Active and continual learning of fusion plasma turbulence surrogate models for digital twinning of a tokamak device

Jackson Barr JACKSON.BARR.17@UCL.AC.UK
Thandikire Madula THANDI.MADULA.17@UCL.AC.UK
Lorenzo Zanisi LORENZO.ZANISI@UKAEA.UK
Vignesh Gopakumar VIGNESH.GOPAKUMAR@UKAEA.UK
Aaron Ho A.HO@DIFFER.NL
Jonathan Citrin J.CITRIN@DIFFER.NL
and JET contributors *

Abstract

Digital twinning of a dynamic system requires fast system state inference. Physics-based computational models that predict future states are often too slow to be actionable, and thus undesirable for offline scenario planning. These tasks may be performed faster if the physics-based model is replaced by a neural network-based surrogate. Obtaining the labels to train the surrogate can be computationally expensive, additionally, some inputs may either be invalid for the range of applicability of the model or result in trivial outputs. Lastly, it is sometimes necessary to generalise the performance of the surrogate across newly deployed systems and the relative extended parameter space. Active and continual learning may be combined to address the need to label only the most relevant experimental inputs for a given system, while retaining knowledge of previous systems and states. Here we propose an active-and-continual learning pipeline for digital twinning of turbulence in the core of tokamak fusion plasmas. Our pipeline leverages an uncertainty-based acquisition function which greatly outperforms random acquisition. We take inspiration from simple continual learning methods found in literature and find that a surrogate can generalise well over tokamak configurations as well as plasma confinement modes. Overall, our work motivates further research in active and continual learning for regression tasks.

1. Introduction

Turbulent transport in Tokamak plasmas is a major roadblock to achieving fusion power (Doyle et al., 2007). It is thus crucial to forecast expected turbulent fluxes in the operating parameter space of a given tokamak for design, validation and optimization at the pre- and post-experiment stage. However, calculating these fluxes using fully non-linear models is very computationally expensive.

In general, although constrained by empirical and theory-based extrapolations, the range of plasma states achieved by a machine is strictly unknown at the outset. Therefore, a general-purpose surrogate would be expensive to obtain, as the computational effort needed to obtain the labels from all potentially needed realisations of the physical model would be

∗. See the author list of 'Overview of JET results for optimising ITER operation' by J. Mailloux et al. published in Nuclear Fusion Special issue: Overview and Summary Papers from the 28th Fusion Energy Conference (Nice, France, 10-15 May 2021)

prohibitively long. Instead, a surrogate can be trained directly on experimental inputs, which better capture the parameter subspace of the machine. The surrogate can then be re-evaluated and updated when new data becomes available.

Experimental data from a given machine may be redundant, and data labelling for similar inputs would be inefficient. Moreover, by the nature of the critical threshold characteristic of tokamak turbulence and measurement uncertainties, not all input plasma states result in unstable modes. Thus, a significant proportion of the computational budget to obtain the turbulent fluxes is spent on stable inputs, which can be wasteful. Active learning may be used to select the inputs that would be most useful to update the surrogate.

Furthermore, plasma discharges may result in two distinct modes of confinement, high-confinement (H) and low-confinement (L) mode, which cover distinct plasma state subspaces with respect to the input parameters of core turbulent transport codes. Future experiments and power plants, such as the International Thermonuclear Experimental Reactor (ITER) and the Spherical Tokamak for Energy Production (STEP), will start operations in the suboptimal L-mode with the final aim of producing stable, high-performing H-mode plasmas. Expanding the regime of applicability of a transport surrogate to H-mode and to new machines with minimal new data points will thus be paramount. This real-world example of continual learning under distribution shift has far-reaching implications for the future of energy generation.

Active and continual learning for classification have received much attention in the machine learning community (Ash et al., 2020; Kirsch et al., 2019; Fang et al., 2017; Kirkpatrick et al., 2017; Farajtabar et al., 2020; Schwarz et al., 2018). However, applications to regression problems remain scarce (with a few recent exceptions, e.g., Ash et al. 2021; He and Sick 2021). In this paper, we combine active and continual learning under distribution shift to build a surrogate of plasma core turbulent transport based on the QuaLiKiz model (Bourdelle et al., 2015; Citrin et al., 2017). Firstly, we propose a two-stage learning paradigm that exploits the binary nature of plasma core instabilities. Our framework uses a small initially labelled dataset to pre-train a stability classifier, and a regressor to predict the related turbulent fluxes. The initial dataset is then augmented by acquiring the labels of candidate inputs that are classified as likely to result in a growing instability, and would most improve the range of validity of the surrogate. Secondly, we apply this learning paradigm in a simulated continual learning scenario, where the applicability of a surrogate is extended from L-mode to H-mode plasmas, as well as across machine configurations, as a proxy for learning across tokamak machines.

2. Data, methods and related work

2.1 Data

We use an experimentally-based dataset from the JET (Joint European Torus) tokamak, which contains the turbulent transport calculation inputs and related QuaLiKiz outputs needed to make a surrogate model. This dataset was originally used in Ho et al. (2021) to train neural network ensembles, without active or continual learning.

The dataset includes both data for the H- and L- modes of confinement, as well as data spanning from the original carbon wall (C-wall) configuration to the current ITER-like beryllium wall (ILW) with tungsten divertor plates. See Ho et al. (2021) for a description

of the 15-dimensional input space. Here we focus on predicting the ion heat flux of the Ion Temperature Gradient turbulence (Horton, 1999), $q_{i,ITG}$ in the GyroBohm units [GB], for which only approximately 25% of inputs will develop turbulent transport. The methods employed in this work can be extend to multiple outputs as explored in Appendix A.

2.2 Methods

Active Learning Given an unlabelled pool of experimental inputs, a classifier and a regressor are pretrained on a small random sample of data whose labels were obtained by running QuaLiKiz in Ho et al. (2021). Hereafter, the networks and the labelled dataset are updated following a two-stage active learning pipeline shown in Figure 1. The classifier is tasked with screening a sample of candidate points of size `CandidateSize` from the unlabelled pool. `CandidateSize` is chosen to be 10,000, which reflects the number of distinct inputs obtained from a single JET plasma discharge. An acquisition function selects which of the candidates should be labelled, and those inputs are then appended to the training data, to retrain the regressor. Specifically, we select `TopUncertain`% (set to 25%) of the inputs with the highest regressor variance. The points that were wrongly classified as unstable are stored in a buffer of size `ClassifierBuffer` (set to 200) which, when filled, is used to retrain the classifier along with the previous training data. Both the regressor and the classifiers are equipped with Dropout layers (Srivastava et al., 2014). We use Monte Carlo Dropout (Gal and Ghahramani, 2016) to estimate the uncertainty of the NN regressor. Although the classifier achieved an actionable performance in the pre-trained phase, more advanced acquisition functions based on both the regressor and classifier uncertainties will need to be explored.

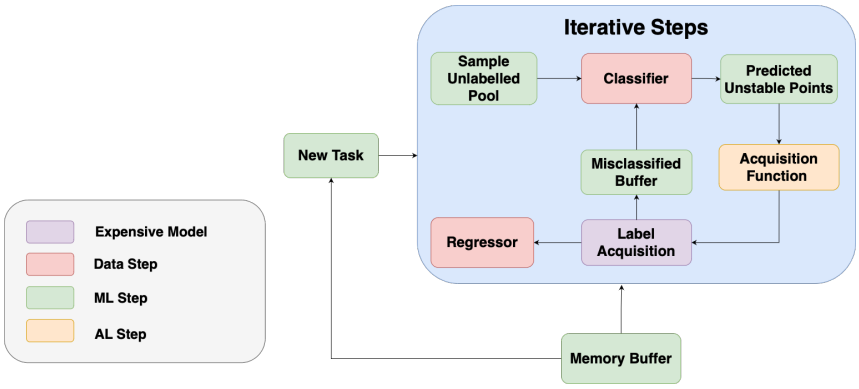


Figure 1: Schematic diagram of active learning framework. Each continual learning task consists of multiple iterations of the steps in the blue box.

Continual Learning It has been shown that retaining some data from previous tasks (i.e., a 'memory buffer') in the training set is beneficial for reducing forgetting (Rolnick et al., 2019). We generalise across tasks sequentially by retaining a fraction α of the training set at the end of each task. It has also been shown that warm starting neural networks by applying a 'shrink-perturb-repeat' technique may improve generalisation across tasks (Ash and Adams, 2020). Model weights, θ, are updated at the start of each new task using the

update rule $\theta_i^t = \lambda\theta_i^{t-1} + p^t$ where $p^t \sim \mathcal{N}(0, \sigma^2)$ and $0 < \lambda < 1$. The case where no shrink-perturb is used is labelled as $\lambda = 1$ in the following. We combine both memory replay and the shrink-perturb technique in our experiments. The surrogate trained at task N is evaluated on holdout data that also includes the entire test sets of the previous tasks. We compute the average forgetting after the N^{th} task as

$$AvgForgetting_N = \frac{1}{N-1}\sum_{i}^{N-1}\Big(MSE(task_i, model_N) - MSE(task_i, model_i)\Big)\,, \qquad (1)$$

where $model_i$ represents the model trained for the i^{th} task in the sequential learning pipeline. The quantity in eq. 1 is expected to be negative, as performance on the updated model on either of the N-1 previous only is expected to somewhat degrade.

3. Experiments

Active Learning We conducted various experiments to investigate the effectiveness of an uncertainty-based acquisition function and the effect of the stability classifier on the pipeline performance. An initial training dataset size of 5,000 was used. Classifier performance across iterations was observed to be relatively unchanged, achieving a constant F1 score of 87%.

Two acquisition methods were evaluated using the trained models' MSE on the holdout set. In addition to the baseline random acquisition, we investigated the use of sampling based on the regressor's uncertainty. As shown in Figure 2, uncertainty-based sampling provides significantly improved performance compared to random sampling. Figure 2 shows that the combination of the classifier and the use of uncertainty-based acquisition provides a much greater decrease in MSE than using either individually. As only 25% of inputs result in a growing turbulence, only a minority of data informs the regressor about the unstable region and as such the classifier stage of the pipeline proves very beneficial. The classifier aids the regressor in promoting a diversity of inputs.

Table 1 summarises the final test loss after 25 pipeline iterations. The combined classifier and uncertainty acquisition model significantly outperforms a regressor trained on the entire dataset and has comparable performance to a model trained using the subset of the data that is unstable. This is done using 16,000 training data points which represent only 0.1% (0.4%) of the full (unstable only) dataset. Appendix C shows results for experiments conducted using various initial training set sizes that are then evaluated at the same final training set size. It was found that initially using 5,000 points had a better final performance than when starting with 1,000 points.

We note that the MSE is correlated with, but not necessarily fully reflective of, the surrogate quality in the final tokamak modelling application. Subtle features of the input-output mapping must also be captured, as shown in van de Plassche et al. (2020). Future work will include more domain-specific metrics.

Active-and-Continual Learning For the continual learning task we learn the heat flux of ions for the ITG turbulence sequentially from the JET C-wall (first in L-mode and then in H-mode) and then similarly for the JET-ILW configuration.

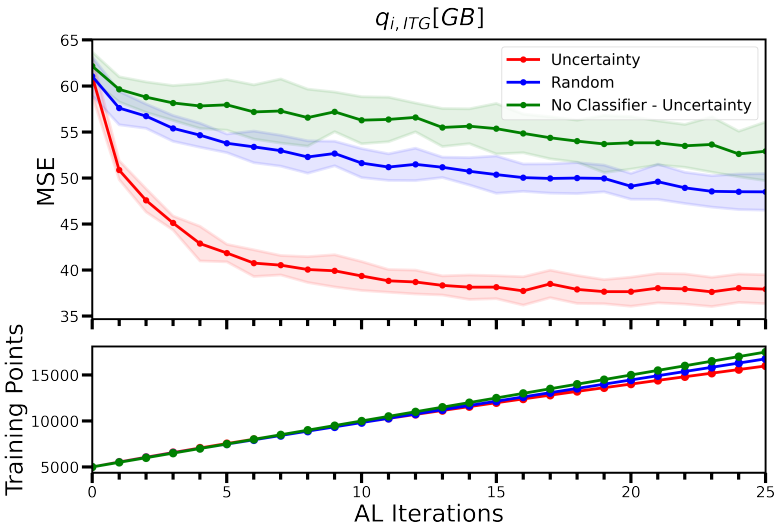


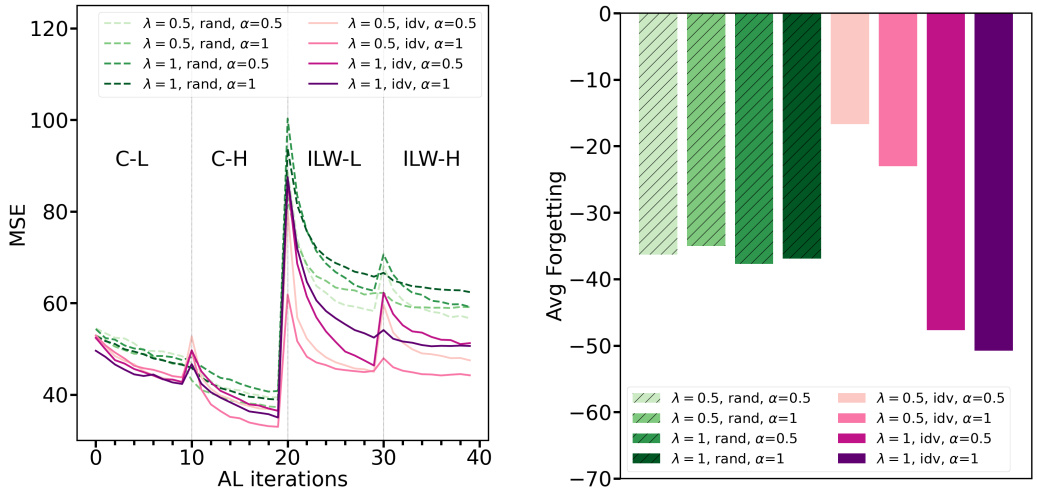
Figure 2: Test losses for $q_{i,ITG}[GB]$ using different acquisition methods. Bottom plots show the number of points sampled from the unlabelled pool at each iteration. An initial training dataset of 5 K points was used. The lines and shaded areas are the means and one standard deviation of 10 random realisations.

Data Acquisition Method	$q_{i,ITG}[GB]$ MSE
No Classifier - Uncertainty	52.9 ± 3.2
Classifier - Random	48.5 ± 2.0
Classifier - Uncertainty	$\mathbf{37.9 \pm 1.6}$
Full Dataset	50.8 ± 0.6
Full Dataset - Unstable Values Only	38.6 ± 0.5

Table 1: The average test MSE and one standard deviation after 25 active learning pipeline iterations trained on the ion temperature gradient flux. This is compared to a model trained using the entire dataset (17M points) and using turbulent subset of points (4M points).

We ran a series of experiments where we combine different memory buffer sizes, α, shrink-perturb hyperparameters, λ, and sampling. Specifically, we use all permutations of the following: $\alpha = \{0.5, 1\}$, $\lambda = \{0.5,1\}$, `acquisition`={random, uncertainty}. The test MSE for these experiments are shown in Figure 3a.

We find that random acquisition also performs worse in the continual learning setting. A much larger increase in MSE is observed when switching from the C-walls to the ILW configuration than when switching from L-mode to H-mode. This is expected as the distribution shift is larger for different tokamak walls compared to different confinement modes. Furthermore, the performance of the uncertainty-based acquisition is improved by using a shrink-perturb λ of 0.5 and a small memory buffer. This result is quantified by the better average forgetting shown in Figure 3b.



(a) (b)

Figure 3: Left: the MSE test loss for the continual learning pipeline experiments. The test MSE is reported for test data including the current and previous tasks. Lines are the means of 10 random realisations (variance is not shown for clarity). In the legends, rand refers to random acquisition and idv to the uncertainty acquisition. Right: the performance of the experiments in terms of average forgetting (higher is better).

4. Conclusion

This paper investigates the use of a two-stage active learning pipeline for the modelling of turbulent transport in tokamak reactors that may be extended to other digital twinning applications. The inclusion of a classifier stage to identify regions of the input space that lead to growing turbulence modes greatly improves the surrogate model performance. Additionally, when used in combination with an uncertainty-based acquisition function the performance of the pipeline matches that of full-dataset training but with only 0.4% of the data, which would have resulted in a significant reduction in computational time spent generating labels. This acquisition strategy was then deployed in a continual learning setting, wherein data pertaining to different machine conditions and plasma confinement states are learned sequentially by the surrogate. A shrink-perturb trick and a limited memory buffer of previous tasks provides less forgetting. However, the performance of the surrogate in the continual learning scenario for a large distribution shift is still suboptimal. Our work shows the potential of active and continual learning for regression tasks and calls for more research on these topics.

Acknowledgments

We would like to thank S. Pamela, B. Joachimi and A. Spurio Mancini for useful comments.

References

Jordan T. Ash and Ryan P. Adams. On warm-starting neural network training. *ArXiv: Learning*, 2020.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *ArXiv*, abs/1906.03671, 2020.

Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham M. Kakade. Gone fishing: Neural active learning with fisher embeddings. In *NeurIPS*, 2021.

C Bourdelle, J Citrin, B Baiocchi, A Casati, P Cottier, X Garbet, F Imbeaux, and JET Contributors. Core turbulent transport in tokamak plasmas: bridging theory and experiment with QuaLiKiz. *Plasma Physics and Controlled Fusion*, 58(1):014036, 2015.

J. Citrin, C. Bourdelle, F.J. Casson, C. Angioni, N. Bonanomi, Y. Camenen, X. Garbet, L. Garzotti, T. Görler, O. Gürcan, F. Koechl, F. Imbeaux, O. Linder, K.L. van de Plassche, P. Strand, and G. Szepesi JET contributors. Tractable flux-driven temperature, density, and rotation profile evolution with the quasilinear gyrokinetic transport model QuaLiKiz. *Plasma Physics and Controlled Fusion*, 59(12):124005, 2017.

EJ Doyle, WA Houlberg, Y Kamada, V Mukhovatov, TH Osborne, A Polevoi, G Bateman, JW Connor, JG Cordey, T Fujita, et al. Plasma confinement and transport. *Nuclear Fusion*, 47(6):S18, 2007.

Meng Fang, Yuan Li, and Trevor Cohn. Learning how to active learn: A deep reinforcement learning approach. In *EMNLP*, 2017.

Mehrdad Farajtabar, Navid Azizan, Alexander Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *AISTATS*, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

Yujiang He and Bernhard Sick. Clear: An adaptive continual learning framework for regression tasks. *AI Perspectives*, 3(1):1–16, 2021.

A. Ho, Jonathan Citrin, C. Bourdelle, Y. Camenen, Francis J Casson, K. L. van de Plassche, and H. Weisen. Neural network surrogate of qualikiz using jet experimental data to populate training space. *Physics of Plasmas*, 28:032305, 2021.

W Horton. Drift waves and transport. *Reviews of Modern Physics*, 71(3):735, 1999.

James Kirkpatrick, Razvan Pascanu, Neil C. Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114:3521 – 3526, 2017.

Andreas Kirsch, Joost R. van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *NeurIPS*, 2019.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. In *NeurIPS*, 2019.

Jonathan Schwarz, Wojciech Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pages 4528–4537. PMLR, 2018.

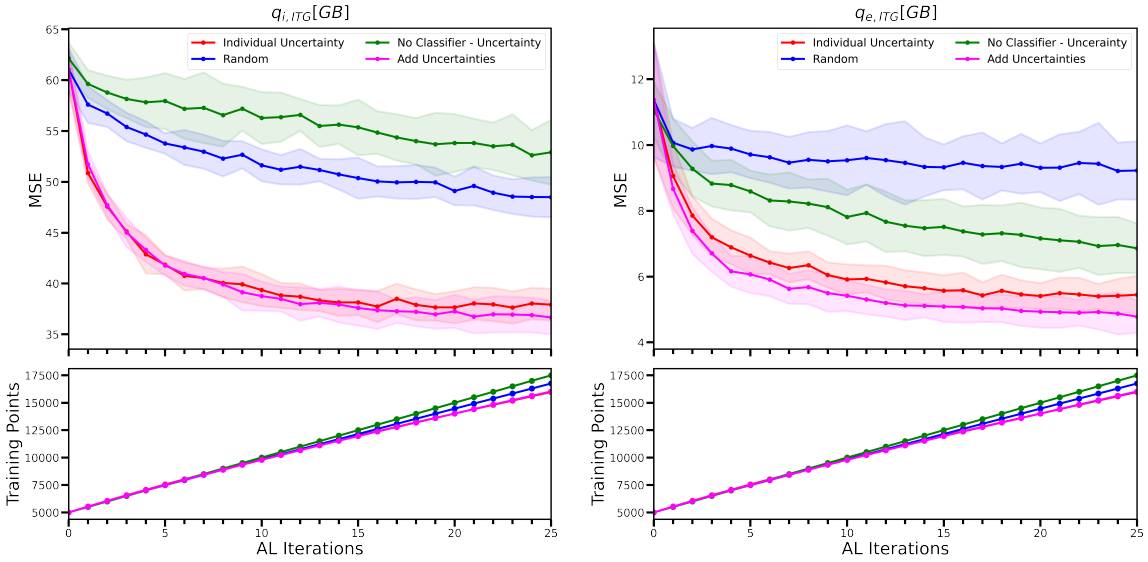
Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014.

Karel Lucas van de Plassche, Jonathan Citrin, Clarisse Bourdelle, Yann Camenen, Francis J Casson, Victor I Dagnelie, Federico Felici, Aaron Ho, Simon Van Mulders, and JET Contributors. Fast modeling of turbulent transport in fusion plasmas using neural networks. *Physics of Plasmas*, 27(2):022310, 2020.

Appendix A: Experiments with Multiple Fluxes

In this section we present some further experiments where the pipeline has been extended to include multiple fluxes, in a multi agent set-up. In addition to the ion heat flux, a surrogate model to predict the electron heat flux, $q_{e,ITG}$ $[GB]$ is also constructed. When considering multiple fluxes, how to best sample points that will best improve the performance of all regressors need to be considered. The simplest approach is to train each model independently but this approach can be wasteful and require the labelling of more data than if the models are trained concurrently. In this work, we sample points for which the sum of the uncertainties of individual fluxes was greatest. We use `TopUncertain`=25% as in the main text.

Figure 4 shows the performance across the two investigated fluxes. It is observed that combining the uncertainties provides competitive performance to training each flux independently. Table 2 summarises the pipeline performance after 25 iterations.



(a) (b)

Figure 4: Test losses for $q_{i,ITG}[GB]$ (top left) and $q_{e,ITG}[GB]$ (top right) fluxes using different acquisition methods. Bottom plots show the number of points sampled from the unlabelled pool at each iteration. An initial training dataset of 5K points was used.

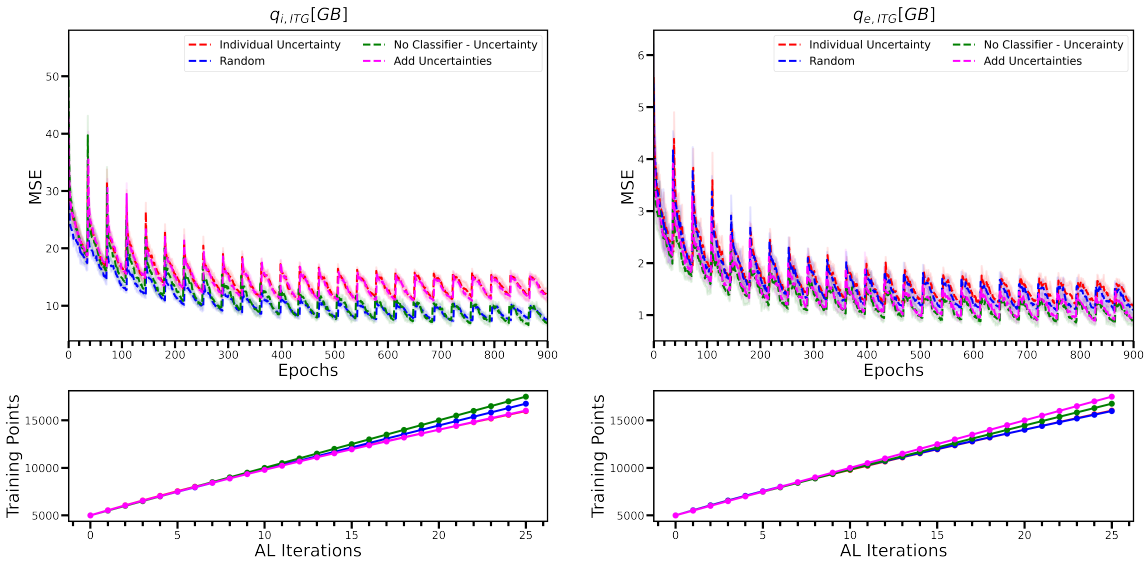
Data Acquisition Method	$q_{i,ITG}[GB]$ MSE	$q_{e,ITG}[GB]$ MSE
No Classifier - Uncertainty Acquisition	52.9 ± 3.2	6.9 ± 0.8
Classifier - Random Sampling	48.5 ± 2.0	9.2 ± 0.9
Classifier - Individual Uncertainty	37.9 ± 1.6	5.4 ± 0.6
Classifier - Sum of Uncertainties	$\mathbf{36.7 \pm 1.7}$	$\mathbf{4.8 \pm 0.5}$
Full Dataset	50.8 ± 0.6	6.0 ± 0.5
Full Dataset - Unstable Values Only	38.6 ± 0.5	6.0 ± 0.5

Table 2: The average test MSE and one standard deviation after 25 iterations of the active learning pipeline trained concurrently on two different fluxes. This is compared to training a model using the entire dataset (17M points) and using only the subset of points that are turbulent (4M points).

Appendix B: Architecture and Training Loss Comparisons

The architecture for the classifier and regressors consists of a simple feed-forward neural network consisting of an input size of 15 and 5 hidden layers with sizes [128, 256, 512, 256, 128]. A dropout rate of 0.1 was employed in each layer along with ReLU activation functions. We used the `Adam` optimiser with a learning rate of 0.001, and a weight decay of 0.0001.

Figure 5 shows the training losses for the different acquisition functions that were explored (including the electron heat flux discussed in Appendix A). It is observed that random sampling with the classifier and uncertainty sampling without the classifier both achieve a lower training loss than using both uncertainty sampling and the classifier. This appears to show that the combination of both methods assists the regressor in avoiding local minima that do not generalise well to the unseen data. The amplitude of the oscillations at each new iteration of the pipeline decrease with further iterations.



(a) (b)

Figure 5: Training losses for $q_{i,ITG}[GB]$ (top left) and $q_{e,ITG}[GB]$ (top right) fluxes using different acquisition methods. Bottom plots show the number of points sampled from the unlabelled pool at each iteration.

Appendix C: Initial Training Set Size

Table 3 compares the pipeline performance starting from a labelled pool 1 thousand data points and starting from 5 thousand but evaluated at the same final training set size. The 1,000 data points pipeline runs had a final training dataset size of 8,500 after 25 iterations, the 5,000 runs had an equivalent amount of data after 7 iterations. As seen in table 3, starting with a larger training set size gives better performance for a given final dataset size. Further studies with larger initial training dataset sizes need to be conducted but preliminary experiments (not included here for brevity) suggest that further increases in the initial training set size provide increasingly marginal gains.

When using a very small initial dataset size the surrogate is expected to be a poor approximation of the true model but this may also be true of the estimates of the regressor uncertainty. If the regressor uncertainty is not well calibrated to measure the ability to predict the true flux, then the new data sampled may be sub-optimal for improving regressor performance.

Data Acquisition Method	Initial Training Set Size	$q_{i,ITG}[GB]$ Unscaled MSE	$q_{e,ITG}[GB]$ Unscaled MSE
No Classifier - Uncertainty Acquisition	1K	64.3 ± 4.8	8.3 ± 1.3
	5K	57.3 ± 3.5	8.3 ± 0.9
Classifier - Random Sampling	1K	57.0 ± 4.3	11.5 ± 2.8
	5K	53.0 ± 1.7	9.5 ± 0.8
Classifier - Individual Uncertainty	1K	51.0 ± 4.4	6.9 ± 1.6
	5K	40.5 ± 1.0	6.3 ± 0.4
Classifier - Addition of Uncertainties	1K	48.6 ± 4.1	5.9 ± 1.5
	5K	40.5 ± 1.1	5.6 ± 0.5
Full Dataset	17M	50.8 ± 0.6	6.0 ± 0.5
Full Dataset - Unstable Values Only	4M	38.6 ± 0.5	6.0 ± 0.5

Table 3: Comparison of the acquisition methods for different initial starting training points. The 1K runs were performed for 25 iterations and the 5K runs were terminated once they reached the same training set size as the 1K runs.