Fast regression of the tritium breeding ratio in fusion reactors

To cite this article: P Mánek *et al* 2023 *Mach. Learn.: Sci. Technol.* **4** 015008

View the article online for updates and enhancements.

MACHINE
LEARNING
Science and Technology

PAPER

Fast regression of the tritium breeding ratio in fusion reactors

P Mánek[1,2,*] , G Van Goffrier[1,*] , V Gopakumar[3] , N Nikolaou[1] , J Shimwell[3] and I Waldmann[1]

1 Department of Physics and Astronomy, University College London, Gower Street, London WC1E 6BT, United Kingdom
2 Institute of Experimental and Applied Physics, Czech Technical University, Husova 240/5, Prague 110 00, Czech Republic
3 UK Atomic Energy Authority, Culham Science Centre, OX14 3DB Abingdon, United Kingdom
* Authors to whom any correspondence should be addressed.

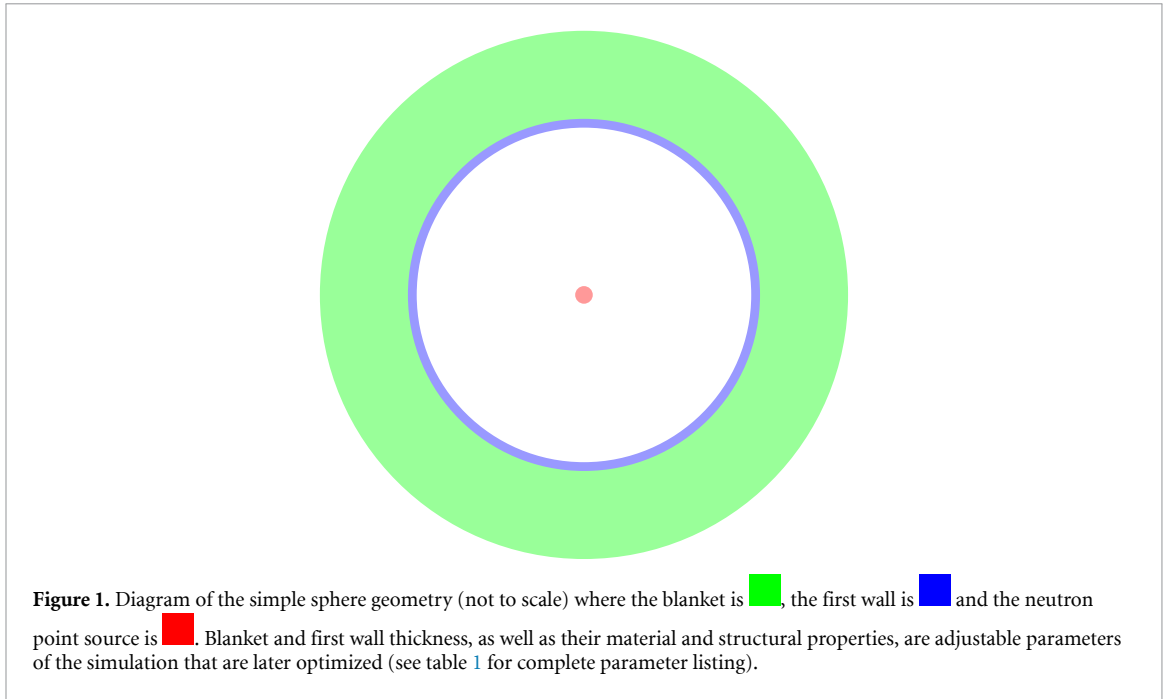E-mail: petr.manek.19@ucl.ac.uk and graham.vangoffrier.19@ucl.ac.uk

Abstract

The tritium breeding ratio (TBR) is an essential quantity for the design of modern and next-generation D-T fueled nuclear fusion reactors. Representing the ratio between tritium fuel generated in breeding blankets and fuel consumed during reactor runtime, the TBR depends on reactor geometry and material properties in a complex manner. In this work, we explored the training of surrogate models to produce a cheap but high-quality approximation for a Monte Carlo (MC) TBR model in use at the UK Atomic Energy Authority. We investigated possibilities for dimensional reduction of its feature space, reviewed 9 families of surrogate models for potential applicability, and performed hyperparameter optimization. Here we present the performance and scaling properties of these models, the fastest of which, an artificial neural network, demonstrated $R^2 = 0.985$ and a mean prediction time of $0.898\ \mu$s, representing a relative speedup of 8×10^6 with respect to the expensive MC model. We further present a novel adaptive sampling algorithm, Quality-Adaptive Surrogate Sampling, capable of interfacing with any of the individually studied surrogates. Our preliminary testing on a toy TBR theory has demonstrated the efficacy of this algorithm for accelerating the surrogate modelling process.

1. Introduction

Surrogate models were developed to resolve computational limitations in the analysis of massive datasets by replacing a resource-expensive procedure with a much cheaper approximation [1]. They are especially useful in applications where numerous evaluations of an expensive procedure are required over the same or similar domains, e.g. in the parameter optimization of a theoretical model. The term 'metamodel' proves especially meaningful in this case, when the surrogate model approximates a computational process which is itself a model for a (perhaps unknown) physical process [2]. There exists a spectrum between 'physical' surrogates which are constructed with some contextual knowledge in hand, and 'empirical' surrogates which are derived purely from samples of the underlying expensive model.

In this work, we develop a family of empirical surrogate models for the tritium breeding ratio (TBR) in an inertial confinement fusion (ICF) reactor. The expensive model that our surrogate model approximates is a Monte Carlo (MC) neutronics simulation, Paramak [3], which returns a prediction of the TBR for a given configuration of a spherical ICF reactor. Although more expensive 3D parametric models exist, we chose the Paramak simulation for its preferable speed in dataset generation in order to most fully demonstrate our methods. We quantify the success of several of our best-performing surrogate models by studying their accuracy and prediction time. We further propose an adaptive sampling algorithm (QASS) suitable for reducing the quantity of expensive samples needed to train our surrogate models.

Paramak facilitates simulation via an OpenMC neutronics workflow that is enclosed in a portable Docker container, which conveniently exposes an HTTP API using the Python 3 `flask` package. Within this setup,

Figure 1. Diagram of the simple sphere geometry (not to scale) where the blanket is ▮, the first wall is ▮ and the neutron point source is ▮. Blanket and first wall thickness, as well as their material and structural properties, are adjustable parameters of the simulation that are later optimized (see table 1 for complete parameter listing).

we employ a Muir energy distribution [4][4] around 14.06 MeV to approximate a deuterium-tritium (D-T) plasma neutron source. As illustrated in figure 1, the simulated reactor geometry was made adjustable in order to study its influence on the TBR. Nuclear data for simulation were extracted from the following sources, in order of preference: FENDL 3.1d [5]; JEFF 3.3 [6]; and ENDF/B-VII.1 [7]. To maintain a model-agnostic approach, variance reduction (VR) techniques were not used to accelerate the MC neutronics simulation [8]. It should be noted that depending on application, VR may constitute a viable alternative to the presented work.

For the remainder of section 1, we will define the TBR and further motivate our research. In section 2 we will present our methodologies for the comparison testing of a wide variety of surrogate modelling techniques, as well as defining an add-on adaptive sampling procedure QASS. After delivering the results of these approaches in section 3, we will give our final conclusions and recommendations in section 4.

1.1. Problem description

Nuclear fusion technology relies on the production and containment of an extremely hot and dense plasma containing enriched Hydrogen isotopes. The current frontier generation of fusion reactors, such as the Joint European Torus and the under-construction ITER, make use of both tritium and deuterium fuel. While at least one deuterium atom occurs for every 5000 molecules of naturally-sourced water, and may be easily distilled, tritium is extremely rare in nature. Tritium may be produced indirectly through irradiation of heavy water (D_2O) during nuclear fission, but only at very low rates which could never sustain industrial-scale fusion power.

Modern D-T reactors rely on tritium breeding blankets, specialized layers of material which partially line the reactor and produce tritium upon neutron bombardment, e.g. by:

$$\ce{^1_0 n + ^6_3 Li -> ^3_1 T + ^4_2 He} \tag{1}$$

$$\ce{^1_0 n + ^7_3 Li -> ^3_1 T + ^4_2 He + ^1_0 n.} \tag{2}$$

The TBR is defined as the ratio of tritium produced per source neutron, whose description in Paramak is facilitated by two classes of parameters (exhaustively listed in table 1). While the geometry of a given reactor is described by continuous parameters, material selections are specified by discrete categorical parameters. For all parameters, we have attempted to cover the full theoretical range of values even where those values are practically infeasible with current technology (e.g. packing fractions close to 1). Simulating broadly around

---

[4] A bug in the Muir distribution involving erroneous normal sampling was recently uncovered (see https://github.com/openmc-dev/openmc/pull/1670), but is disregarded in the present methods-focused work.

Table 1. Input parameters supplied to Paramak and surrogates in alphabetical order. Groups of fractions marked†‡ are independently required to sum to 1.

	Parameter name (abbreviation)	Domain
Blanket	Breeder fraction†	$[0, 1]$
	Breeder ^{6}Li enrichment fraction	$[0, 1]$
	Breeder material (BBM)	$\{Li_2TiO_3, Li_4SiO_4\}$
	Breeder packing fraction	$[0, 1]$
	Coolant fraction†	$[0, 1]$
	Coolant material (BCM)	$\{D_2O, H_2O, He\}$
	Multiplier fraction†	$[0, 1]$
	Multiplier material (BMM)	$\{Be, Be_{12}Ti\}$
	Multiplier packing fraction	$[0, 1]$
	Structural fraction†	$[0, 1]$
	Structural material (BSM)	$\{SiC, eurofer\}$
	Thickness	$[0, 500]$cm
First wall	Armour fraction‡	$[0, 1]$
	Coolant fraction‡	$[0, 1]$
	Coolant material (FCM)	$\{D_2O, H_2O, He\}$
	Structural fraction‡	$[0, 1]$
	Structural material (FSM)	$\{SiC, eurofer\}$
	Thickness	$[0, 20]$cm

typical values of parameters also improves the accuracy of the model nearer to typical values, and further aids in demonstrating the robustness of constructed models.

In our work, we set out to produce a fast TBR surrogate model, which takes the same input parameters as the MC model used in Paramak and approximates its output with the greatest achievable regression performance, while also minimizing the required quantity of expensive-model samples needed for training. This represents a significant step forward in computational fusion-reactor design, as speed-ups achieved in TBR evaluation can lead to a speed-up in numerical optimization of reactor parameters, although such optimization is beyond the scope of the present work.

2. Methodology

Labeling the expensive Paramak model $f(x)$, a surrogate model is a function $\hat{f}(x)$ such that $f(x)$ and $\hat{f}(x)$ minimize a selected dissimilarity metric. In order to be considered *viable*, $\hat{f}(x)$ is required to achieve an expected evaluation time lower than that of $f(x)$. In this work, we consider two methods of producing viable surrogates: (a) a conventional decoupled approach, which evaluates $f(x)$ on a set of uniformly-random samples and trains surrogates in a supervised scheme, and (b) an adaptive approach, which attempts to compensate for localized regression performance insufficiencies by interleaving multiple epochs of sampling and training. Several high-accuracy and deployment-ready surrogate models are developed using the decoupled approach, and their performance characterized numerically, while the adaptive approach is studied exclusively as a proof-of-concept.

We selected several state-of-the-art regression algorithms to perform surrogate training on sampled point sets. Listed in table 2, these implementations define nine surrogate families which are detailed in section 3. We note that each presented algorithm defines hyperparameters that may influence its performance. Their problem-specific optimal values are searched within the scope of this work, in particular in Experiments 1 & 2 that are outlined in section 2.1.

To compare the quality of the produced surrogates, we define a variety of metrics listed in table 3. For regression performance analysis, we include a selection of absolute metrics (MAE, S) to assess the models' approximation capability and to set practical bounds on the expected uncertainty of their predictions. In addition, we also track relative measures (R^2, $R^2_{\text{adj.}}$) that are better-suited for comparison between this work and others, as they are invariant with respect to the selected domain and image space. For analysis of computational complexity, surrogates are assessed in terms of wall time (captured by the Python 3 `time` package). This is motivated by common practical use-cases of our work, where surrogate models are trained as replacements for Paramak. All times reported (training, test, evaluation) are normalized by the corresponding dataset size, i.e. correspond to 'time to process a single datapoint.'

Even though some surrogates support acceleration by means of parallelization, we used non-parallelized implementations. The only exception to this is the ANN family, which requires a considerable amount of processing power for training on conventional CPU architectures. Lastly, to prevent undesirable bias by

Table 2. Considered surrogate model families, their selected abbreviations and implementations. $\mathcal{H}$ denotes the set of hyperparameters, family-dependent priors that control the learning process, and are tuned separately. Families with fewer hyperparameter represent a smaller surrogate domain to explore.

Surrogate family	Abbr.	Impl.	$|\mathcal{H}|$
Support vector machines [9]	SVM	SciKit [10]	3
Gradient boosted trees [11–13]	GBT	SciKit	11
Extremely randomized trees [14]	ERT	SciKit	7
AdaBoosted decision trees[a] [15]	ABT	SciKit	3
Gaussian process regression [16]	GPR	SciKit	2
k nearest neighbours	KNN	SciKit	3
Artificial neural networks	ANN	Keras [17]	2
Inverse distance weighing [18]	IDW	SMT [19]	1
Radial basis functions	RBF	SMT	3

[a] Note that ABTs can be viewed as a subclass of GBTs.

Table 3. Metrics recorded in experiments. In formulations, we work with a training set of size N_0 and a test set of size N, values $y^{(i)} = f(x^{(i)})$ and $\hat{y}^{(i)} = \hat{f}(x^{(i)})$ denote images of the ith testing sample in Paramak and the surrogate respectively. The mean $\bar{y} = N^{-1}\sum_{i=1}^{N} y^{(i)}$ and P is the number of input features.

Regression performance metrics	Notation	Mathematical formulation
Mean absolute error	MAE	$N^{-1}\sum_{i=1}^{N}|y^{(i)} - \hat{y}^{(i)}|$
Standard deviation of error	S	$\text{StdDev}_{i=1}^{N}\left\{|y^{(i)} - \hat{y}^{(i)}|\right\}$
Coefficient of determination	R^2	$1 - \sum_{i=1}^{N}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 \left[\sum_{i=1}^{N}\left(y^{(i)} - \bar{y}\right)^2\right]^{-1}$
Adjusted R^2	$R^2_{\text{adj.}}$	$1 - (1 - R^2)(N-1)(N-P-1)^{-1}$
Computational complexity metrics		
Mean training time	$\bar{t}_{\text{trn.}}$	(wall training time of $\hat{f}(x))N_0^{-1}$
Mean prediction time	$\bar{t}_{\text{pred.}}$	(wall prediction time of $\hat{f}(x))N^{-1}$
Relative speedup	ω	(wall evaluation[a] time of $f(x)$) $(N\bar{t}_{\text{pred.}})^{-1}$

[a] This corresponds to evaluation of Paramak on all points of the test set. In surrogates, the equivalent time period is referred to as the "prediction time."

training set selection, all reported metrics are obtained via five-fold cross-validation. In this setting, a sample set is uniformly divided into five disjoint folds, each of which is used as a test set for models trained on the remaining four. Having repeated the same experiment for each division, the overall value of individual metrics is reported in terms of their mean and standard deviation over all folds.

2.1. Decoupled approach

Experiments related to the decoupled approach are organized in four parts, further described in this section. In summary, we aim to optimize the hyperparameters of each surrogate family separately, and later compare the best results between families.

The objective of Experiment 1 is to simplify the regression task for surrogates prone to suboptimal performance in discrete spaces. To this end, training points are filtered to a single selected discrete feature assignment, and surrogates are trained only on the remaining continuous features. This is repeated several times to explore variances in behavior, particularly in four distinct assignments that are obtained by setting blanket and first wall coolant materials to one of: $\{H_2O, He\}$. Experiment 2 conventionally measures surrogate performance on the full feature space without any parameter restrictions. In both experiments, hyperparameter tuning is facilitated by Bayesian optimization [20], where we select the hyperparameter configuration that produces the model that maximizes R^2. The process is terminated after 1000 iterations or 2 days, whichever condition is satisfied first. The results of Experiments 1 & 2 are depicted in figures 3 and 4 respectively, and described in section 3.1.1.

In Experiment 3, the twenty best-performing hyperparameter configurations for each model family are used to train surrogates on sets of various sizes to investigate their scaling properties. In particular, we track the metrics from table 3 as functions of training set size (1, 2, 5, 10, 12, 15 and 20 thousands of samples) individually for each family. This allows their comparison based on observed trends, and estimation of optimal training set sizes. The results of this experiment are shown in figure 5 and discussed in section 3.1.2.

Figure 2. Schematic of QASS algorithm.
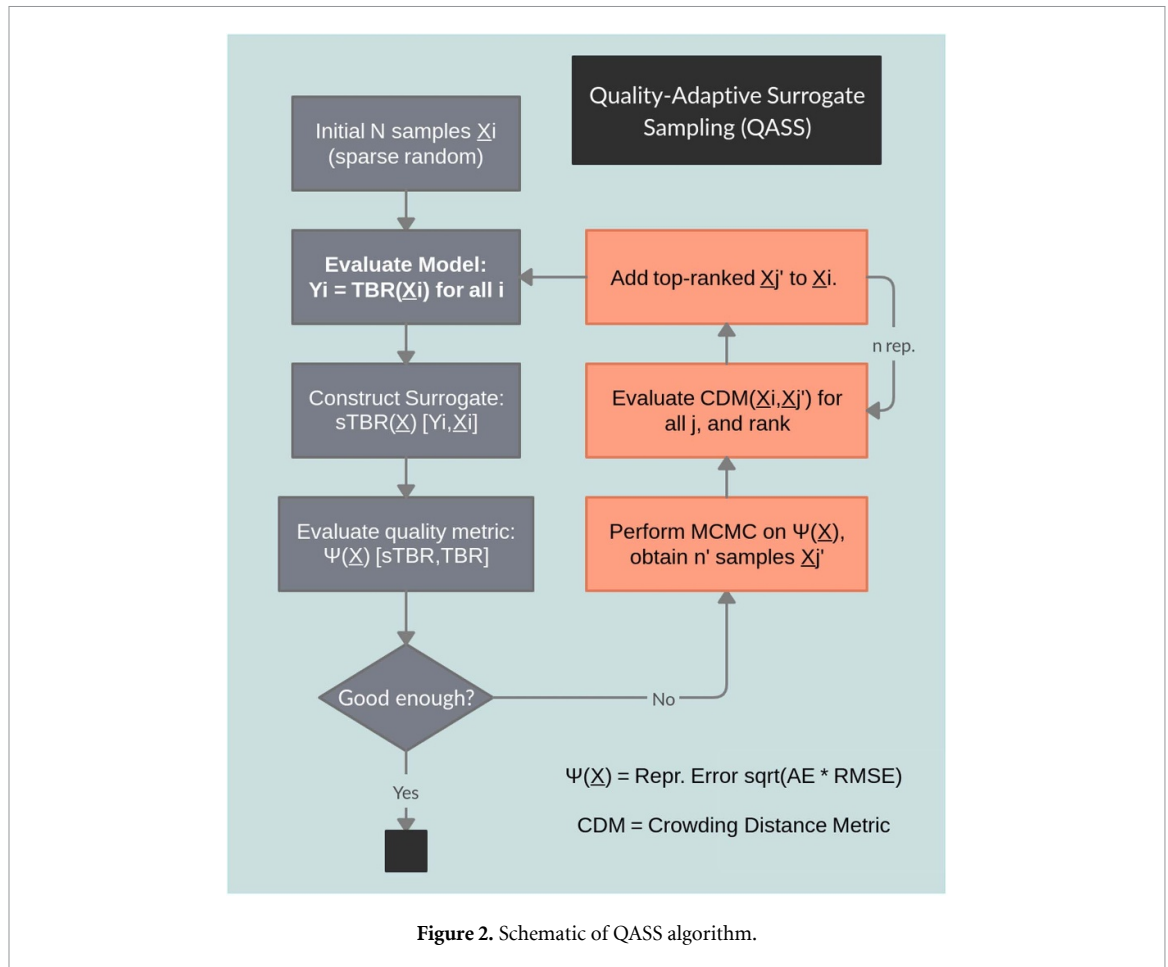
Finally, Experiment 4 aims to produce surrogates suitable for practical use by retraining selected well-scaling instances on large training sets. The results of this process are displayed in figure 6 and in table 5, and summarized in section 3.1.3.

2.2. Adaptive approach

Adaptive sampling techniques appear frequently in the literature and have been specialized for surrogate modelling, where precision is implicitly limited by the quantity of training samples which are available from the expensive model. Garud's [21] 'Smart Sampling Algorithm' achieved notable success by incorporating surrogate quality and crowding distance scoring to identify optimal new samples, but was only tested on a single-parameter domain. We theorized that a nondeterministic sample generation approach, built around Markov Chain Monte Carlo methods (MCMC), would fare better for high-dimensional models by more thoroughly exploring all local optima in the feature space. MCMC produces each sample point according to a jump step drawn from a shared proposal distribution. These sample points will converge to a desired posterior distribution, so long as the acceptance probability meets certain statistical criteria (see [22] for a review).

Many researchers have embedded surrogate methods into MCMC strategies for parameter optimization [23, 24], in particular the ASMO-PODE algorithm [25] which makes use of MCMC-based adaptive sampling. Our approach draws inspiration from ASMO-PODE, but instead uses MCMC to generate samples which increase surrogate precision throughout the entire parameter space.

We designed the quality-adaptive surrogate sampling algorithm (QASS, figure 2) to iteratively increment the training/test set with sample points which maximize surrogate error and minimize a crowding distance metric (CDM) [26] in feature space. Error maximization is desirable for these sample points because it identifies regions of parameter space where the surrogate most needs to be improved. On each iteration following an initial training of the surrogate on N uniformly random samples, the surrogate was trained and absolute error calculated. MCMC was then performed to sample the error function generated by performing nearest-neighbor interpolation on these test error points. The resultant samples were culled by 50%

according to the CDM, and then the *n* highest-error candidates were selected for reintegration with the training/test set, beginning another training epoch. Validation was also performed during each iteration on independent, uniformly-random sample sets.

3. Results

3.1. Decoupled approach

3.1.1. Hyperparameter tuning
The results displayed in figure 3 indicate that in the first, simplified case GBTs clearly appear to be the most accurate as well as the fastest surrogate family in terms of mean prediction time. Following that, we note that ERTs, SVMs and ANNs also achieved satisfactory results with respect to both examined metrics. In addition, prediction times of GBTs and SVMs show relatively lower variance than those of ERTs and ANNs. Even though the remainder of tested surrogate families do not exhibit prohibitive complexity, their regression performance fall below the average.

Comparing these results with those of the second, unrestricted experiment (shown in figure 4), we observe that many surrogate families consistently underperform. The least affected models appear to be GBTs, ANNs and ERTs, which are known to be capable of capturing relationships involving mixed feature types that were deliberately withheld in the first experiment. With only negligible differences, the first two of these families appear to be tied for the best performance as well as the shortest prediction time. We observe that ERTs and RBFs also demonstrated satisfactory results, clearly outperforming the remaining surrogates in terms of regression performance, and in some cases also prediction time.

Following both hyperparameter tuning experiments, we conclude that while domain restrictions employed in the first case have proven effective in improving the regression performance of some methods, their performance fluctuates considerably depending on the selected slices. For instance, the variance in SVM performance in slice 1 is much lower than in slices 2–3, and both KNNs and ABTs perform much better in slices 1–2 than in slice 3. Furthermore, in all instances the best results are achieved by families of surrogates that do not benefit from this restriction (GBTs, ANNs, ERTs).

3.1.2. Scaling benchmark
The results shown in figure 5 suggest that in terms of regression performance the most accurate families from the previous experiments consistently maintain their relative advantage over others, even as more training points are introduced. While such families achieve nearly comparable performance on the largest dataset, in the opposite case tree-based ensemble approaches clearly outperform ANNs. For instance, GBTs achieve $MAE = 0.107$, nearly half of the $MAE = 0.186$ achieved by ANNs, representing a clear benefit given vastly disparate training and prediction times. This trend continues for set sizes up to 6000.

Consistent with our expectations, the shortest training times were achieved by instance-based learning methods (KNN, IDW) that are trained trivially at the expense of increased lookup complexity later during prediction. Furthermore, we observe that the majority of tree-based ensemble algorithms also perform and scale well, unlike RBFs and GPR which appear to behave superlinearly. We note that ANNs, which are the only family to utilize parallelization during training, show an inverse scaling characteristic. We suspect that this effect may be caused by a constant multi-threading overhead that dominates the training process on relatively small sets.

Finally, all tested families with the exception of previously mentioned instance-based models offered desirable prediction times. Analogous to previous experiments, GBTs, ABTs and ANNs appeared to be tied, as they not only exhibited comparable times but also similar scaling slopes. After those, we note a clear hierarchy of ERTs, SVMs, GPR and RBFs, trailed by IDW and KNNs.
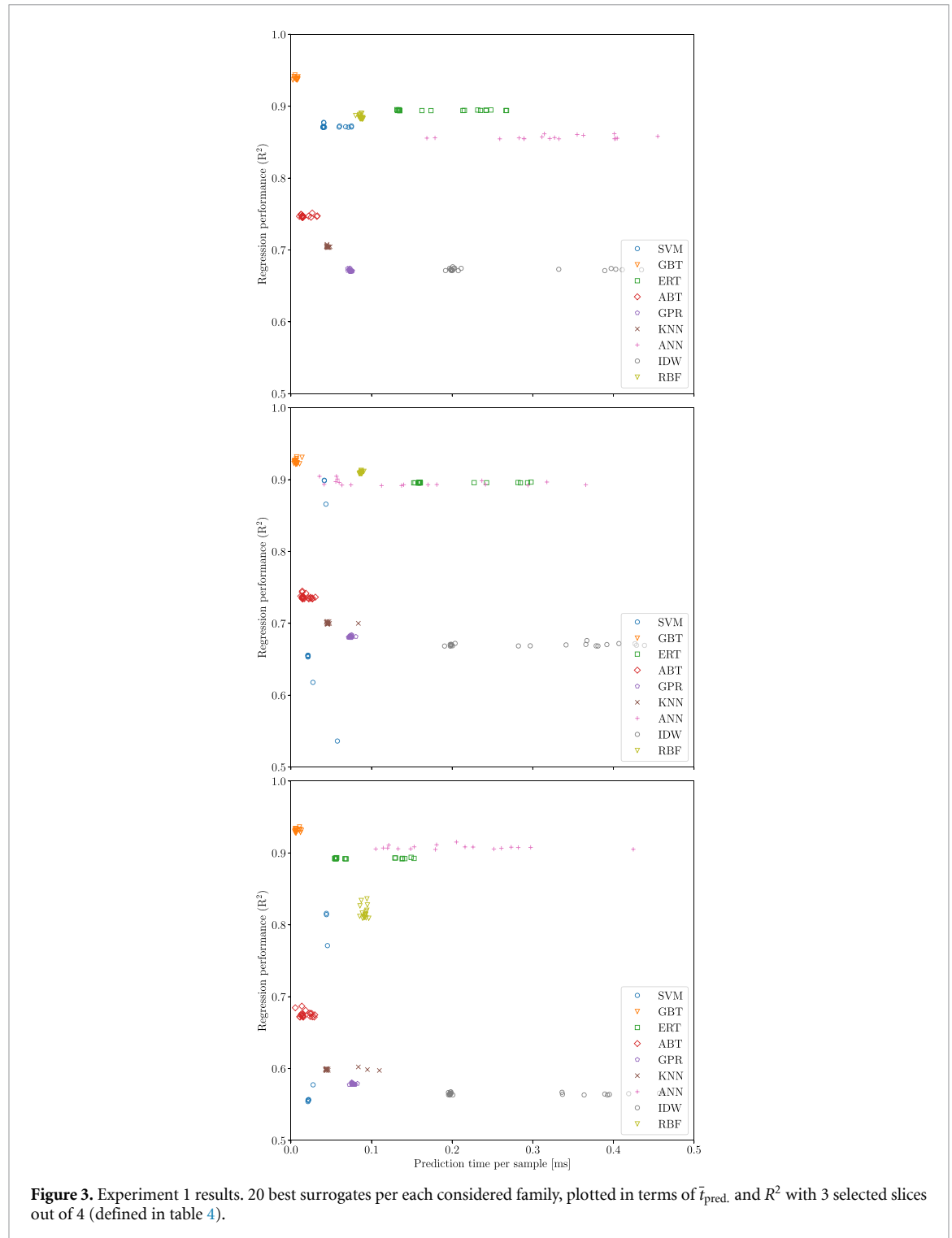
3.1.3. Model comparison
In Experiment 4, we aim to create models that yield: (a) the best regression performance regardless of other features, (b) acceptable performance with the shortest mean prediction time, or (c) acceptable performance with the smallest training set. To this end, we trained 8 surrogates that are presented in figure 6 and table 5. We compared these surrogates with the baseline represented by Paramak per-sample evaluation time $\bar{t}_{\text{eval.}} = 7.777\,049\,573\,054\,314 \pm 2.810\,359\,210\,393\,0337\,\text{s}$, which was measured earlier on a set of 500 000 samples.

Having selected ANNs, GBTs, ERTs, RBFs and SVMs based on the results of Experiments 2 & 3, we utilized the best-performing hyperparameters. In pursuit of goal (a), the best approximator (Model 1, ANN) achieved $R^2 = 0.998$ and mean prediction time $\bar{t}_{\text{pred.}} = 1.124\,\mu\text{s}$. These correspond to a standard

Table 4. Slices 1–4 of the domain space (discrete parameter assignments) explored in Experiment 1. Columns correspond to abbreviated parameter names listed in table 1.

BBM	BCM	BMM	BSM	FCM	FSM
Li_4SiO_4	H_2O	$Be_{12}Ti$	eurofer	H_2O	eurofer
Li_4SiO_4	He	$Be_{12}Ti$	eurofer	H_2O	eurofer
Li_4SiO_4	H_2O	$Be_{12}Ti$	eurofer	He	eurofer
Li_4SiO_4	He	$Be_{12}Ti$	eurofer	He	eurofer

Figure 3. Experiment 1 results. 20 best surrogates per each considered family, plotted in terms of $\bar{t}_{pred.}$ and R^2 with 3 selected slices out of 4 (defined in table 4).
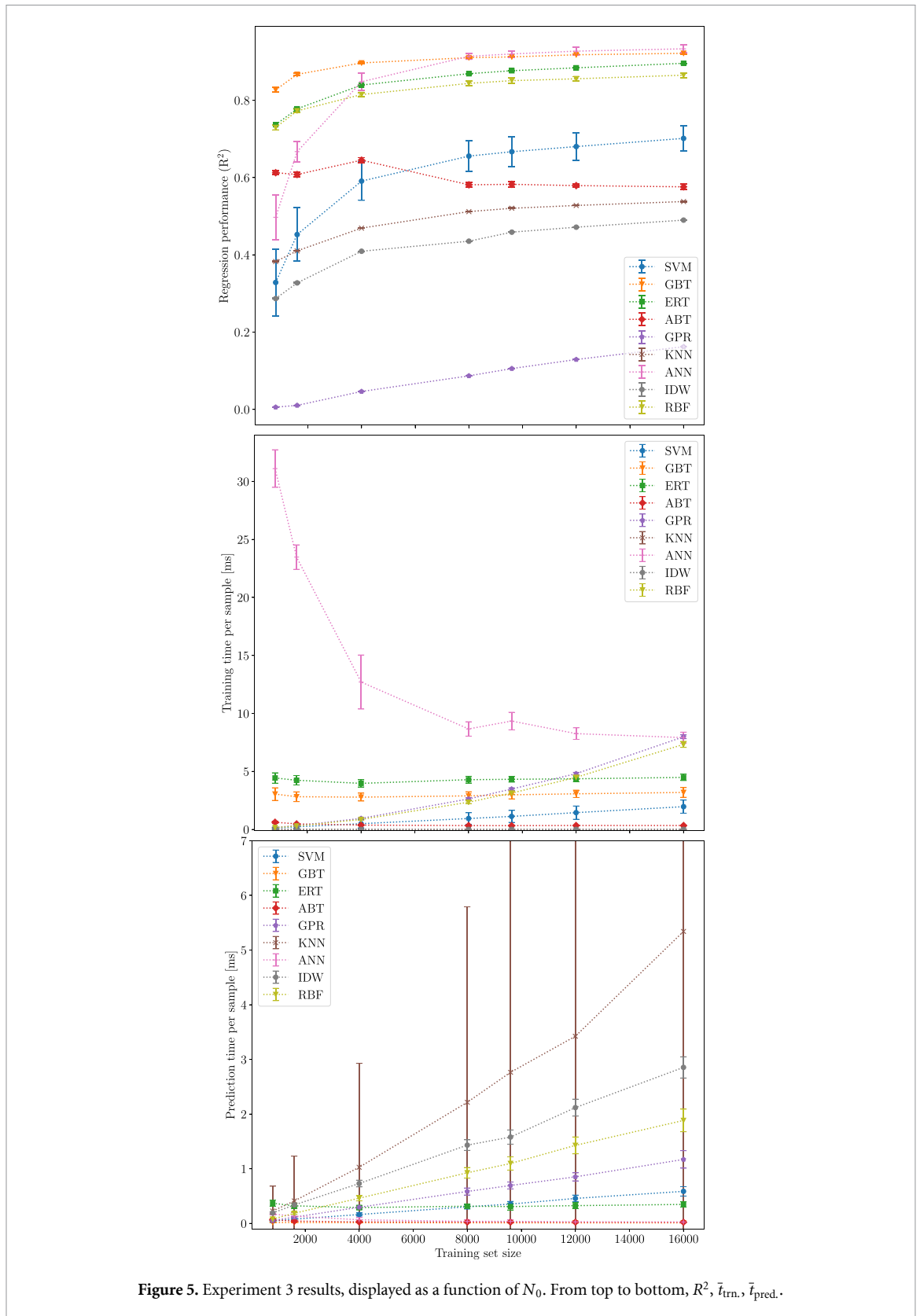
Figure 4. Experiment 2 results, plotted analogously to figure 3.

error $S = 0.013$ and a relative speedup $\omega = 6.92 \times 10^6$ with respect to Paramak. Satisfying goal (b), the fastest model (Model 2, ANN) achieved $R^2 = 0.985$, $\bar{t}_{\text{pred.}} = 0.898$ μs, $S = 0.033$ and $\omega = 8.66 \times 10^6$. While these surrogates were trained on the entire available set of 500 000 datapoints, to satisfy goal (c) we also trained a more simplified model (Model 4, GBT) that achieved $R^2 = 0.913$, $\bar{t}_{\text{pred.}} = 6.125$ μs, $S = 0.072$ and $\omega = 1.27 \times 10^6$ with a set of size only 10 000.

Overall we found that due to their superior performance, boosted tree-based approaches seem to be advantageous for fast surrogate modelling on relatively small training sets (up to the order of 10^4). Conversely, while neural networks perform poorly in such a setting, they dominate on larger training sets (at least of the order of 10^5) both in terms of regression performance and mean prediction time.

3.2. Adaptive approach

In order to test our QASS prototype, several functional toy theories for TBR were developed as alternatives to the expensive MC model. QASS performance was verified by training an ANN on these theories for varied quantities of initial, incremental, and MCMC candidate samples. By far the most useful of these was the following sinusoidal theory, as ANNs trained on this model demonstrated similar performance to those on the expensive MC model:

$$\text{TBR} = |C|^{-1} \sum_{i \in C} \left[1 + \sin(2\pi n(x_i - 1/2)) \right], \tag{3}$$

where C denotes the continuous parameter space, and n is an adjustable wavenumber parameter.

An increase in initial samples with increment held constant had a strong impact on final surrogate precision, an early confirmation of basic functionality. An increase in MCMC candidate samples was seen to have a positive but very weak effect on final surrogate precision, suggesting that the runtime of MCMC on each iteration could be limited for increased efficiency. We also found that an optimum increment exists and is monotonic with initial sample quantity, above or below which models showed slower improvement on both the training and evaluation sets, and a larger minimum error on the evaluation set. This performance distinction will be far more significant for an expensive model such as Paramak, where the number of sample evaluations is the primary computational bottleneck.

A plateau effect in surrogate error on the evaluation set was universal to all configurations, and initially suspected to be a residual effect of retraining the same ANN instance without adjustment to data normalization. A 'Goldilocks scheme' for checking normalization drift was implemented and tested, but did

Figure 5. Experiment 3 results, displayed as a function of N_0. From top to bottom, R^2, $\bar{t}_{\text{trn.}}$, $\bar{t}_{\text{pred.}}$.

not affect QASS performance. Schemes in which the ANN is periodically retrained were also discarded, as the retention of network weights from one iteration to the next was demonstrated to greatly benefit QASS efficiency. Further insight came from direct comparison between QASS and a baseline scheme with uniformly random incremental samples, shown in figure 7.
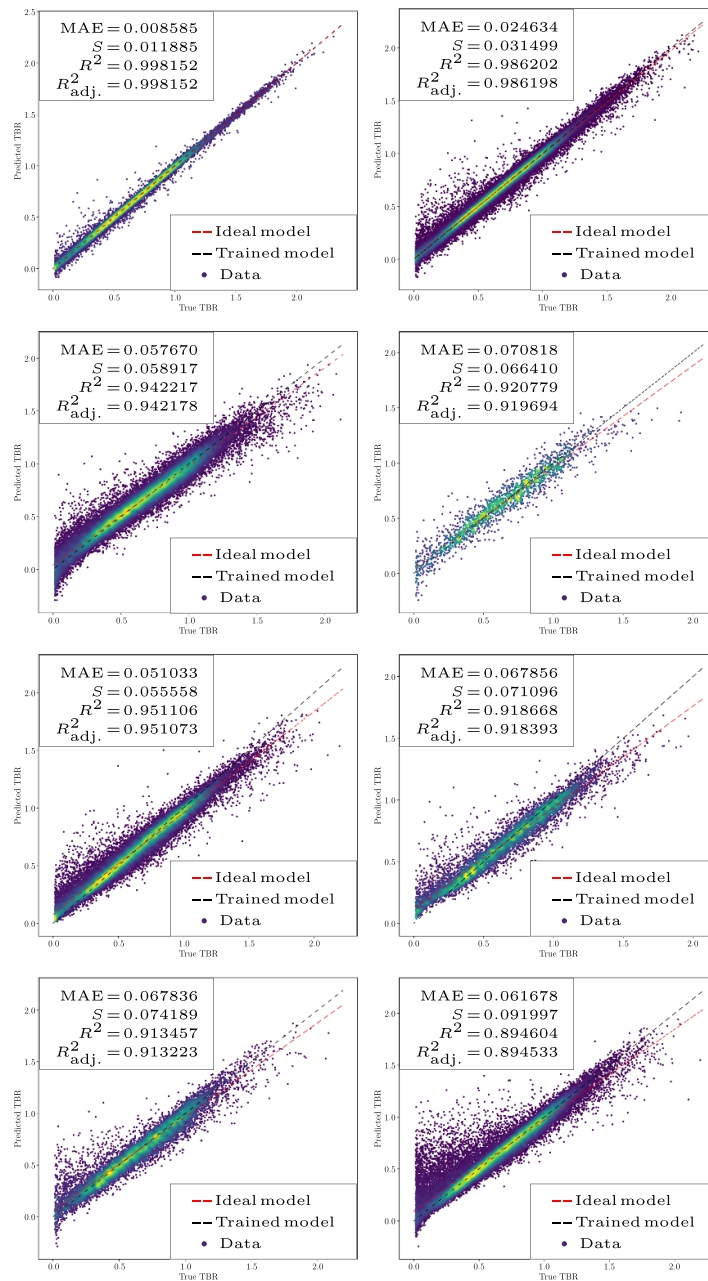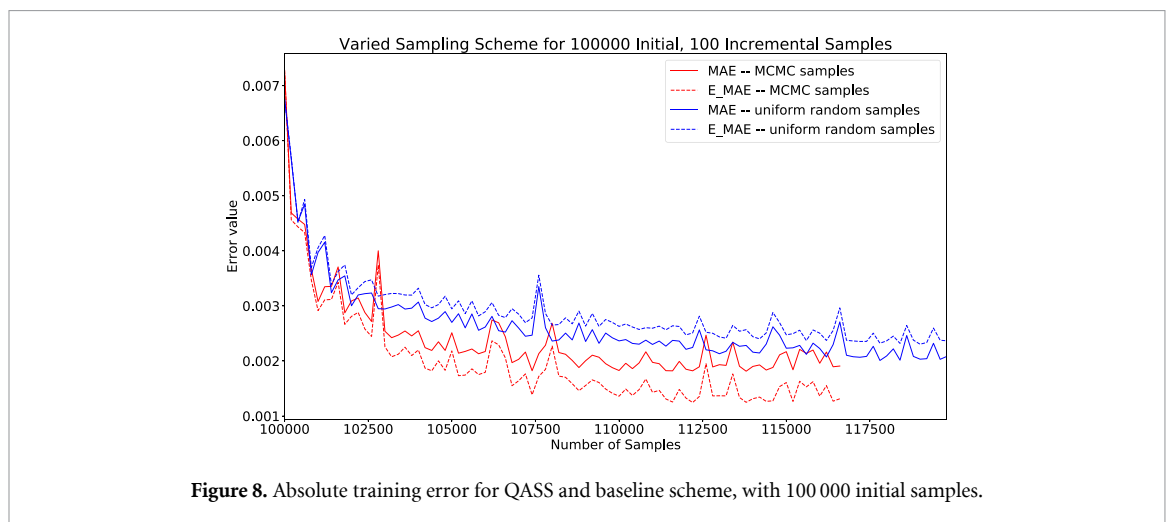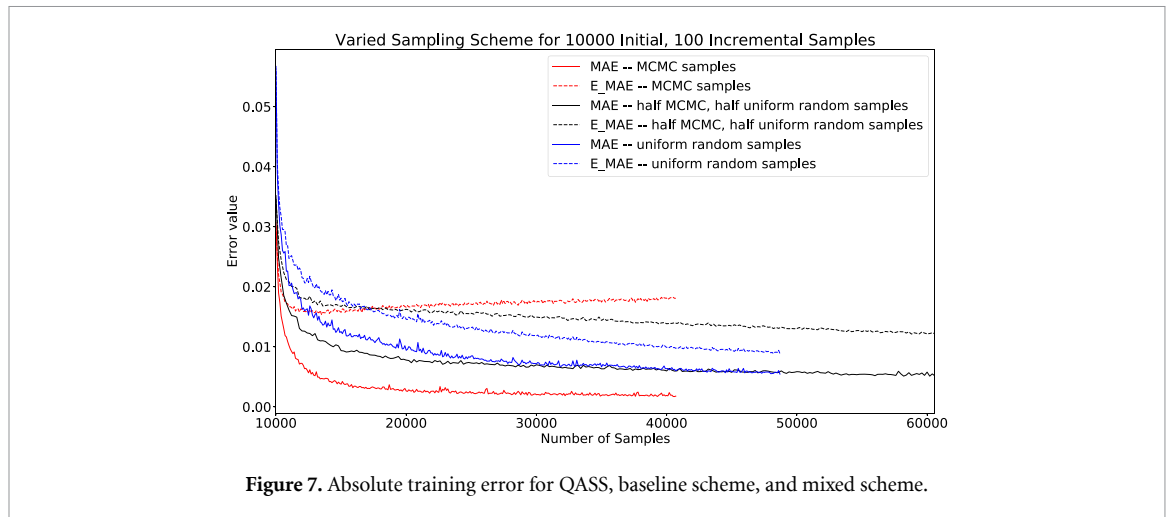
Figure 6. Regression performance of Models 1–8 (from left to right, top to bottom) in Experiment 4, viewed as true vs. predicted TBR on a test set of a selected cross-validation fold. Points are colored by density.

Such tests revealed that while QASS has unmatched performance on its own adaptively-sampled training set, it is outperformed by the baseline scheme on uniformly-random evaluation sets. We inferred that while QASS excels in learning the most strongly peaked regions of the TBR theory, this comes at the expense of precision in broader, smoother regions where uniformly random sampling suffices. Therefore a mixed scheme was implemented, with half MCMC samples and half uniformly-random samples incremented on each iteration, which is also shown in figure 7. An increase in initial sample size was also observed to improve precision in these smooth regions of the toy theory, as the initial samples were uniformly-random. As shown in figure 8, with 100 000 initial samples it was possible to obtain a $\sim 40\%$ decrease in error as compared to the baseline scheme, from 0.0025 to 0.0015 mean averaged error. Comparing at the point of termination for QASS, this corresponds to a $\sim 6\%$ decrease in the number of total samples needed to train a surrogate model while achieving the same error.

Table 5. Results of Experiment 4. Here, means and standard deviations are reported over five cross-validation folds, $|\mathcal{T}|$ denotes cross-validation set size ($\times 10^3$) and ω is a relative speedup with respect to $\bar{t}_{\text{eval.}} = 7.777\,049\,573\,054\,314 \pm 2.810\,359\,210\,393\,0337\text{s}$ measured in Paramak over 500 000 samples. The best-performing method(s) under each metric are highlighted in bold.

Model	$|\mathcal{T}|$	Regression performance				Computational complexity		
		MAE [TBR]	S [TBR]	R^2 [rel.]	$R^2_{\text{adj.}}$ [rel.]	$\bar{t}_{\text{trn.}}$ [ms]	$\bar{t}_{\text{pred.}}$ [ms]	ω [rel.]
1 (ANN)	500.0	**0.008 777 ± 0.000 269**	**0.012 512 ± 0.000 535**	**0.997 995 ± 0.000 150**	**0.997 995 ± 0.000 150**	3.658 670 ± 0.035 377	0.001 124 ± 0.000 062	6 916 416×
2 (ANN)	500.0	0.025 271 ± 0.000 719	0.033 191 ± 0.001 331	0.985 065 ± 0.001 069	0.985 061 ± 0.001 069	2.989 270 ± 0.026 018	**0.000 898 ± 0.000 037**	**8 659 251×**
3 (GBT)	200.0	0.058 242 ± 0.000 528	0.059 233 ± 0.000 337	0.941 086 ± 0.000 844	0.941 046 ± 0.000 845	2.220 903 ± 0.010 040	0.006 647 ± 0.000 218	1 169 933×
4 (GBT)	10.0	0.070 804 ± 0.001 843	0.071 597 ± 0.003 491	0.913 014 ± 0.006 027	0.911 823 ± 0.006 110	**1.621 323 ± 0.007 535**	0.006 125 ± 0.000 291	1 269 777×
5 (ERT)	200.0	0.051 286 ± 0.000 288	0.056 296 ± 0.000 486	0.950 486 ± 0.000 739	0.950 453 ± 0.000 739	2.634 038 ± 0.009 780	0.214 195 ± 0.003 631	36 308×
6 (ERT)	40.0	0.067 868 ± 0.000 302	0.071 722 ± 0.000 461	0.917 489 ± 0.001 009	0.917 210 ± 0.001 009	2.368 460 ± 0.005 461	0.187 990 ± 0.008 412	41 370×
7 (RBF)	50.0	0.068 405 ± 0.000 813	0.076 889 ± 0.001 908	0.909 963 ± 0.003 076	0.909 719 ± 0.003 084	3.452 536 ± 0.018 824	1.512 068 ± 0.016 163	5143×
8 (SVM)	200.0	0.062 351 ± 0.000 493	0.094 484 ± 0.001 577	0.890 579 ± 0.002 923	0.890 505 ± 0.002 925	33.346 811 ± 0.381 933	2.415 167 ± 0.010 751	3220×

Figure 7. Absolute training error for QASS, baseline scheme, and mixed scheme.

Figure 8. Absolute training error for QASS and baseline scheme, with 100 000 initial samples.

4. Conclusion

We employed a broad spectrum of data analysis and machine learning techniques to develop a library of fast and high-quality surrogate models for the expensive Paramak TBR model. After reviewing 9 surrogate model families, examining their behaviour on a constrained and unrestricted feature space, and studying their scaling properties, we retrained the best-performing instances to produce properties desirable for practical use. The fastest surrogate, an artificial neural network trained on 500 000 datapoints, attained an $R^2 = 0.985$ with mean prediction time of 0.898 μs, representing a relative speedup of $8 \cdot 10^6$ with respect to Paramak. Furthermore, we demonstrated the possibility of achieving comparable results using only a training set of size 10 000.

We additionally developed a novel adaptive sampling algorithm, QASS, capable of interfacing with any of the surrogate models presented in this work. Preliminary testing on a toy theory, qualitatively comparable to Paramak, demonstrated the effectiveness of QASS and key behavioral trends. With 100 000 initial samples and 100 incremental samples per iteration, QASS achieved a $\sim$40% decrease in surrogate error compared to a baseline random sampling scheme. Further optimization over the hyperparameter space has strong potential to increase this performance by further reduction of necessary expensive samples, in particular by decreasing the required quantity of initial samples. This will allow for future deployment of QASS on top of any of our most effective identified TBR surrogate models.

Data availability statement

Relevant source code, model instances and datasets are freely available online as well as a more detailed technical report [27, 28].

Acknowledgments

ORCID iDs

P Mánek ● https://orcid.org/0000-0003-4306-0209
V Gopakumar ● https://orcid.org/0000-0003-0904-3448
N Nikolaou ● https://orcid.org/0000-0001-8453-7574
J Shimwell ● https://orcid.org/0000-0001-6909-0946
I Waldmann ● https://orcid.org/0000-0002-4205-5267

References

[1] Søndergaard J 2003 Optimization using surrogate models *PhD Thesis* Technical University of Denmark
[2] Myers R and Montgomery D 2002 *Response Surface Methodology: Product and Process Optimization Using Designed Experiments* 2nd edn (New York: Wiley)
[3] UKAEA 2020 Paramak (available at: https://github.com/ukaea/paramak)
[4] OpenMC 2019 Muir energy spectrum (available at: https://docs.openmc.org/en/stable/pythonapi/generated/openmc.stats.Muir.html)
[5] FENDL-3.1d: fusion evaluated nuclear data library ver.3.1d (available at: www-nds.iaea.org/fendl/)
[6] The joint evaluated fission and fusion file (JEFF) version 3.3 (available at: www.oecd-nea.org/dbdata/jeff/jeff33/)
[7] Chadwick M *et al* 2011 ENDF/B-VII.1 Nuclear Data for Science and Technology: cross sections, covariances, fission product yields and decay data *Nucl. Data Sheets* **112** 2887–996
[8] Kleijnen J P C, Ridder A A N and Rubinstein R Y 2013 *Variance Reduction Techniques in Monte Carlo Methods* (Boston, MA: Springer US) pp 1598–1610
[9] Fan R E, Chang K W, Hsieh C J, Wang X R and Lin C J 2008 *J. Mach. Learn. Res.* **9** 1871–4 (available at: http://jmlr.org/papers/v9/fan08a.html)
[10] Pedregosa F *et al* 2011 *J. Mach. Learn. Res.* **12** 2825–30
[11] Friedman J H 2001 *Ann. Stat.* **29** 1189–232
[12] Friedman J 1999 Stochastic gradient boosting technical report
[13] Hastie T, Tibshirani R and Friedman J 2009 *The Elements of Statistical Learning: Data Mining, Inference and Prediction* (New York: Springer) (https://doi.org/10.1007/978-0-387-84858-7)
[14] Geurts P, Ernst D and Wehenkel L 2006 *Mach. Learn.* **63** 3–42
[15] Drucker H 1997 Improving regressors using boosting techniques *ICML* **97** 107–15
[16] Williams C K and Rasmussen C E 2006 *Gaussian Processes for Machine Learning* vol 2 (Cambridge, MA: MIT Press)
[17] Chollet F *et al* 2015 Keras (available at: https://keras.io)
[18] Shepard D 1968 A two-dimensional interpolation function for irregularly-spaced data *Proc. 1968 23rd ACM National Conf.* pp 517–24
[19] Bouhlel M A, Hwang J T, Bartoli N, Lafage R, Morlier J and Martins J R R A 2019 *Adv. Eng. Softw.* **135** 102662
[20] Močkus J 1975 On Bayesian methods for seeking the extremum *Optimization Techniques IFIP Technical Conference* (Springer) pp 400–4
[21] Garud S, Karimi I and Kraft M 2016 *Comput. Chem. Eng.* **96** 103–14
[22] Zhou J, Su X and Cui G 2018 *J. Contaminant Hydrol.* **216** 50–57
[23] Zhang J, Zheng Q, Chen D, Wu L and Zeng L 2020 *Water Resour. Res.* **56** e2019WR025721
[24] Gong W and Duan Q 2017 *Environ. Modelling Softw.* **95** 61–75
[25] Ginting V, Pereira F, Presho M and Wo S 2011 *Comput. Geosci.* **15** 691
[26] Solonen A, Ollinaho P, Laine M, Haario H, Tamminen J and Jarvinen H 2012 *Bayesian Anal.* **7** 715–36
[27] Mánek P and Van Goffrier G 2020 UCL TBR group project (available at: https://github.com/ucl-tbr-group-project)
[28] Mánek P and Van Goffrier G 2020 Surrogate modelling of the tritium breeding ratio CDT DIS note Department of Physics and Astronomy, University College London