Higher Fidelity Surrogate Models for Gyrokinetic Simulations

M. Lanzarone[1], Y. Camenen[1], G. Fuhr[1], F. Almuhisen[2], C. Bourdelle[2] , J. Citrin[3,4], K. L. van de Plassche[3,5]

[1]CNRS, Aix-Marseille Univ. , PIIM UMR7345, Marseille, France
[2]CEA, IRFM, F-13108 Saint-Paul-lez-Durance, France
[3]DIFFER: Dutch Institute for Fundamental Energy Research, Eindhoven, The Netherlands
[4]Science and Technology of Nuclear Fusion Group, Eindhoven University of Technology, Eindhoven, The Netherlands
[5]Ignition Computing, Keizersgracht 16D, 5611GD Eindhoven, The Netherlands

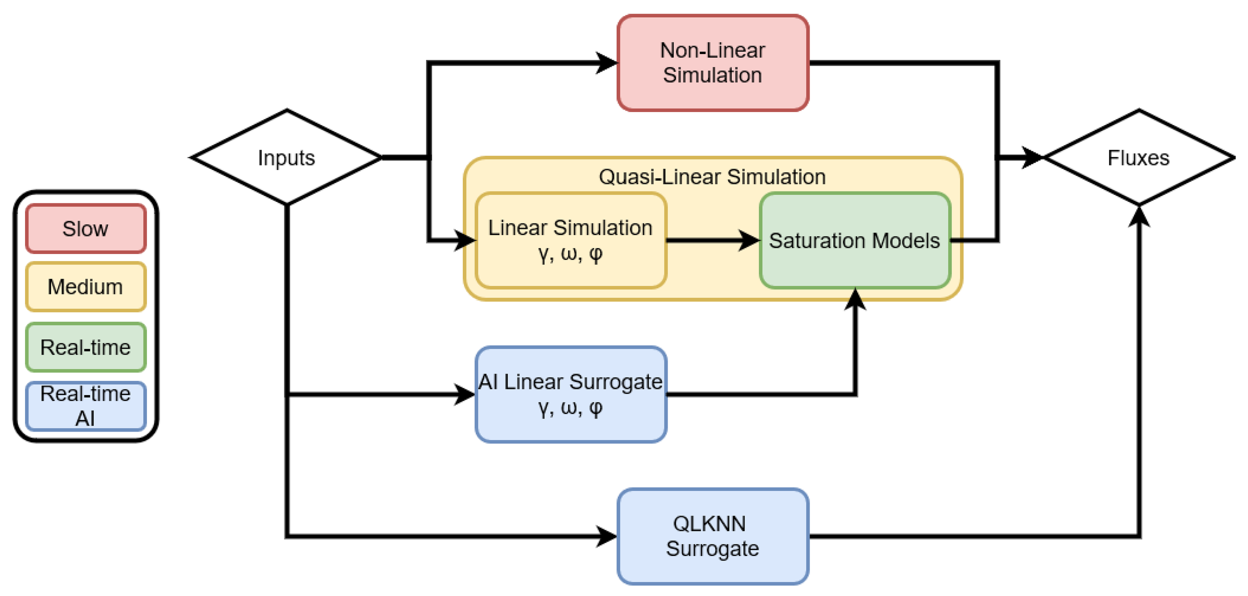
Introduction

Integrated Modelling

- Simulates plasma profile evolution and resulting transport on confinement timescales.
- Uses: Tokamak Design and ITER Scenario Simulations
  - Ideally need a high volume of accurate simulations.
- Bottlenecked by gyrokinetic simulations used to calculate fluxes which need to run ~10^4 times per second of plasma for integrated modelling

Gyrokinetic Models – Flux Tube δf

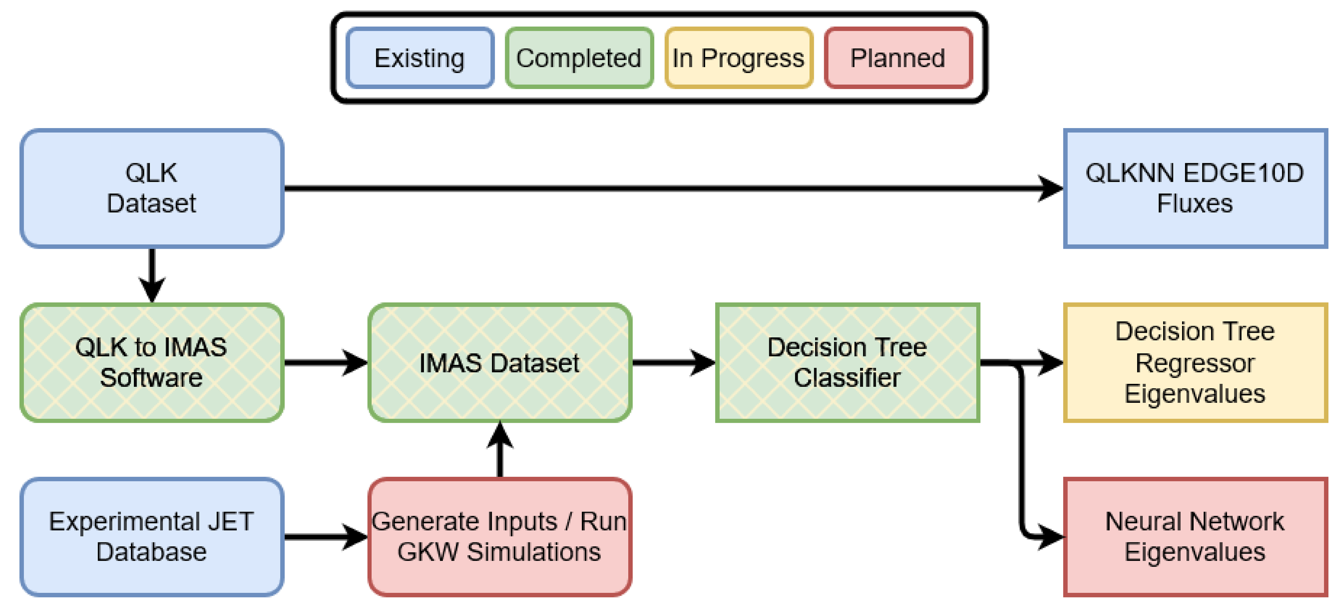
- Solve Fokker-Planck and Maxwell equations, integrating over the gyro-motion of particles to reduce dimensionality
- Inputs: gradients (Temperature, Density, Pressure, etc)
- Outputs: fluxes (particle, heat, momentum, etc)
- Even the fastest reduced models (quasi-linear) are too slow for real-time integrated modelling
- Can AI surrogate models trained on existing simulation results act as a reliable substitute for running new simulations?



FASTER Project

Improve on existing Gyrokinetic AI Surrogates [1,2]

- Train on higher fidelity Linear GKW [3] simulations instead of QuaLiKiz (QLK) [4]
  - GKW is electromagnetic as opposed to electrostatic
  - GKW uses arbitrary magnetic geometry as opposed to circular flux surfaces
  - However significantly increased runtime per simulation
  - Smaller training set (~100,000 vs 256M) with limited computation time
- Train on linear simulations to allow for experimentation of saturation models
- Train on Integrated Modelling and Analysis Suite (IMAS) [5] normalised data
- Test and compare different kinds of machine learning processes for stability classification and linear response regression
  - Decision Trees (XGBoost [6], this poster)
  - Neural Networks (NNs)
  - Gaussian Processes



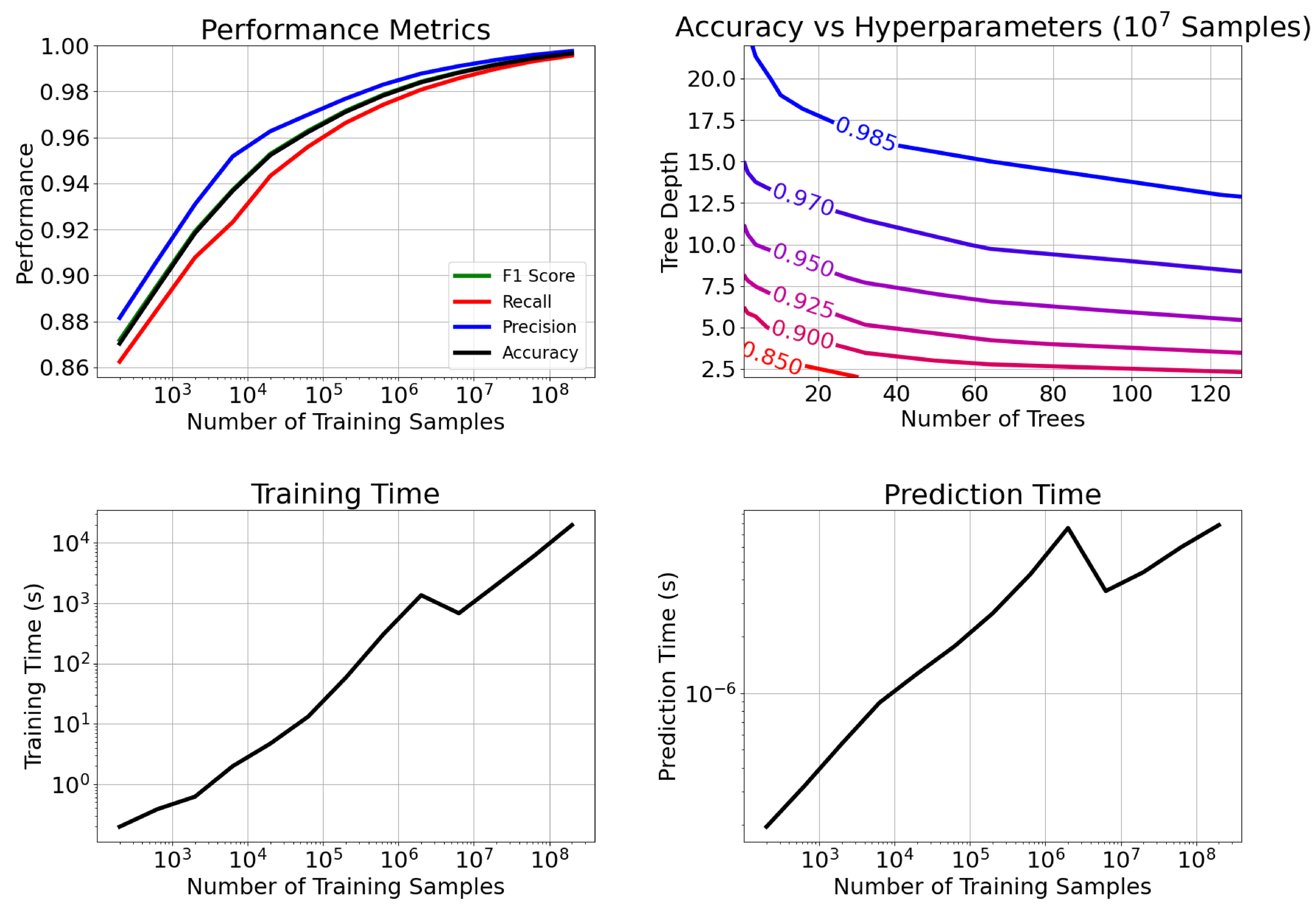
XGBoost Stability Classifier

- Training Dataset: 8D Hyper-rectangle (before conversion to IMAS) of 256M QLK Linear Simulations covering a wide domain of the parameter space to reduce interpolation.
- Trained using XGBoost Decision Tree ensembles (up to 25 depth and 512 trees)
- How does our model accuracy scale with number of training samples?

GKW simulations will be generated with more inputs available than were used for the existing QLK dataset

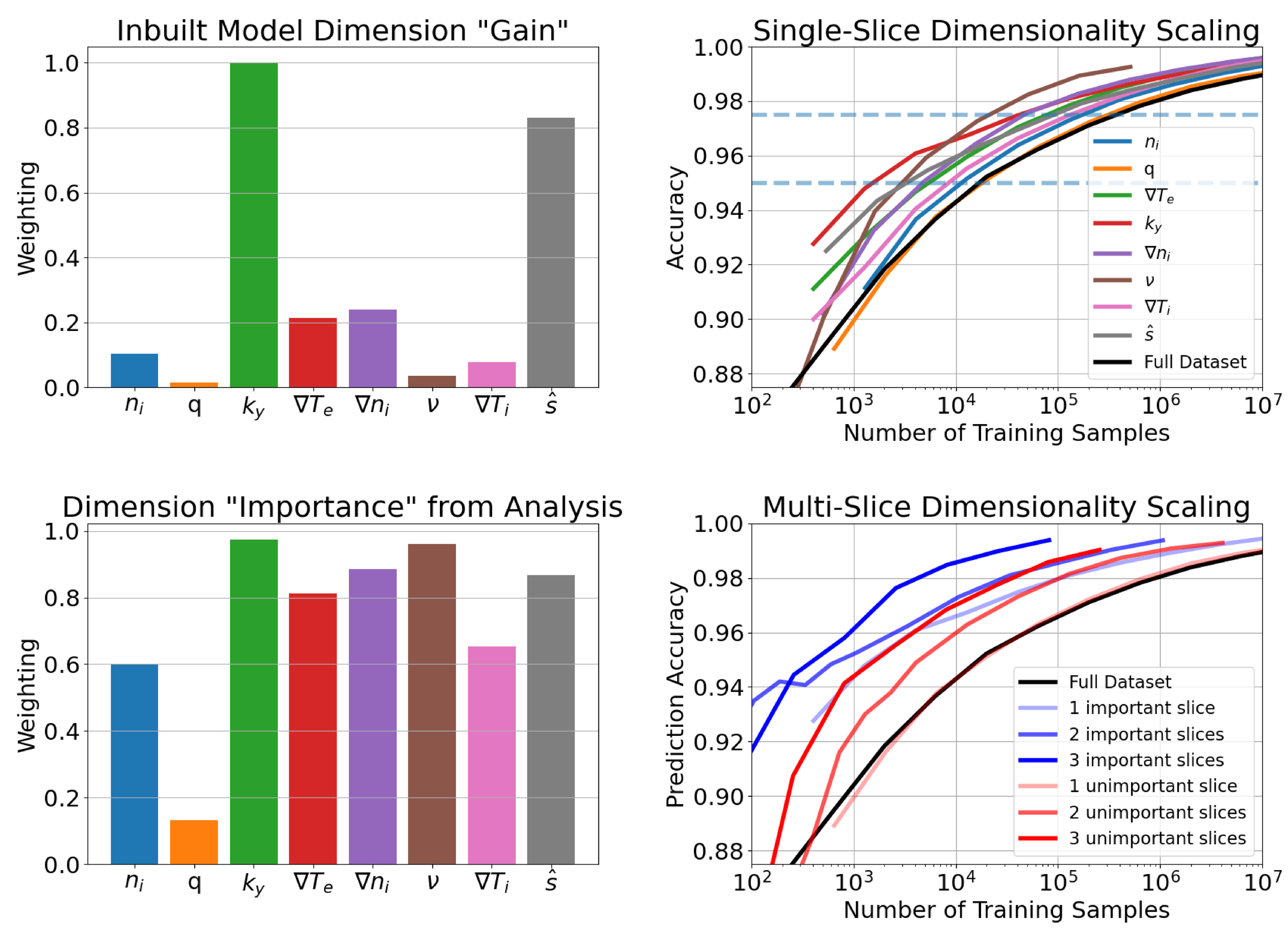
- How is the model scaling affected when an input is fixed?
- Which inputs are the most relevant for the stability decision tree?

Stabiltiy Decision Tree Classifier Scaling



- High baseline accuracy at low amounts of training points (90% with only 1000 points).
- Excellent scaling to almost 100% accuracy when training with 80% of the dataset (200M points).
- Good accuracy even with lower numbers of trees and reduced depth.
- Roughly linear increase in training time and prediction time with number of samples.
- Extremely fast prediction time on the order of 10^{-6} s

Dimensionality Scaling



- Top left: built-in average "information gain" when using a given input to split the data
- Top right: precision gain from fixing a given variable and training new models on subsets of the resulting data slice
- Bottom left: the resulting quantified "importance" of the variables from these results averaged at the 0.95 and 0.975 accuracy thresholds (dashed lines from top right plot)
- Built-in methods of XGBoost to classify variable importance not sufficient
- Bottom right: scaling of fixing multiple slices to further reduce dimensions

Conclusion and Further Work

Conclusion

- Conversion software to link QLK simulations to IMAS format developed and used to convert existing dataset on which to perform training.
- Highly accurate implementation of XGBoost Classifier to quickly predict stability
- Exploration of scaling with number of training points and number of dimensions in preparation for using smaller amounts of higher dimensional data (GKW)

Further Work

- Utilise XGBoost Regressors to predict growth rate and frequencies of dominant mode
- Train NN to predict growth rates and frequencies based on IMAS normalised QLK data
- Generate database of GKW simulations and apply XGBoost and NN pipelines
- Compare performance of different AI models trained on the same data

References

[1] K.L van de Plassche, J. Citrin, C. Bourdelle, Y. Camenen *et al.*, Phys. Plasmas, **27** (2020) 022310
[2] J. Citrin, C. Bourdelle, Y. Camenen *et al.*, Nucl. Fusion, **55** (2015) 092001
[3] A. G. Peeters, Y. Camenen, *et al.*, Comput. Phys. Commun. **180** (2009) 2650
[4] http://qualikiz.com
[5] https://gitlab.com/gkdb/imas-gk
[6] T. Chen, C. Guestrin, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794) (2016). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

This research was funded, in whole or in part, by l'Agence Nationale de la Recherche (ANR), project ANR-22-CE30-0023.

This work has been carried out within the framework of the EUROfusion Consortium, funded by the European Union via the Euratom Research and Training Programme (Grant Agreement No 101052200 — EUROfusion). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.