Automated ICRF heating surrogate modeling via machine learning

Alvaro Sanchez-Villar[1,*], *Nicola* Bertelli[1], *Syun'ichi* Shiraiwa[1], *Zhe* Bai[2], *Talita* Perciano[2], *E. Wes* Bethel[3], *Julien* Hillairet[4], *Joseph* Miller[5], *Gregory M.* Wallace[6], and *John C.* Wright[6]

[1]Princeton Plasma Physics Laboratory, Princeton, NJ 08540, USA
[2]Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA
[3]San Francisco State University, San Francisco, CA 94132, USA
[4]CEA-IRFM, F-13108 Saint-Paul-lez-Durance, France
[5]University of California, Los Angeles, CA 90095, USA
[6]MIT Plasma Science and Fusion Center, Cambridge, MA 02139, USA

Abstract. This work introduces automated machine learning workflows that address critical bottlenecks in surrogate model development for Ion Cyclotron Range of Frequencies (ICRF) heating applications. The automated framework includes data analysis tools that transform raw datasets into actionable insights in seconds, replacing weeks of manual exploratory effort and ensuring consistent, reproducible dataset characterization. By integrating advanced hyperparameter optimization (HPO) methods including Bayesian optimization via BoTorch and Tree-structured Parzen Estimators (TPE), the framework significantly reduces model development time from weeks to hours, decreasing computational cost and required expertise, while enabling high-accuracy surrogate models. Compared to traditional hyperparameter scanning (HPS) techniques such as methodical, randomized, and grid searches, HPO methods achieve superior convergence and predictive performance, even when compared to already well-tuned reference models. On NSTX High Harmonic Fast Wave (HHFW) heating datasets, both Random Forest Regressor (RFR) and neural network surrogates demonstrate improved accuracy, achieving R^2 values beyond 0.97 and 0.98, respectively. The results show that while HPO gains are modest for robust architectures like RFR, they become essential for more sensitive models such as neural networks, highlighting the trade-offs across optimization strategies. Through automated workflows that eliminate manual hyperparameter tuning and require minimal ML expertise, this work enables widespread adoption of high-fidelity surrogate models across the fusion community for real-time plasma control, uncertainty quantification, rapid experimental scenario development, and integrated system optimization.

1 Introduction

Ion Cyclotron Range of Frequencies (ICRF) heating [1] represents a critical component in many modern magnetic confinement fusion devices, enabling efficient plasma heating and current drive capabilities essential for sustained fusion reactions. The optimization and control of ICRF systems require accurate, fast predictions of heating profiles and power deposition patterns across various plasma conditions and device configurations. Traditional physics-based radiofrequency (RF) computational models such as TORIC [2], while highly accurate, often require substantial computational resources and execution times that render them impractical for real-time control applications, scenario optimization, and inter-shot predictive modeling.

Surrogate modeling emerges as a compelling solution to bridge the gap between computational accuracy and operational efficiency. By leveraging machine learning techniques to approximate complex physics-based models, surrogate models can provide near-instantaneous predictions while maintaining acceptable accuracy levels. This approach enables previously computationally prohibitive applications such as real-time plasma control, optimization of heating scenarios, in-the-loop code acceleration, rapid parameter space exploration for experimental planning, and synthetic data generation.

The development of robust surrogate models requires careful consideration of model architecture selection, hyperparameter tuning, training methodologies, and uncertainty quantification. Our recent efforts focused on the evaluation of Random Forest Regressor (RFR), Multilayer Perceptron (MLP), and Gaussian Process Regressor (GPR) capabilities to serve as baseline algorithms to achieve real-time capable predictions [3] and extrapolative properties [4]. However, previous methodologies [3–6] rely on suboptimal hyperparameter scanning approaches (methodical, randomized, and gridded) that are time-consuming and difficult to reproduce.

The primary challenge is not only achieving adequate surrogate model performance, but also doing so efficiently, reproducibly, and accessibly, without requiring extensive machine learning expertise. Manual hyperparameter tuning often demands weeks or months of expert effort per application or dataset, severely limiting the adoption of surrogate modeling techniques across a broader fusion research community. In addition, manual data analysis, necessary to

*Corresponding author: asvillar@pppl.gov

evaluate data distributions, correlations, and data quality, is time-intensive and may overlook critical dataset characteristics.

This work addresses these critical bottlenecks in the surrogate modeling implementation process by introducing automated workflows for data ingestion, analysis, and hyperparameter optimization. The key research questions are: (1) Can automated data ingestion and analysis pipelines replace manual exploratory efforts while ensuring reproducibility and reliable dataset characterization? (2) Can automated hyperparameter optimization reduce both the time and expertise required for developing production-ready surrogate models, while outperforming traditional manual scanning approaches? (3) What are the trade-offs between different optimization strategies across various model architectures in fusion surrogate modeling? The key contributions include: (1) automated data ingestion, analysis, and visualization of datasets, (2) automated hyperparameter optimization targeting the most significant bottleneck in surrogate development, and (3) comprehensive evaluation of advanced optimization strategies, including Bayesian optimization methods.

This paper is structured as follows. Section 2 describes the automated data ingestion, analysis, and visualization workflow and its application to ICRF heating datasets. Section 3 presents automated hyperparameter optimization strategies, comparing traditional approaches with advanced methods and evaluating their impact on model performance. Section 4 provides concluding remarks and outlines future work directions.

2 Automated Data Ingestion and Analysis

The foundation of effective surrogate modeling lies in the quality and representativeness of the training data. For ICRF heating applications, the data encompasses a multi-dimensional parameter space characterizing plasma conditions, device configuration, and heating system parameters. Key input parameters include plasma density and temperature profiles, magnetic field configurations, antenna specifications, frequency settings, and geometric factors specific to the tokamak device. The target variables typically consist of heating power deposition profiles for both electrons and ions.

2.1 Data Collection and Preprocessing

The data collection process starts with identifying the parameter space of interest for our surrogate modeling purposes from typical experimental scenarios. Then it is required to choose a sampling strategy. In order to sample systematically, and in an automated and optimized manner, here we use the Latin Hypercube Sampling (LHS) method. LHS offers several advantages: (1) ensuring uniform coverage across all parameter dimensions, (2) minimizing correlation between sampling points, (3) providing optimal parameter space-filling properties for surrogate training, and (4) avoiding the "curse of dimensionality". The implemented workflows sample and transfer the required inputs, and execute the reference code (in this case, TORIC [2]) in batched simulations via `SLURM`. Once the simulations have finished, the workflows retrieve the dataset in flexible formats (`.csv`, `.pkl`, `.npy`). Depending on the target application and requirements, the appropriate format is chosen. By default, we operate with pickle files due to their binary characteristic and their associated rapid loading speed. We also use the pandas `DataFrame` class as the main element for moderately sized datasets to store the data for each simulation batch (i.e. all variables for the $O(10^4)$ simulations). For larger batches we recommend using binary `.npy` arrays or `HDF5` formats.

The workflow supports automated data ingestion, loading, and content summarization, properly identifying the number of input and output variables in the dataset. Automated data preprocessing encompasses standardization and normalization techniques, applied according to the characteristics of each input variable. By default, the workflow performs standardization where the variables are subtracted from their mean and divided by their standard deviation. This ensures that all input features contribute equally during training, enhancing the stability, efficiency, and convergence of surrogate models, particularly those that rely on gradient-based optimization, such as neural networks.

2.2 Data Analysis and Visualization

The automated analysis pipeline generates comprehensive visualizations and statistical summaries to provide insights into dataset characteristics and quality. Figure 1 shows the key components of this visualization analysis for the particular case of the electron HHFW power absorption dataset for NSTX:

• Violin Plots for Input and Output Distributions: the automated generation of violin plots reveals the statistical distribution characteristics of both input parameters (see Figure 1(a)) and heating profiles (see Figure 1(b)). The effectiveness of using the LHS method can be observed in the uniform distributions of the input variables: the core electron density (n_{e0}), the toroidal mode number (N_φ), the density inner profile exponent (α_{n_e}), and the core electron temperature (T_{e0}). In contrast, the distributions in the output electron power absorption profile (P_e) variables are more concentrated around particular values. Note that P_e indices 0-438 represent core and edge locations, respectively, corresponding to $\rho = 0$ and 1, where $\rho \in [0, 1]$ is the square root of the normalized poloidal flux coordinate. Figure 1(b) shows that most of the values of P_e are negligible close to the plasma core ($P_{e,0}$) while the peak value of the entire dataset occurs also at the plasma core, denoting that while in most scenarios the electron power deposition occurs at larger ρ, there are a few scenarios in which the value can be significant at the core. However, note that the volume in low ρ regions is significantly smaller. This P_e behavior changes approaching the edge where the expected heating rises, reaching its maximum around $P_{e,350}$. These visualizations provide insights into data variability, concentration patterns, and potential outliers that may impact model training. For heating profiles, violin plots reveal

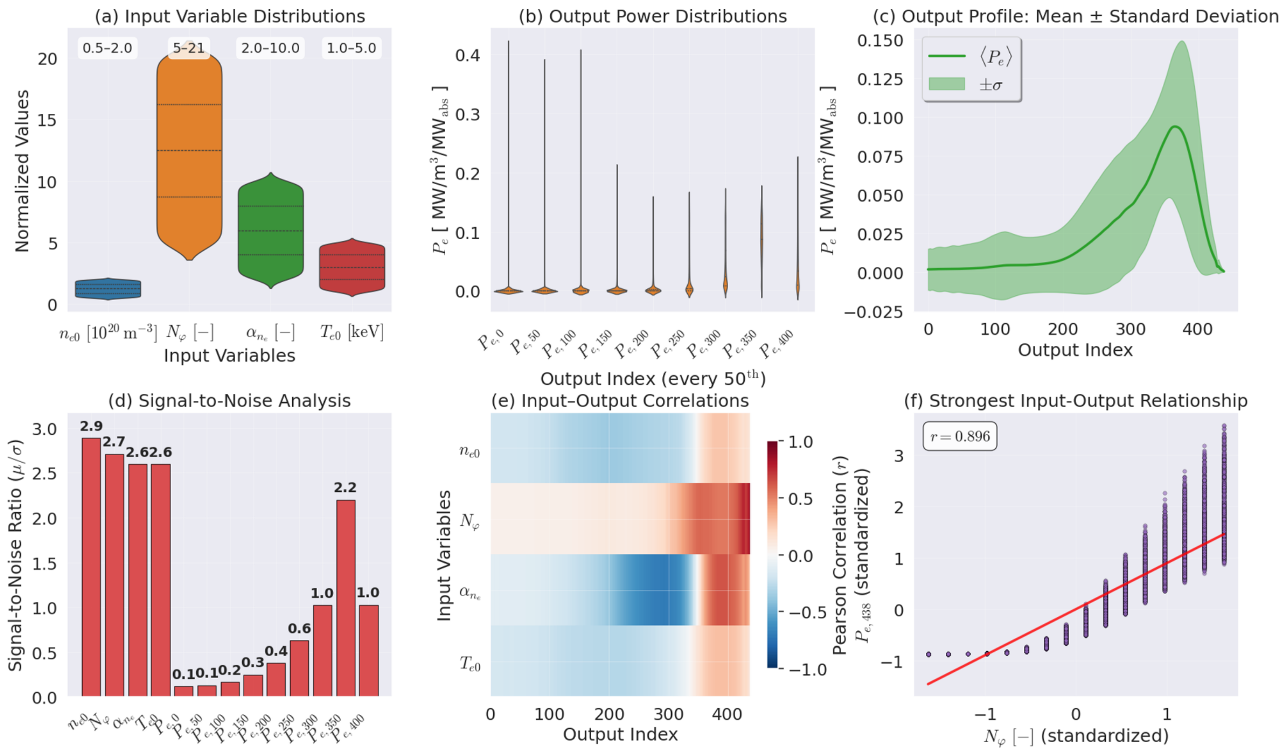

Figure 1. Automated analysis of electron HHFW power absorption dataset for NSTX. (a) Input parameter distributions showing Latin Hypercube Sampling coverage, (b) output power profile distributions across the normalized flux coordinate $\rho \in [0, 1]$ (indexed), (c) mean heating profiles with standard deviation bands, (d) signal-to-noise ratio analysis, (e) Pearson correlation heatmap between inputs and output variables, (f) strongest input-output correlation with maximum correlation $r = 0.896$ between the toroidal mode number and the value of the electron heating power at the edge.

whether power deposition patterns exhibit narrow, peaked distributions or broader, more variable patterns, informing the selection of appropriate modeling approaches.

• Profile Analysis: Figure 1(c) illustrates the mean value and standard deviation of the heating profile predictions, showing that, in general, the maximum and most widespread heating is located around $P_{e,350}$, as the violin plots also indicated. This comprehensive analysis of heating profile characteristics includes peak locations and width distributions, identifying the range of profile types present in the dataset and quantifying their variability.

• Signal-to-Noise Analysis: Figure 1(d) shows the ratio between the mean (μ) and standard deviation (σ) values, termed as the signal-to-noise ratio. This calculation assesses data quality across different input parameters. Note that while the heating data appears well-characterized close to the edge, the variability increases towards the core, decreasing the quality of the data in that region. This analysis identifies parameters with high measurement uncertainty or computational noise, enabling informed decisions about data filtering and preprocessing requirements.

• Correlation Analysis: Figure 1(e) shows the input-output correlations using the Pearson correlation coefficient (r) of two variables (X, Y), defined as $r_{X,Y} = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}}$. This automated correlation analysis generates heatmaps showing relationships between input parameters and output profile characteristics, revealing the strongest input-output correlations, and identifying potential parameter redundancies or unexpected dependencies that may impact model performance. For instance, we observe that the toroidal mode number is strongly correlated with the output power. However, when approaching the location of the maximum mean power absorption and towards the core, we observe that all correlations decrease rapidly, indicating that the correlation is non-linear. Additionally, we can also correlate each individual variable and examine its correlation with the inputs. Figure 1(f) shows where strongest input-output correlation occurs, which is for the edge power absorption and the toroidal mode number, with $r = 0.896$.

3 Automated Hyperparameter Optimization

Hyperparameter optimization represents the most significant bottleneck in achieving effective surrogate models. This process is not only time- and resource-consuming but can severely limit the achievement of accurate surrogate model architectures. Traditional methodologies reported in similar applications of surrogate models rely on rudimentary hyperparameter scanning (HPS) techniques, which are inefficient in both computational resources and human labor, and often ineffective at finding optimal surrogate solutions. While HPS methods such as methodical scanning, randomized searches, and gridded sampling provide a reasonable starting point to achieve moderate regression accuracies, more refined hyperparameter optimization (HPO) methodologies are generally required to deploy surrogates at the implementation level. This work builds upon previous efforts in the fusion energy sciences community that focused on the use of randomized, gridded, and methodical scans (i.e. HPS) [3–6]. The automated HPO workflow presented here integrates multiple optimization strategies to accommodate different computational budgets and optimization objectives, with a focus on testing different automated hyperparameter tuning methodologies using available open-source software.

HPS includes three main methods. First, methodical scanning involves systematic evaluation of hyperparameter combinations across predefined ranges. While comprehensive, this approach scales poorly with the number of hyperparameters and requires extensive computational resources. The method provides reproducible results but often fails to identify optimal configurations in high-dimensional parameter spaces. Second, grid search methods evaluate hyperparameters at regularly spaced intervals across specified ranges. This approach ensures coverage of the parameter space but suffers from the curse of dimensionality, requiring exponentially increasing computational resources as the number of hyperparameters grows. Grid search also allocates equal computational budget to all parameter combinations, regardless of their likelihood of yielding optimal results. Third, random search methods sample hyperparameter combinations randomly from specified distributions. This approach often outperforms grid search in high-dimensional spaces and provides better coverage of the parameter space with limited computational budget. However, random search lacks the ability to learn from previous evaluations and may spend significant effort on unpromising parameter regions.

HPO techniques play a major role in the streamlined and effective implementation of surrogate models. These methods require only the computational expense of the optimization algorithm itself, which, if properly employed, can dramatically reduce computational cost and require minimal human intervention. Advanced HPO methods leverage probabilistic models or population-based strategies to efficiently explore the search space. Bayesian optimization (BO) [7] methods, such as Gaussian Process (GP) regression, construct a surrogate model of the objective function $f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), \mathcal{K}(\mathbf{x}, \mathbf{x}'))$, where μ and $\mathcal{K}$ denote the mean function and kernel, respectively. The surrogate is updated iteratively with new evaluations and is used to select future configurations via acquisition functions that balance exploration and exploitation.

An alternative is the Tree-structured Parzen Estimator (TPE) [8], which models conditional densities $p(\mathbf{x} \mid y < y^*)$ and $p(\mathbf{x} \mid y \geq y^*)$, where y^* is a performance threshold. Configurations are sampled from regions of the search space that are likely to yield improvements, thereby implicitly incorporating a prior over promising regions.

3.1 Implementation

The automated HPO workflows developed are based on the `Optuna` [9] framework, enabling systematic comparison of different approaches. The implementation includes:

- Random sampling as a baseline method.
- TPE sampling for efficient exploration of discrete parameter spaces.
- `BoTorch` sampling for advanced BO using Gaussian Process surrogates [10].

The methodology compares the three different approaches (Random, Optuna TPE, and Optuna BoTorch) and returns a study including the best model in terms of the selected target optimization variable selected. The two optimization variables selected are the mean squared error (MSE) and the coefficient of determination (R^2) implemented in `scikit-learn` [11].

For the RFR [11] models, we optimize the following hyperparameters: `n_estimators`, `max_depth`, `min_samples_split`, `min_samples_leaf`, and `max_features`. For the MLP models developed with `PyTorch` [12], which are fully connected networks, we optimize several hyperparameters: `n_layers`, `hidden_size`, `dropout_rate`, `learning_rate`, `batch_size`, and `activation_fn`.

3.2 Results and Analysis

Here we demonstrate the hyperparameter tuning capabilities for the electron power deposition dataset generated for the HHFW for NSTX [4]. The methodology is tested to determine whether it is possible to (1) achieve effective surrogate models via these methods, and (2) improve upon the models achieved in our previous iterations [3, 4]. Thus, the reference models in this case are those from Ref. [4] using the final outlier-free dataset. This demonstration is particularly challenging given that the achieved reference point is the result of an extensive investigation with multiple HPS methods applied across multiple iterations of the models, resulting in a successful reference architecture design. Note that we focus on demonstrating major convergence results rather than finding the global optimal value; therefore, we run each hyperparameter tuning study for 50 iterations.

Figure 2 compares three hyperparameter tuning methodologies—Random Search, Optuna-TPE, and Optuna-BoTorch for the Random Forest Regressor (RFR). Each method's R^2 score progression is shown with solid lines, while the best score achieved at each iteration is indicated with dashed lines. A dashed gray line marks the performance of the previously defined reference model, and golden stars indicate the iteration at which each method reached its highest accuracy.

For the RFR model, all methods were able to surpass the reference R^2 score of 0.96, though the improvement was moderate. The BoTorch-based Bayesian optimization achieved the best result ($R^2 = 0.9713$) in only 19 iterations, outperforming both TPE and Random Search in terms of convergence speed and final score. Despite the lack of

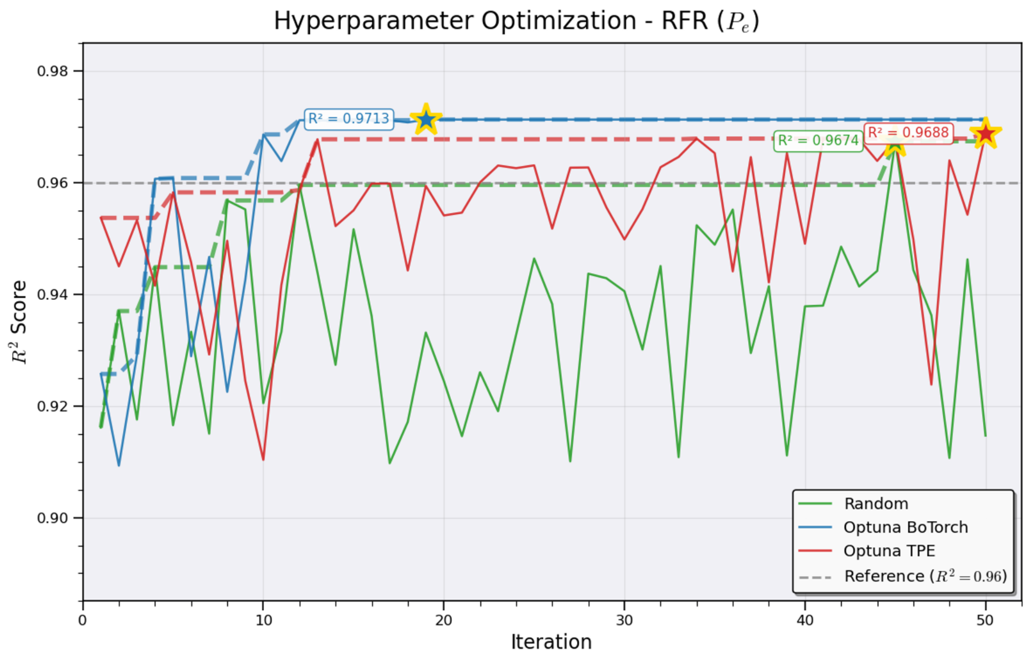

Figure 2. Hyperparameter optimization convergence for Random Forest Regressor models on NSTX electron HHFW heating data. Comparison of three methods over 50 iterations: Random sampling (green), Optuna-BoTorch (blue), and Optuna-TPE (red). Solid lines show individual R^2 scores at each iteration, dashed lines show the best R^2 score achieved up to each iteration. Stars indicate the iteration where each method achieved its highest score. Gray dashed line shows reference (Ref.) model baseline (R^2 = 0.96). The Optuna BoTorch method achieves the best performance (R^2 = 0.9713) in 19 iterations.

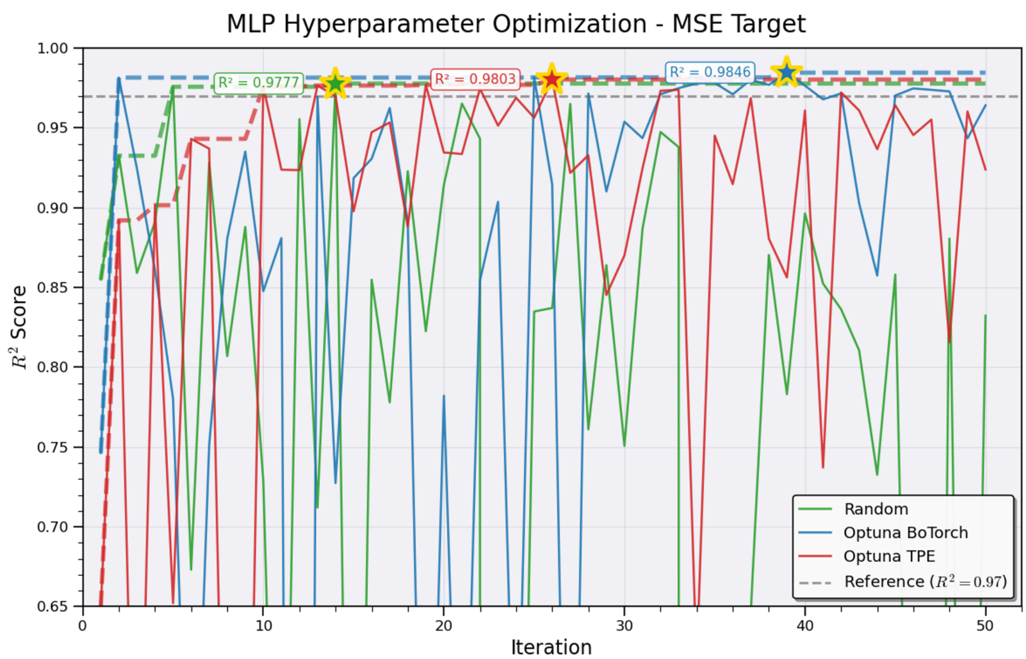

Figure 3. Hyperparameter optimization convergence for Multilayer Perceptron models on NSTX electron HHFW heating data, using MSE as optimization target. Comparison of three methods over 50 iterations: Random sampling (green), Optuna-BoTorch (blue), and Optuna-TPE (red). Solid lines show individual R^2 scores at each iteration, dashed lines show the best R^2 score achieved up to each iteration. Stars indicate the iteration where each method achieved its highest score. Gray dashed line shows reference (Ref.) model baseline (R^2 = 0.97). Results demonstrate high sensitivity of neural networks to hyperparameter choices.

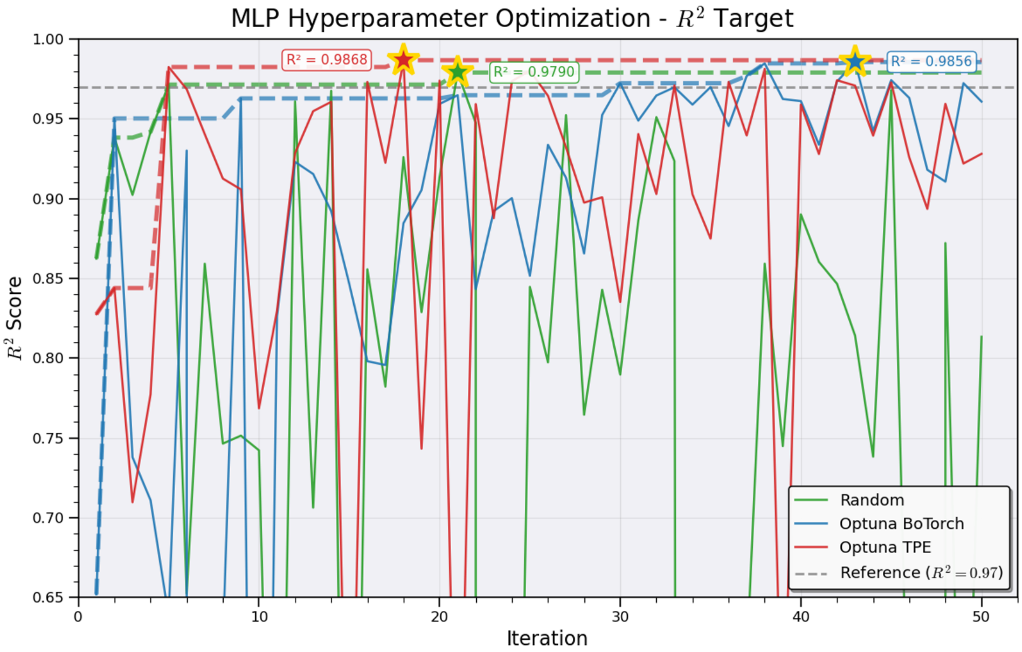

Figure 4. Hyperparameter optimization convergence for Multilayer Perceptron models on NSTX electron HHFW heating data, using R^2 as optimization target. Comparison of three methods over 50 iterations: Random sampling (green), Optuna-BoTorch (blue), and Optuna-TPE (red). Solid lines show individual R^2 scores at each iteration, dashed lines show the best R^2 score achieved up to each iteration. Stars indicate the iteration where each method achieved its highest score. Gray dashed line shows reference (Ref.) model baseline (R^2 = 0.97). All methods show a minor improvement in the best score compared to using MSE as optimization target.

structure in its exploration, Random Search never fell below R^2 = 0.91, underscoring the robustness of the RFR architecture. The optimization-based methods displayed clear convergence patterns, with BoTorch in particular exhibiting rapid and stable improvement.

Hyperparameter analysis revealed that the most impactful parameter for RFR performance was the number of estimators (`n_estimators`), which reached the upper limit of its search range (500) early in the BoTorch run. This observation is consistent with the established understanding of RFR behavior and suggests that future efforts may benefit more from allocating computational resources toward increasing estimator counts rather than extensive hyperparameter optimization. Overall, RFR proved to be relatively insensitive to hyperparameter tuning, and even simple methods and also architectures produced reliable models.

In contrast, the Multilayer Perceptron (MLP) models demonstrated significant sensitivity to hyperparameter settings, as can be noticed in Figure 3. All optimization strategies outperformed the reference model, with BoTorch again achieving the best performance. Notably, BoTorch required only two iterations to exceed the reference score, while Random Search and TPE required five and ten iterations, respectively. BoTorch approach reaches diminishing returns early in the optimization, performance plateaus while only marginal improvements are achieved in later iterations, eventually reaching the highest score among all methods.

The Optuna-BoTorch method performed best in terms of accuracy, achieving optimal performance quickly but subsequently plateauing. The Random sampling method surprisingly provided competitive performance, achieving results similar to Optuna-TPE. Optuna-TPE surpassed the reference model performance within 13 iterations but did not reach its highest performance until the final iteration, suggesting that additional iterations might reveal even better hyperparameter combinations.

The large variance in R^2 scores across iterations, especially under Random Search, highlights the difficulty of tuning neural architectures. The MLP model's behavior illustrates the complexity of its hyperparameter space and reinforces the importance of advanced hyperparameter optimization strategies. In particular, BO methods such as BoTorch offer clear advantages in the neural architecture search, yielding faster convergence and higher-quality models with minimal human intervention.

These results emphasize the distinction between hyperparameter scanning (HPS) and hyperparameter optimization (HPO): while scanning methods provide a baseline, optimization-driven approaches are essential for maximizing performance, especially in architectures as sensitive as deep neural networks.

3.3 MSE vs R² Optimization Target Analysis

An important consideration in hyperparameter optimization is the choice between different optimization targets. While neural networks often use MSE as the training loss function, R^2 often provides a more interpretable evaluation metric for regression problems.

Neural networks require differentiable loss functions for gradient-based optimization, making MSE a natural

Table 1. Comparison of hyperparameter optimization methods for RFR and MLP models, in terms of best R^2, iterations until the best score, iterations to outperform the reference model, and total time for the optimization.

Model	Method	Best R^2	Iter. (Best R^2)	Iter. (R^2>Ref.)	Total Time (s)
RFR	Random Search	0.9674	45	45	274
	TPE	0.9668	50	13	372
	BoTorch	**0.9713**	**19**	**4**	743
	Reference (Ref.)	0.96			
MLP	Random Search (MSE opt.)	0.9777	**14**	5	379
	Random Search (R^2 opt.)	0.9790	21	5	295
	TPE (MSE opt.)	0.9803	26	10	241
	TPE (R^2 opt.)	**0.9868**	18	5	242
	BoTorch (MSE opt.)	0.9846	39	**2**	234
	BoTorch (R^2 opt.)	0.9856	43	30	265
	Reference (Ref.)	0.97			

choice for training due to its smooth, convex properties that facilitate backpropagation. However, R^2 provides a more interpretable measure of performance by quantifying the fraction of variance explained by the model.

When performing hyperparameter optimization, the choice of objective function affects the optimization trajectory. Using R^2 as the optimization target focuses on maximizing explained variance, while MSE optimization minimizes absolute prediction errors. These different objectives can lead to different optimal hyperparameter configurations. Figure 4 demonstrates this comparison and shows that, overall, the outcome of the hyperparameter tuning is quite similar for both methods, while the Bayesian and TPE methods show more stable and gradual convergence. A major difference is that the final R^2 score achieved is greater when we use R^2 as a target. In particular, we notice an effect in the TPE-based result. Overall, this method shows a more gradual approach to the optimal architecture, and still unclear whether the optimization processes have reached a converged state.

Table 1 summarizes the performance improvements achieved by automated hyperparameter optimization across different model types and optimization methods. All optimization studies took on the order of a few minutes (up to $\sim$ 10 minutes) to finalize, while showing no major differences between studies except for BoTorch early convergence to a computationally expensive RFR architecture maximizing `n_estimators`. Overall, the results demonstrate that while RFR models show modest improvements with automated hyperparameter optimization, MLP models benefit significantly from proper hyperparameter tuning. The automated HPO workflow reduces the time required for hyperparameter tuning from weeks/months to minutes/hours while achieving superior performance compared to HPS approaches.

4 Conclusions and Future Work

This work introduces automated machine learning workflows that address critical bottlenecks in surrogate model development, particularly for ICRF heating applications. The automated hyperparameter optimization framework dramatically reduces model development time while demonstrating significant capabilities for efficiency gains. These results make surrogate modeling accessible to researchers without extensive machine learning expertise.

The automated data analysis pipeline transforms raw simulation datasets into actionable insights within seconds, automatically identifying critical dataset characteristics including distribution patterns, correlation structures, and data quality metrics. This capability eliminates weeks of manual exploratory analysis and ensures consistent, reproducible dataset evaluation across different physics applications.

Advanced HPO methods consistently outperformed HPS approaches based on randomized, gridded, and methodical scans. While the improvements obtained were modest in absolute terms, it should be noted that the reference models were particularly successful architectures that had already been extensively optimized. The capabilities brought by these methods are not only significant for high-precision fusion applications where small improvements in predictions can substantially impact outcomes, but also for more complex architectures where optimization via HPS methods would be unfeasible.

Bayesian optimization via BoTorch demonstrated superior convergence properties, reaching optimal performance in most scenarios both in terms of accuracy and speed. For ensemble methods like the Random Forest, we found that computational resources are better allocated toward increasing model complexity in terms of the number of estimators rather than extensive hyperparameter exploration.

Our comprehensive evaluation of optimization targets (MSE vs R^2) reveals that both approaches yield comparable results for neural architectures, providing flexibility in optimization strategy selection based on computational constraints and interpretability requirements.

These automated workflows make surrogate modeling more accessible by removing expertise barriers that previously limited adoption. The standardized analysis and optimization procedures ensure reproducible results across different research groups and enable systematic comparison of surrogate model performance across various fusion physics applications. The demonstrated reliability and speed of these methods enable previously computationally prohibitive applications including real-time plasma control optimization, uncertainty quantification for experimental planning, and large-scale parameter space exploration for scenario development.

While this work demonstrates significant advances, several important limitations should be acknowledged. Methodologically, the current framework has been optimized primarily for accuracy, without accounting for computational efficiency or model size, factors critical for real-time deployment and large-scale simulations. Future work should incorporate multi-objective optimization to balance accuracy and performance cost, and evaluate scalability with respect to data volume, spatial resolution, and parallel inference capabilities for high-performance computing workflows. Advanced hyperparameter optimization approaches, including scalable neural architecture search methods such as DeepHyper [13], offer promising strategies for identifying efficient, high-performing models. Furthermore, the integration of physics-informed constraints, automated model selection, and ensemble strategies will be essential for enabling robust performance across diverse plasma regimes.

At the application level, generalization across different plasma conditions, device geometries, and RF heating methods remains an open challenge, as current validation is restricted to a single tokamak (NSTX) and a single heating scheme (HHFW). Expanding the framework to include additional ICRF applications such as surrogate modeling of ion cyclotron wall conditioning [14] using Petra-M [15, 16] simulations in ITER-relevant scenarios and ultimately to broader areas of fusion physics beyond ICRF heating will be critical to realizing its full utility across the fusion research landscape.

In summary, this work establishes a foundation for automated surrogate model development in fusion energy sciences, offering a workflow that serves as a starting point for methodological extensions, broader validation, and applications to related projects.

5 Acknowledgments

This work was supported by the U.S. Department of Energy under contract number DE-AC02-09CH11466. This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231 using NERSC award FES m3716. 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.

References

[1] J.R. Wilson and P.T. Bonoli, Progress on ion cyclotron range of frequencies heating physics and technology in support of the International Tokamak Experimental Reactor, Phys. Plasmas **22**, 021801 (2015). https://doi.org/10.1063/1.4901090

[2] M. Brambilla, Numerical simulation of ion cyclotron waves in tokamak plasmas, Plasma Phys. Control. Fusion **41** 1, (1999). https://doi.org/10.1088/0741-3335/41/1/002

[3] A. Sanchez-Villar et al., Real-time capable modeling of ICRF heating on NSTX and WEST via machine learning approaches, Nucl. Fusion **64** 096039 (2024). https://doi.org/10.1088/1741-4326/ad645d

[4] A. Sanchez-Villar et al., Machine learning enhanced predictions of ICRF heating: Overcoming numerical limitations via data curation, Phys. Plasmas **32** 06250 (2025). https://doi.org/10.1063/5.0268341

[5] M.D. Boyer et al., Real-time capable modeling of neutral beam injection on NSTX-U using neural networks, Nucl. Fusion **59** 056008 (2019). https://doi.org/10.1088/1741-4326/ab0762

[6] G.M. Wallace et al., Towards fast and accurate predictions of radiofrequency power deposition and current profile via data-driven modelling: applications to lower hybrid current drive J. Plasma Phys. **88**, 895880401 (2022). https://doi.org/doi:10.1017/S0022377822000708

[7] B. Shahriari et al., Taking the human out of the loop: A review of Bayesian optimization. IEEE Proceedings **104** 148 (2016). https://doi.org/doi:10.1109/JPROC.2015.2494218

[8] S. Watanabe, Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance, arXiv 2304.11127 (2023). https://doi.org/10.48550/arXiv.2304.11127)

[9] T. Akiba et al., Optuna: A next-generation hyperparameter optimization framework, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining **2623** (2019). https://doi.org/10.1145/3292500.3330701

[10] M. Balandat et al., BoTorch: A framework for efficient Monte-Carlo Bayesian optimization, arXiv 1910.06403 (2020). https://doi.org/10.48550/arXiv.1910.06403

[11] F. Pedregosa et al., Scikit-learn: Machine Learning in Python *J. Mach. Learn. Res.* **12** 2825 (2011). https://dl.acm.org/doi/10.5555/1953048.2078195

[12] A. Paszke et al., PyTorch: An Imperative Style, High-Performance Deep Learning Library, Advances in Neural Information Processing Systems **32** 8024 (2019), (arXiv:) 1912.01703. https://doi.org/10.48550/arXiv.1912.01703)

[13] P. Balaprakash et al., Deephyper: Asynchronous Hyperparameter Search for Deep Neural Networks, IEEE 25th Int. Conf. on High Performance Computing (HiPC) pp. 42-51 (2018). https://doi.org/10.1109/HiPC.2018.00014

[14] T. Wauters et al., Wall conditioning in fusion devices with superconducting coils, Plasma Phys. Control. Fusion **62** 034002 (2020). https://doi.org/10.1088/1361-6587/ab5ad0

[15] S. Shiraiwa et al. RF wave simulation for cold edge plasmas using the MFEM library, EPJ Web of Conf. **157** 03048 (2017). https://doi.org/10.1051/epjconf/201715703048

[16] S. Shiraiwa et al., Finite element modeling of RF waves in fusion plasmas: progress in past decades and future role, 25th Topical Conference on Radio-Frequency Power in Plasmas, Schloss Hohenkammer, Germany, May 19-22 (2025). https://www.ipp.mpg.de/5487515/Talks