Shadow-Mask data, a convex hull algorithm was applied to identify and define the key changing points on the Shadow-Mask output of the carrier [17], giving a total of approximately 70,000 changing points. Fig. 2 provides an overview of the divertor structure, the CAD model used, and the simulation results, including the Shadow-Mask output for the selected carrier.

4. Machine learning shadow mask predictions

This work makes use of machine learning (ML) to develop interpretable and generalizable reduced-order models. Feedforward neural networks, also known as multilayer perceptrons (MLPs), are fundamental models in deep learning, designed to approximate a target function f^* [18,19]. In the context of this work, the neural network defines a mapping $y = f(x;\theta)$, where x represents the inputs (such as plasma parameters or PFC's configurations), y represents the predicted results (e.g., shadow mask distributions or heat flux profiles), and θ (comprising the weights and biases of the neurons) are the parameters optimized during training. By learning the optimal θ, the network serves as a surrogate model, providing accurate and efficient approximations of computationally expensive simulations. This capability makes feedforward networks particularly valuable for accelerating iterative workflows in plasma physics, such as real-time or inter-shot predictions for divertor protection and control.

For this model, a deep feedforward neural network (NN) was developed to generate a surrogate model for the HEAT code, referred to as HEAT-ML, for predicting the Shadow Mask in a divertor carrier. The inputs selected to train the NN were four equilibrium parameters: the plasma current (I_p), edge safety factor q_{95}, and two incident field angles, $\frac{B_\theta}{B_\phi}$ (α_1 and α_2), at the top and bottom of the carrier. The parameters used for training the neural network were chosen from the following ranges: plasma current (I_p): $I_p \in [-3.11, -12]$ MA, edge safety factor (q_{95}): $q_{95} \in [2.4, 6.8]$, incident angle 1 (α_1): $\alpha_1 \in [0.496, 4.68]$ degrees and incident angle 2 (α_2): $\alpha_2 \in [1.31, 8.07]$ degrees. Fig. 3

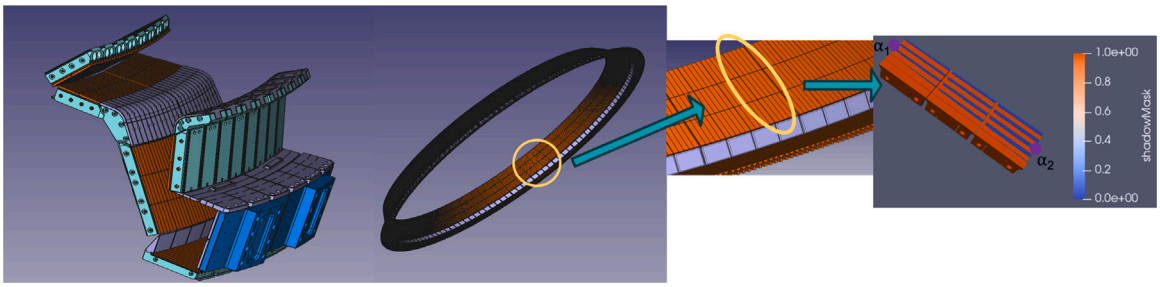
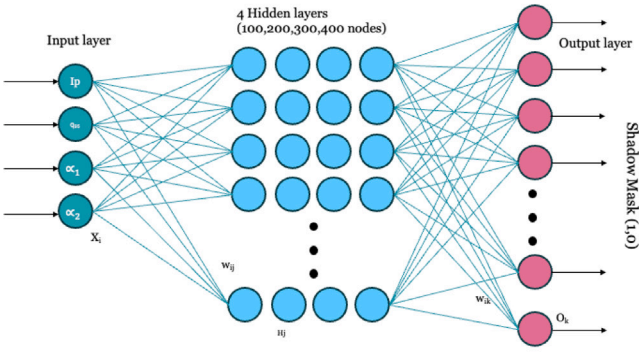


Fig. 2. A visual sequence illustrating the divertor structure and simulation results. The first image shows a 20-degree section CAD model of the divertor. The second image provides a 360-degree overview of the entire divertor, displaying the region known as "Tile-4". The third image zooms in on a specific region of interest, focusing on 15 tiles of the divertor in the "Tile-4" region. The fourth image presents the shadowmask output from the HEAT code, showing the carrier behavior for the selected tiles, orange color corresponds to a value of 1 or shadowed and blue color to a value of 0 or non-shadowed.

Parameter	Value
Number of Input Parameters	4
Number of Outputs	70,000 points (0,1)
Number of Hidden Layers	4
Neurons per Layer	100, 200, 300, 400
Loss Metric	BCELoss
Optimization Algorithm	Adam
Training Time	33 seconds (GPU)

(a)



(b)

Fig. 3. (a) Training parameters for the MLP model, and (b) neural network architecture.

shows the architecture of the NN alongside a table summarizing the key parameters.

A rapid hyperparameter search was done exploring a small range of network architectures. It was found that the configuration with 4 layers, consisting of a conic or funnel-like shape of 100, 200, 300 and 400 nodes respectively, provided a robust balance between model performance and computational efficiency. This selection was determined within a supervised learning framework.

The model was implemented using PyTorch, a framework well-suited for deep learning due to its two key advantages: accelerated computation on graphical processing units (GPUs) and robust support for numerical optimization of mathematical expressions. The use of GPUs allowed for parallelized training, reducing the training time from 25 min to just 30 s, a 50x speedup compared to CPU-based computation. This acceleration was crucial for efficiently handling the computational demands of deep learning [20].

In addition to developing the model, ensuring its practical usability required careful deployment. This process, as depicted on the right in Fig. 4, involved integrating the model where it is needed, whether on a plasma control server, in a cloud engine, etc. Such deployment is essential to bridge the gap between model development and real-world application, enabling the system to perform effectively in its intended environment.

The metric used to assess the performance of the NN model was the coefficient of determination ($R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$, where y_i the true value, $\hat{y}_i$ the NN prediction, and $\bar{y}$ the mean value of the y_i's). For this work, $R^2 > 0.75$ was achieved in 95.7% of the test cases (178 cases), while $R^2 < 0.75$ occurred in 4.3% of the test cases (8 cases), the general value of the coefficient for the database was of $R^2 = 0.955$.

Table 1
Performance metrics and database split for the MLP model.

Parameter	Value
Total simulations (HEAT runs)	930
Training cases	744
Test (Validation) cases	186
General R^2 score	0.955
Test cases with $R^2 > 0.75$	178 (95.7%)
Test cases with $R^2 < 0.75$	8 (4.3%)

Table 1 summarizes the overall performance metrics and the database split, detailing the total number of simulations, the division between training and test cases, and the distribution of R^2 scores for the test cases.

Figs. 5, 6(b), and 6(a) present examples of the shadow mask predictions using HEAT-ML. The figure numbering corresponds to the equilibrium numbers used to create the database. Fig. 5 illustrates one of the best predictions achieved by HEAT-ML. It also includes a close-up of the region where mismatches in the Shadow-Mask pattern are observed. The blue arrows highlight the small red regions where the model did not predict correctly. Fig. 6 depicts the cases with the worst performance, with R^2 values of 0.6653 and 0.5295. Despite being the lowest predictions, the results are not overly poor, as the model, even when incorrect, does not allocate heat flux to physically impossible locations, such as the side of a tile. As shown in all the figures, a plot of the corresponding equilibrium is displayed on the right side of each. Based on the parameters of these two cases, which are not at the edges of the range of the parameter sets, it is not obvious why the ML model performs poorly for these shots, and is under further investigation.

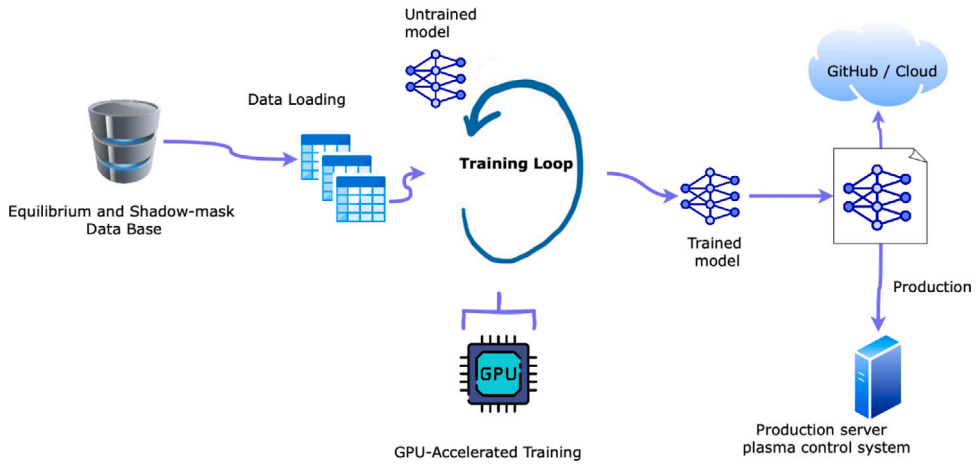


Fig. 4. Basic structure of a PyTorch project with data loading, training and deployment to production.

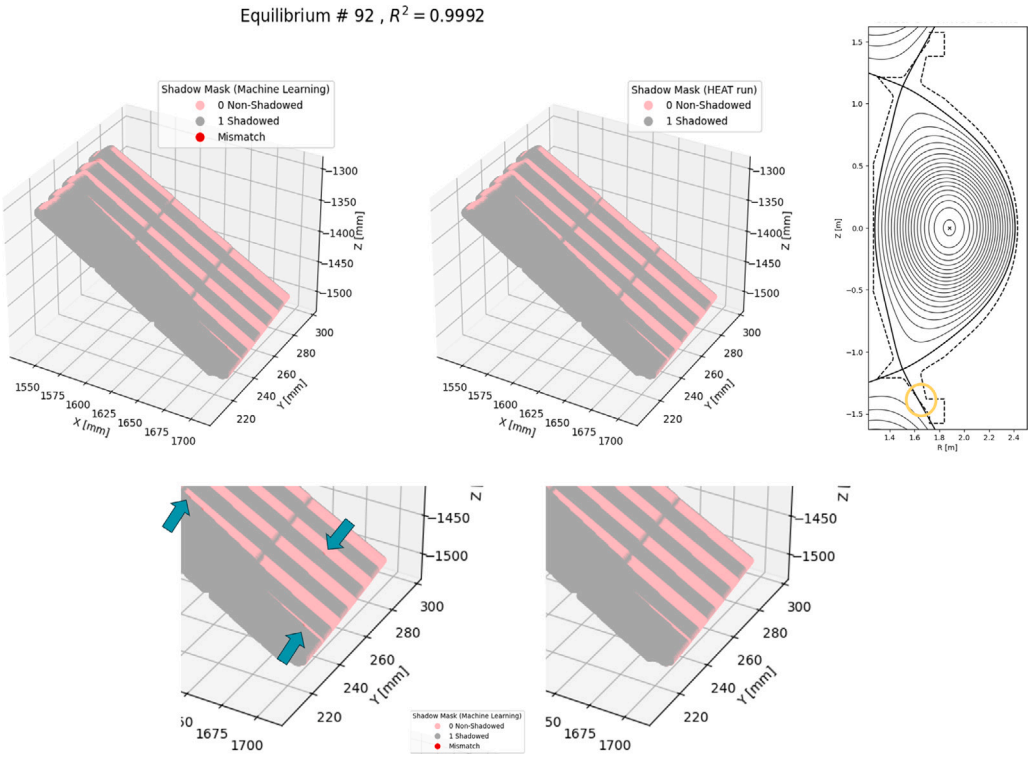
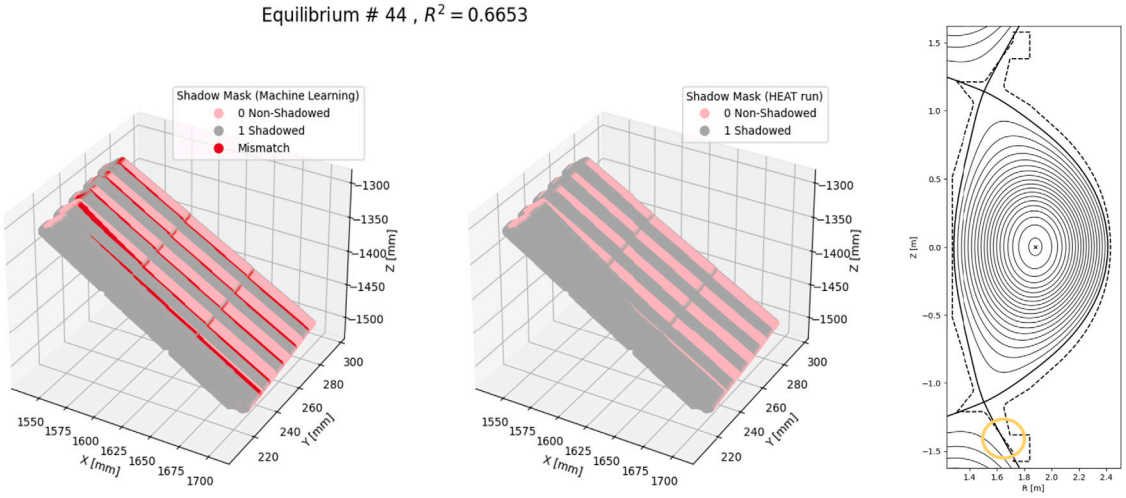


Fig. 5. Combined view of the shadow mask prediction and a zoomed-in detail for Equilibrium #92. The blue arrows in the bottom image indicate the small regions where the prediction algorithm made incorrect predictions.

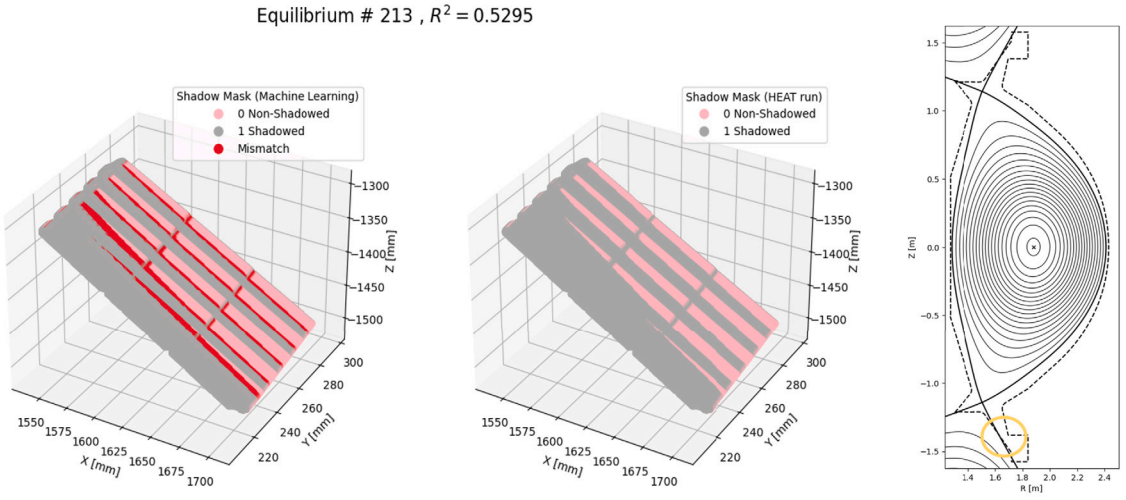
5. Shadow mask calculation implementation and heat flux estimation results

After training the neural network and obtaining its optimal parameters (weights and biases), this trained model was integrated into the HEAT code as an optional module for the Shadow Mask calculation. This integration enables the HEAT code to use the neural network directly during execution for Shadow Mask calculations, then using the existing HEAT code infrastructure for the heat flux computation.

The runtime for the entire HEAT code using the ML version for the Shadow Mask calculation process is of approximately 90 s. This timing is primarily attributed to overhead and file I/O operations, as the NN predictions themselves take only a few milliseconds. The heat flux prediction results for the analyzed equilibriums are presented in Fig. 7. Specifically, Fig. 7(a) corresponds to $R^2 = 0.9992$, Fig. 7(b) to $R^2 = 0.6653$, and Fig. 7(c) to $R^2 = 0.5295$. A small plot of each equilibrium is displayed in the middle of the corresponding carrier 3D plots. The general RMSE score is equal to 3.352 MW/m^2 calculated as $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_{\text{pred},i} - y_{\text{true},i}\right)^2}$ where $y_{\text{pred},i}$ corresponds to the heat flux value calculated by HEAT using the ML Shadow-mask option and $y_{\text{true},i}$. corresponds to the heat flux value calculated normally. For the specific cases analyzed in this work, they have RMSE values of 0.45, 8.85 and 5.66 MW/m^2 respectively. Given that the peak perpendicular heat flux in a full-power attached divertor is approximately of 50 MW/m^2,



(a) Equilibrium 44



(b) Equilibrium number 213 with a $R^2 = 0.5295$

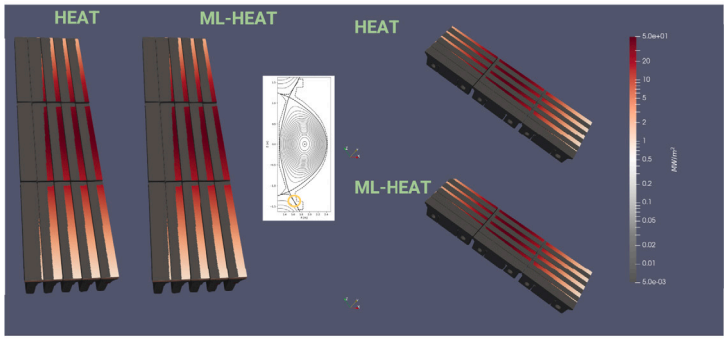
Fig. 6. Shadow mask comparison for Equilibrium #44 and Equilibrium #213. In the images on the right, the red regions indicate where the prediction did not succeed.

the overall RMSE of 3.352 MW/m^2 corresponds to a relative error of roughly 6.7%.

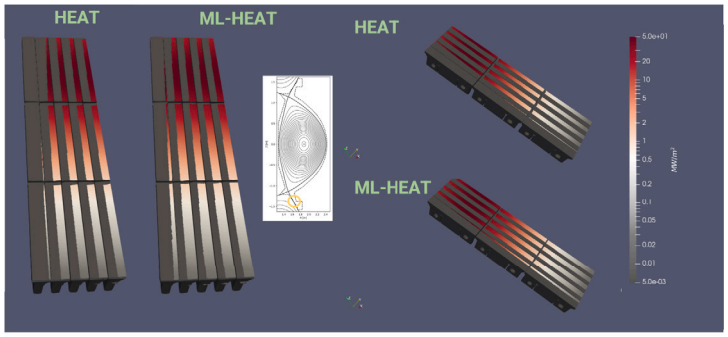
6. Applications and future work

Its important to point out the applications and some of the limitations of the current ML surrogate model. Due to the fixed mesh used, the current ML surrogate model cannot be quickly adapted to new divertor geometries (a new database of HEAT runs would need to be created, which is manageable on the 45 min time frame, but not for making topological optimization). For fixed divertor geometry, using the ML-HEAT model can produce divertor heatload maps in milliseconds (when stripping out the overhead due to file I/O, setup time, etc.). The primary application then of these ML surrogate models for the HEAT code is to enable real-time or between-discharge divertor protection, allowing for closed-loop feedback control to protect the divertor from potential damage or excessive stress and quicker between shot decision-making and scenario planning. The implementation of the ML surrogate model into the SPARC plasma control system is under discussion. Although the current work focuses on accelerating the HEAT code using ML, the approach is inherently modular and can be extended in the future. In particular, potential extensions may include coupling ML-based analysis with infrared (IR) diagnostics for real-time heat flux estimation. Such integration remains as another possible direction for further research and could enhance the overall diagnostic and control capabilities.

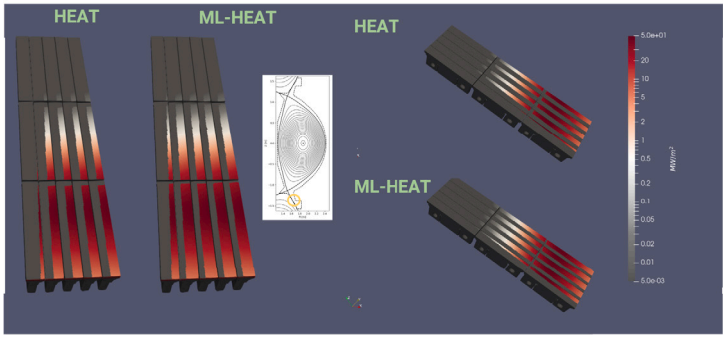
Future work will also focus on improving the generalization of the predictions, making sure they are not limited to specific PFC's. Advanced ML techniques for learning on generic meshes will be explored (e.g. MeshGraphNet [21], Geo-FNO [22], etc.). However, the bottleneck in the 2-D axisymmetric HEAT simulations is not in the field-line tracing solver, but rather in the collision detection algorithm used to determine when field lines intersect CAD geometries. ML techniques for solving the field-line tracing ODE but also to accelerate collision detection can potentially learn a more flexible model, useful for real-time control and between-discharge divertor protection, but also for divertor design.



(a) Comparison of the heat flux prediction using the regular and ML versions of the Shadow Mask calculation in the HEAT code for Equilibrium #92. $R^2 = 0.9992$ and RMSE = 0.45 MW/m^2



(b) Comparison of the heat flux prediction using the regular and ML versions of the Shadow Mask calculation in the HEAT code for Equilibrium #44. $R^2 = 0.6653$ and RMSE = 8.85 MW/m^2



(c) Comparison of the heat flux prediction using the regular and ML versions of the Shadow Mask calculation in the HEAT code for Equilibrium #213. $R^2 = 0.5295$ and RMSE = 5.66 MW/m^2

Fig. 7. Comparison of heat flux predictions with the use of ML in the HEAT code. The code was configured with the parameters of parallel heat flux width λ_q = 1.0 mm, heat flux spreading parameter S = 1.0 mm and the power crossing the separatrix into the SOL P_{SOL} = 20 MW.

CRediT authorship contribution statement

D. Corona: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **M. Scotto d'Abusco:** Supervision, Methodology, Investigation. **M. Churchill:** Validation, Supervision, Software, Methodology, Investigation, Data curation, Conceptualization. **S. Munaretto:** Validation, Supervision, Resources, Project administration, Funding acquisition, Conceptualization. **A. Kleiner:** Methodology, Investigation. **A. Wingen:** Supervision, Software, Methodology, Investigation. **T. Looby:** Validation, Software, Project administration, Investigation, Conceptualization.

Declaration of competing interest

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

Acknowledgments

This work was supported by the U.S. Department of Energy under contract number DE-AC02-09CH11466, DE-AC05-00OR22725 and the support from Commonwealth Fusion Systems. The United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

Data availability

The data that has been used is confidential.

References

[1] V. Gorse, R. Mitteau, J. Marot, the WEST TEAM, Using a physics constrained U-Net for real-time compatible extraction of physical features from WEST divertor hot-spots, J. Fusion Energy 43 (1) (2024) 13, http://dx.doi.org/10.1007/s10894-024-00405-y.

[2] R. Mitteau, et al., WEST operation with real time feedback control based on wall component temperature toward machine protection in a steady state tungsten environment, Fusion Eng. Des. 165 (2021) 112223, http://dx.doi.org/10.1016/j.fusengdes.2020.112223.
[3] M. Blatzheim, D. Böckenhoff, the W7-X Team, Neural network regression approaches to reconstruct properties of magnetic configuration from Wendelstein 7-X modeled heat load patterns, Nucl. Fusion 59 (12) (2019) 126029, http://dx.doi.org/10.1088/1741-4326/ab4123.
[4] F. Pisano, et al., Learning control coil currents from heat-flux images using convolutional neural networks at wendelstein 7-X, Plasma Phys. Control. Fusion (2020) http://dx.doi.org/10.1088/1361-6587/abce19.
[5] E. Aymerich, et al., Physics informed neural networks towards the real-time calculation of heat fluxes at W7-X, Nucl. Mater. Energy 34 (2023) 101401, http://dx.doi.org/10.1016/j.nme.2023.101401.
[6] J. Degrave, et al., Magnetic control of tokamak plasmas through deep reinforcement learning, Nature 602 (7897) (2022) 414–419, http://dx.doi.org/10.1038/s41586-021-04301-9.
[7] J. Seo, et al., Avoiding fusion plasma tearing instability with deep reinforcement learning, Nature 626 (8000) (2024) 746–751, http://dx.doi.org/10.1038/s41586-024-07024-9.
[8] A.J. Creely, the SPARC team, Overview of the SPARC tokamak, J. Plasma Phys. (ISSN: 14697807) 86 (2020) http://dx.doi.org/10.1017/S0022377820001257.
[9] S. Ku, C.S. Chang, P.H. Diamond, Full-f gyrokinetic particle simulation of centrally heated global ITG turbulence from magnetic axis to edge pedestal top in a realistic tokamak geometry, Nucl. Fusion (ISSN: 00295515) 49 (2009) http://dx.doi.org/10.1088/0029-5515/49/11/115021.
[10] T. Looby, et al., A software package for plasma-facing component analysis and design: The heat flux engineering analysis toolkit (HEAT), Fusion Sci. Technol. (ISSN: 19437641) 78 (2022) http://dx.doi.org/10.1080/15361055.2021.1951532.
[11] R. Stambaugh, the ITER Physics Expert Group on Divertor, Chapter 4: Power and particle control, Nucl. Fusion 39 (1999).
[12] M. Kotschenreuther, P.M. Valanju, S.M. Mahajan, J.C. Wiley, On heat loading, novel divertors, and fusion reactors, Phys. Plasmas (ISSN: 1070664X) 14 (2007) http://dx.doi.org/10.1063/1.2739422.
[13] A.Q. Kuang, et al., Divertor heat flux challenge and mitigation in SPARC, J. Plasma Phys. (ISSN: 14697807) (2020) http://dx.doi.org/10.1017/S0022377820001117.
[14] D. Iglesias, et al., An improved model for the accurate calculation of parallel heat fluxes at the JET bulk tungsten outer divertor, Nucl. Fusion 58 (10) (2018) 106034, http://dx.doi.org/10.1088/1741-4326/aad83e.
[15] B. Labombard, et al., ADX: a high field, high power density, advanced divertor and RF tokamak, Nucl. Fusion 55 (5) (2015) 053020, http://dx.doi.org/10.1088/0029-5515/55/5/053020.
[16] T. Looby, M.L. Reinke, A. Wingen, T. Gray, E.A. Unterberg, D. Donovan, 3D ion gyro-orbit heat load predictions for NSTX-U, Nucl. Fusion (ISSN: 17414326) 62 (2022) http://dx.doi.org/10.1088/1741-4326/ac8a05.
[17] M.D. Berg, O. Cheong, M.V. Kreveld, M. Overmars, Computational geometry: Algorithms and applications, Springer, 2008, http://dx.doi.org/10.1007/978-3-540-77974-2,
[18] A. Courville, I. Goodfellow, Y. Bengio, Deep Learning, MIT Press, 2016.
[19] F. Chollet, Deep Learning with Python, Manning Shelter Island, 2018.
[20] L.A. Eli Stevens, T. Viehmann, Deep Leanring with PyTorch, Manning Shelter Island, 2021.
[21] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, P.W. Battaglia, Learning mesh-based simulation with graph networks, 2020, URL http://arxiv.org/abs/2010.03409.
[22] Z. Li, D.Z. Huang, B. Liu, A. Anandkumar, Fourier neural operator with learned deformations for PDEs on general geometries, 2022, http://dx.doi.org/10.48550/arXiv.2207.05209, URL http://arxiv.org/abs/2207.05209. arXiv:2207.05209 [cs, math].