Science and Technology Facilities Council

UK Atomic Energy Authority

Hartree Centre

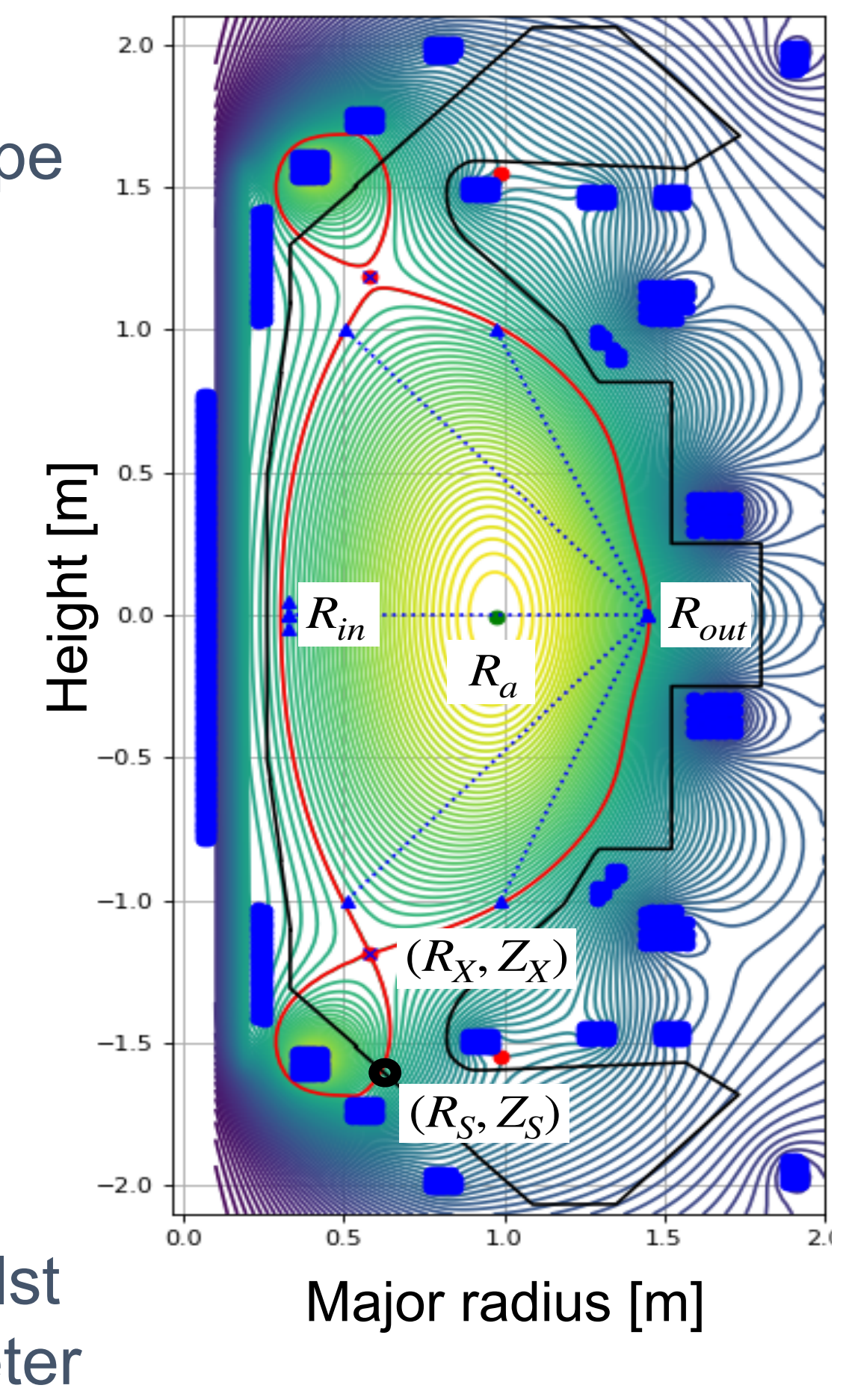
hartree.stfc.ac.uk

Efficient generation of synthetic datasets for magnetic confinement fusion

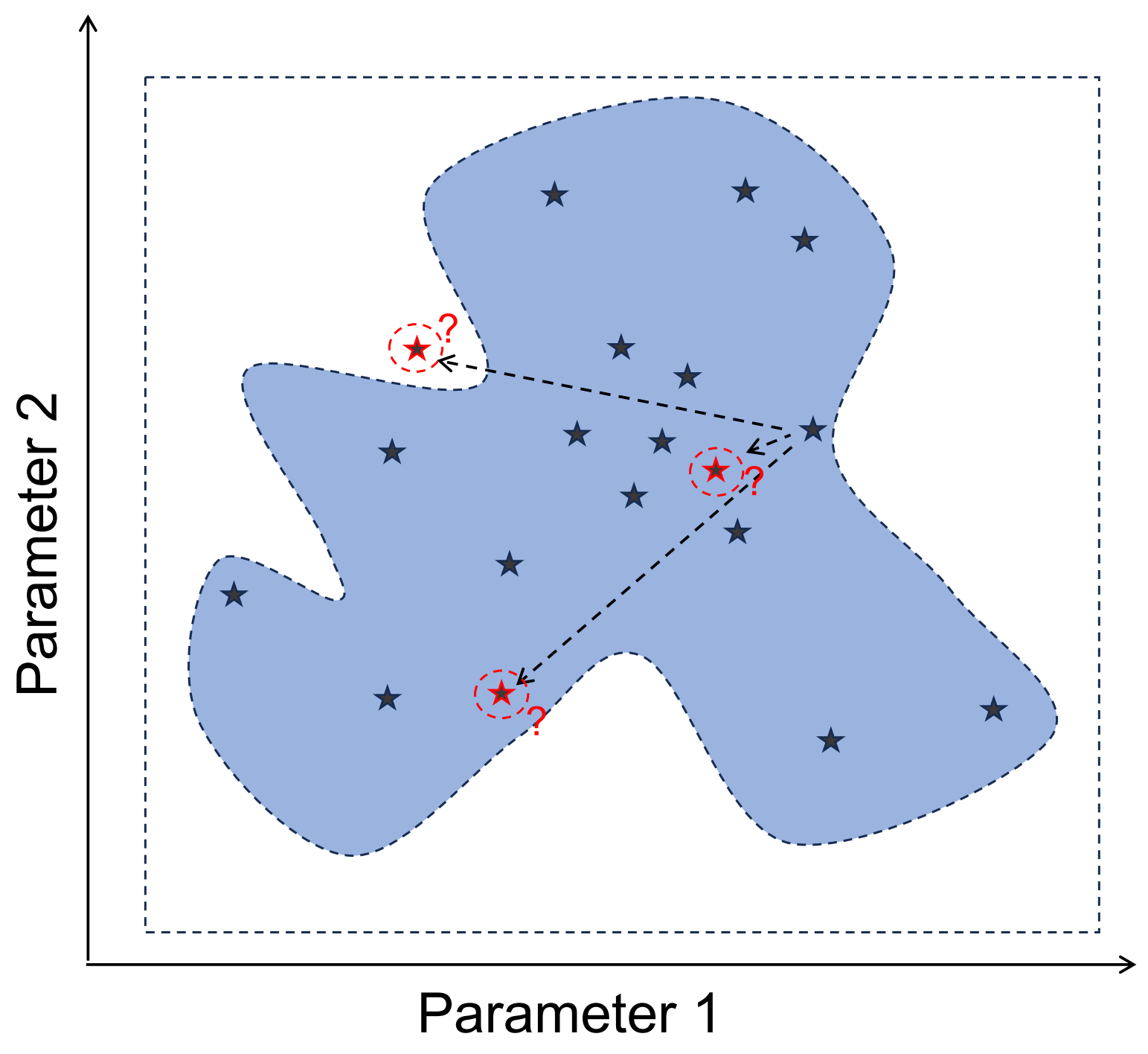
Abbie Keats[1], George K. Holt[1], Adriano Agnello[1], Nicola C. Amorisco[1], Dominic Richards[1], Stanislas Pamela[2], James Buchanan[2]
[1]STFC Hartree Centre, UK, [2]UK Atomic Energy Authority, UK
abbie.keats@stfc.ac.uk

Abstract We present three case studies demonstrating the minimisation of human intervention from the process of generating datasets with relevance to different problems in tokamak fusion experiment design and control: 1) Markov Chain Monte Carlo algorithm for generating a library of plasma equilibrium configurations; 2) the efficient generation of large SD1D and Hermes-3 datasets with strategies for deployment on HPC systems that minimise resource requirements; 3) ranked batch-mode active learning for the efficient training of machine learning models.

Monte Carlo methods for synthetic training sets



- **Task:** Generate a large synthetic library of plasma equilibrium configurations using FreeGSNKE [1], to be used for plasma shape emulation and control
  - Inputs: coil currents and plasma profiles
  - Outputs:
    - Position of magnetic axis (R_a, Z_a)
    - Position of inner and outer separatrix (R_{in}, R_{out})
    - Lower-outer strike point (R_s, Z_s)
    - X-point (R_X, Z_X)
    - Connection length, safety factor (q), internal inductance, plasma betas, etc.
- **Challenge:** to explore the operational space of the tokamak whilst remaining within an unknown non-trivial boundary of the parameter space, where:
  1. Grad-Shafranov solver is convergent
  2. Coil currents are within operational limits
  3. Equilibrium configurations are desirable, e.g. divertor detachment, longer divertor legs, $q_{\min} > 1$ on axis etc. , which is quantified by a score function, $\mathcal{L}(eq)$.



- **Solution:** a purpose-built Markov Chain Monte Carlo algorithm is used to sample new equilibrium configurations with an acceptance probability at step $n+1$ of $p(eq(n+1)|eq(n)) = \min\left(1, \frac{\mathcal{L}(eq(n+1))}{\mathcal{L}(eq(n))}\right)$.

Currently applied to shape emulation towards real-time control on MAST-U [2].

Dataset of 1D SOL simulations

- SD1D and Hermes-3 are plasma fluid simulators built on BOUT++ [3,4].
- They solve a time-dependent isotropic plasma fluid model for density, pressure and momentum of both the ions and neutrals.
- SD1D is a single-component 1D plasma model. Hermes-3 extends the SD1D model to include multi-dimensions, and multi-ion and impurity components in the plasma.
- Used for the study of tokamak scrape-off layer (SOL) and divertor detachment [3,4].

SD1D dataset generation

- SD1D dataset of 10,000 samples is generated in a subset of the MAST-U parameter space.
- Uniform random sampling.

Fixed/varying inputs

Parameter	SD1D	Hermes-3
Connection length	25 m	[15,30] m
Flux tube area expansion	1	[1.25,2.5]
Recycling fraction	0.95	$[0.95, 0.9999]$
Impurity species	C	Ne
Upstream density	$[1, 10]\times10^{19}\mathrm{m}^{-3}$	$[0.2,1.2]\times10^{19}\mathrm{m}^{-3}$
Impurity fraction	$[0, 0.05]$ %	$[0, 1.5]\%$
Upstream power flux	$[1, 10]\times10^{7}\mathrm{W/m}^2$	$[0.22, 1.08]\times10^{8}$ $\mathrm{W/m}^2$

Hermes-3 dataset generation

Outputs

Parameter	Range
Radiation front position	$[10, 30]$ m
Electron temperature at divertor target	$[10^{-1}, 10^{2}]$ eV

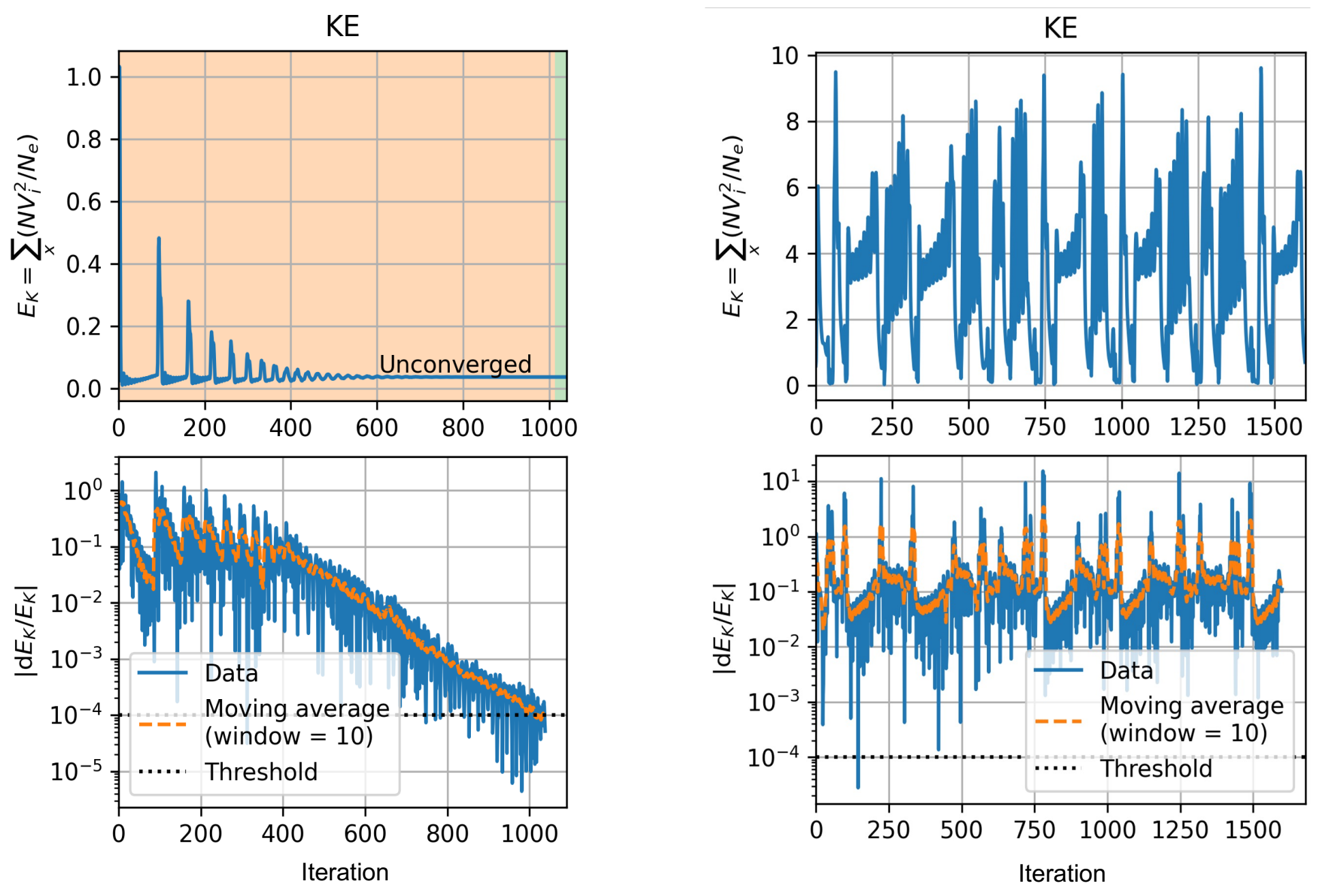
- Initial course sampling of input space $\mathbf{X}$ using Sobol sequencing to generate $\mathbf{X}_0$.
- Run simulations with inputs $\mathbf{X}_0$ to generate dataset D_0.
- Next, finely random sample input space to generate $\mathbf{X}_1$.
- Launch simulations with inputs $\mathbf{X}_1$ from the final state of the nearest converged simulation in $\mathbf{X}_0$ to generate D_1.
- This method reduces compute time of dataset generation by a factor ~3

References

[1] N. C. Amorisco et al. (2023*), Physics of Plasmas*, submitted, “FreeGSNKE: a Python-based dynamic free-boundary toroidal plasma equilibrium solver”.
[2] A. Agnello et al. (2023), *Physics of Plasmas*, submitted, “Emulation for Scenario Design and Classical Control of Tokamak Plasmas”.
[3] B. Dudson et al. (2018), *IOP,* “The role of particle, energy and momentum losses in 1D simulations of divertor detachment”.
[4] B. Dudson et al. (2023), “Hermes-3: Multi-component plasma simulations with BOUT++”
[5] B. Zhu, et al. (2022), *Journal of Plasma Physics,* “Data-driven model for divertor plasma detachment prediction”.
[6] S. Dasbach and S. Wiesen (2023), *Nuclear Materials and Energy*, “Towards fast surrogate models for interpolation of tokamak edge plasmas”.
[7] Cardoso et al. (2017), *Information Sciences,* “Ranked batch-mode active learning”.
[8] G. K. Holt, A. Keats et al. (2023), *Physics of Plasmas*, submitted, “Divertor Detachment Emulation”.

Online convergence checking

- BOUT++ input scripts specify number of time steps to simulate.
- Often, steady-state convergence is reached before this.
- Avoid wasting compute by checking status of simulation convergence while running and gracefully exiting once converged.

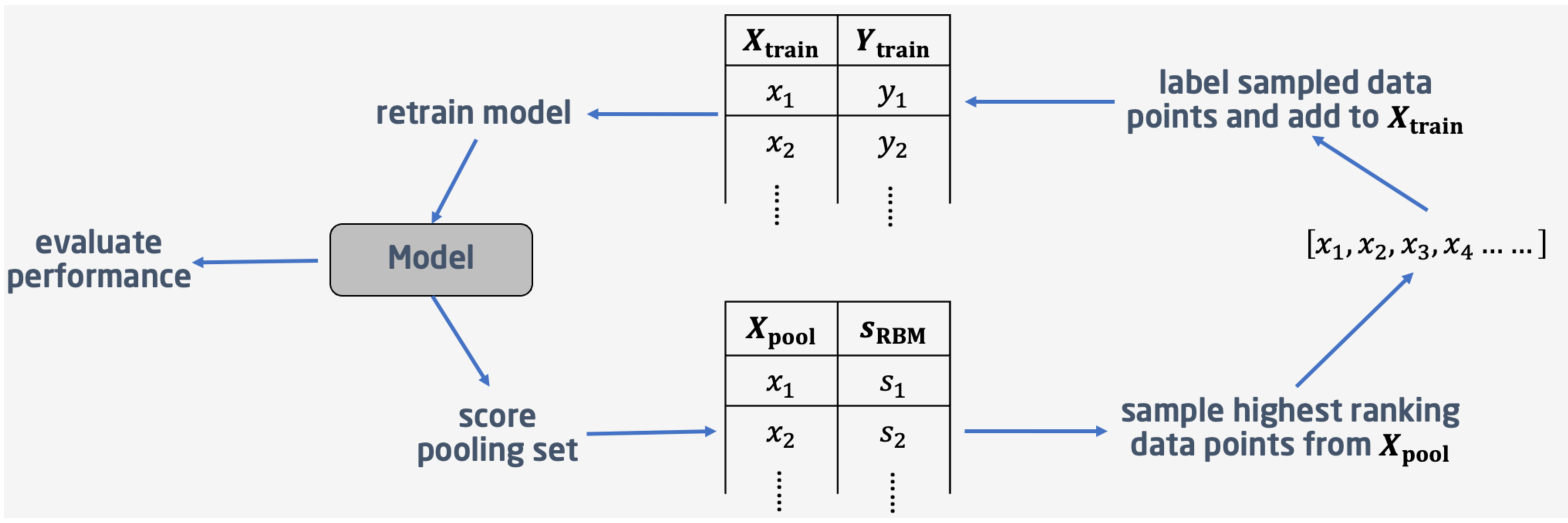


Kinetic energy evolution of a converging (left) and non-converging (right) simulation.

Active learning

- Need to build a fast surrogate model from SD1D dataset for reinforcement learning and optimal design studies [5,6].
- SD1D dataset is computationally expensive to generate.
- To meet computational demands, implement active learning: a method of selecting data that prioritizes the efficient training of ML models.

Ranked batch-mode (RBM) sampling method

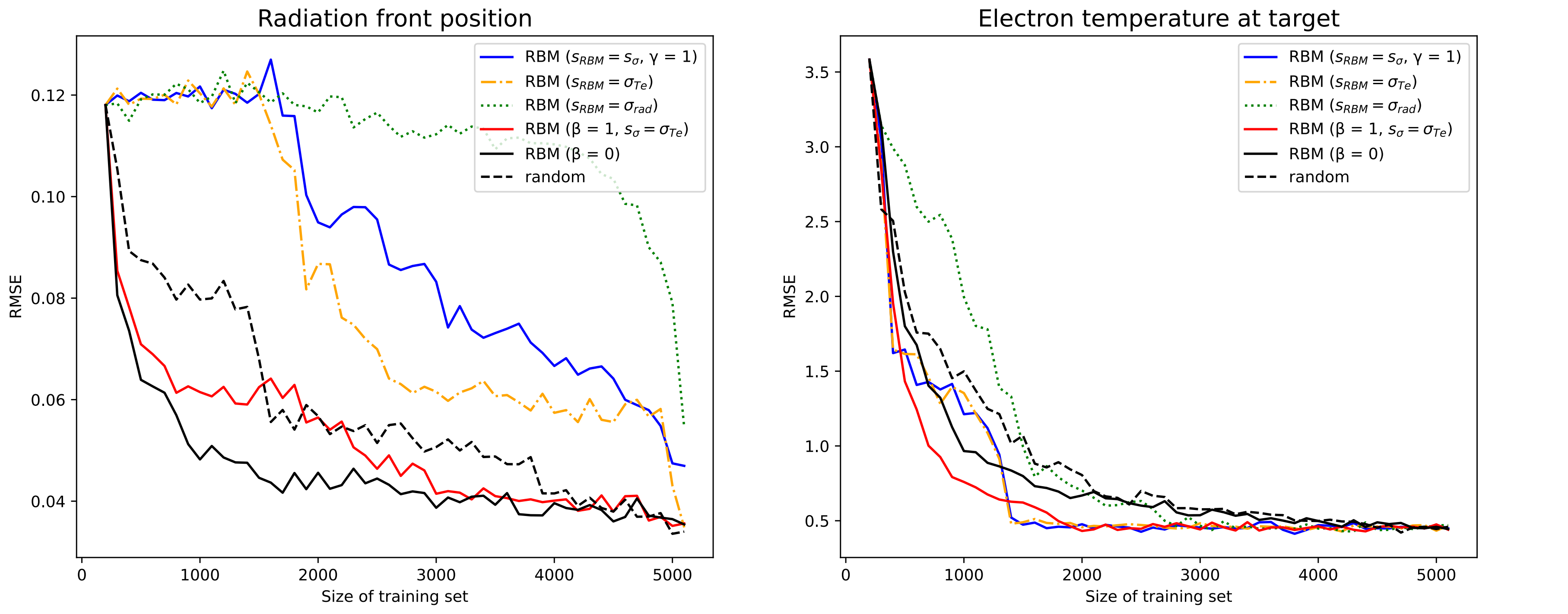

$\boldsymbol{X}_{\text{train}}$
$\boldsymbol{Y}_{\text{train}}$
x_1
y_1
x_2
y_2
retrain model
label sampled data points and add to $\boldsymbol{X}_{\text{train}}$
evaluate performance
Model
$[x_1, x_2, x_3, x_4 \dots \dots]$
$\boldsymbol{X}_{\text{pool}}$
$\boldsymbol{s}_{\text{RBM}}$
x_1
s_1
x_2
s_2
score pooling set
sample highest ranking data points from $\boldsymbol{X}_{\text{pool}}$

Ranked batch-mode score

- A ranked batch mode score ($\boldsymbol{s}_{\mathbf{RBM}}$) is assigned to data points from an unlabeled (pooling) dataset to evaluate how impactful a given data point is likely to be for training a model [7,8]:

$$\boldsymbol{s}_{\mathbf{RBM}} = \boldsymbol{\alpha}(\mathbf{1} - \boldsymbol{s}_{\mathbf{sim}}) + \boldsymbol{\beta}(\mathbf{1} - \boldsymbol{\alpha})\boldsymbol{s}_{\boldsymbol{\sigma}}.$$

- $\alpha = \frac{|X_{\text{pool}}|}{|X_{\text{train}} + X_{\text{pool}}|}$
- similarity score: $s_{\text{sim}} = \frac{1}{d(X_{\text{pool}}, X_{\text{train}})}$
  - $d(X_{\text{pool}}, X_{\text{train}})$ is the pairwise distance
- uncertainty score: $s_\sigma = \sigma_{\text{rad}} + \gamma\sigma_{\text{Te}}$
  - s_σ is evaluated using predicted outputs from an ensemble of models, with the same optimal hyperparameters (evaluated for the entire dataset) but varied learning rate
  - $\sigma_{\text{rad/Te}}$ is the standard deviation of the predicted radiation front position/electron temperature at target from each ensemble of models
- β, γ: constant weighting factor



SD1D results: RMSE of radiation front position (left) and electron temperature at divertor target (right) against size of training set, for different RBM sampling parameters and random sampling (black dashed lines). The results given by the blue, yellow and green lines are for a RBM score that only depends on the uncertainty score, the black line is for a RBM score that only depends on the similarity score, and the red line is for a RBM score that depends on both uncertainty and similarity scores [8].