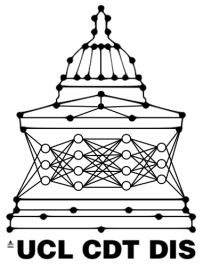


UCL CDT DIS Note

21st April 2020

UK Atomic Energy Authority

Results from TBR Group Project

Petr Mánek[a] and Graham Van Goffrier[a]

[a]University College London

The abstract of my report.

Contents

1 Introduction

The analysis of massive datasets has become a necessary component of virtually all technical fields, as well as the social and humanistic sciences, in recent years. Given that rapid improvements in sensing and processing hardware have gone hand in hand with the data explosion, it is unsurprising that software for the generation and interpretation of this data has also attained a new frontier in complexity. In particular, simulation procedures such as Monte Carlo (MC) event generation can perform physics predictions even for theoretical regimes which are not analytically soluble. The bottleneck for such procedures, as is often the case, lies in the computational time and power which they necessitate.

Surrogate models, or metamodels, can resolve this limitation 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 optimisation 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 the underlying expensive model.

In this internship project, in coordination with the UK Atomic Energy Authority (UKAEA) and Culham Centre for Fusion Energy (CCFE), we sought to develop a surrogate model for the tritium breeding ratio (TBR) in a Tokamak nuclear fusion reactor. Our expensive model was a MC-based neutronics simulation [3], itself a spherical approximation of the Joint European Torus (JET) at CCFE, which returns a prediction of the TBR for a given reactor configuration. We took an empirical approach to the construction of this surrogate, and no results described here are explicitly dependent on prior physics knowledge.

For the remainder of Section 1, we will define the TBR and set the context of this work within the goals of the UKAEA and CCFE. In Section 2 we will describe our datasets generated from the expensive model for training and validation purposes, and the dimensionality reduction methods employed to develop our understanding of the parameter domain. In Section 3 we will present our methodologies for the comparison testing of a wide variety of surrogate modelling techniques, as well as a novel adaptive sampling procedure suited to this application. After delivering the results of these approaches in Section 4, we will give our final conclusions and recommendations for further work.

1.1 Problem Description

Nuclear fusion technology relies on the production and containment of an extremely hot and dense plasma. In this environment, by design similar to that of a star, hydrogen atoms attain energies sufficient to overcome their usual electrostatic repulsion and fuse to form helium [4]. Early prototype reactors made use of the deuterium (^2H) isotope of hydrogen in order to achieve fusion under more accessible conditions, but achieved limited success. The current frontier generation of fusion reactors, such as JET and the under-construction International Thermonuclear

Experimental Reactor (ITER), make use of tritium (^3H) fuel for further efficiency gain. Experimentation at JET dating back to 1997 [5] has made significant headway in validating deuterium-tritium (D-T) operations and constraining the technology which will be employed in ITER in a scaled up form.

However, tritium is much less readily available as a fuel source than deuterium. 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. It may be produced indirectly through irradiation of heavy water (deuterium oxide) during nuclear fission, but only at very low rates which could never sustain industrial-scale fusion power.

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

$$^1_0n + \ ^6_3Li \rightarrow \ ^3_1T + \ ^4_2He \tag{1}$$

$$^1_0n + \ ^7_3Li \rightarrow \ ^3_1T + \ ^4_2He + \ ^1_0n \tag{2}$$

where T represents tritium and ^7Li, ^6Li are the more and less frequently occurring isotopes of lithium, respectively. ^6Li has the greatest tritium breeding cross-section of all tested isotopes [4], but due to magnetohydrodynamic instability of liquid lithium in the reactor environment, a variety of solid lithium compounds are preferred.

The TBR is defined as the ratio between tritium generation in the breeding blanket per unit time and tritium fuel consumption in the reactor. The MC neutronics simulations previously mentioned therefore must account for both the internal plasma dynamics of the fusion reactor and the resultant interactions of neutrons with breeding blanket materials. Neutron paths are traced through a CAD model (e.g. Figure 1) of a reactor with modifiable geometry.

The input parameters of the computationally-expensive TBR model therefore fall into two classes. Continuous parameters, including material thicknesses and packing ratios, describe the geometry of a given reactor configuration. Discrete categorical parameters further specify all relevant material sections, including coolants, armours, and neutron multipliers. One notable exception is the enrichment ratio, a continuous parameter denoting the presence of ^6Li. Our challenge, put simply, was to produce a TBR function which takes these same input parameters and approximates the MC TBR model with the greatest achievable accuracy.

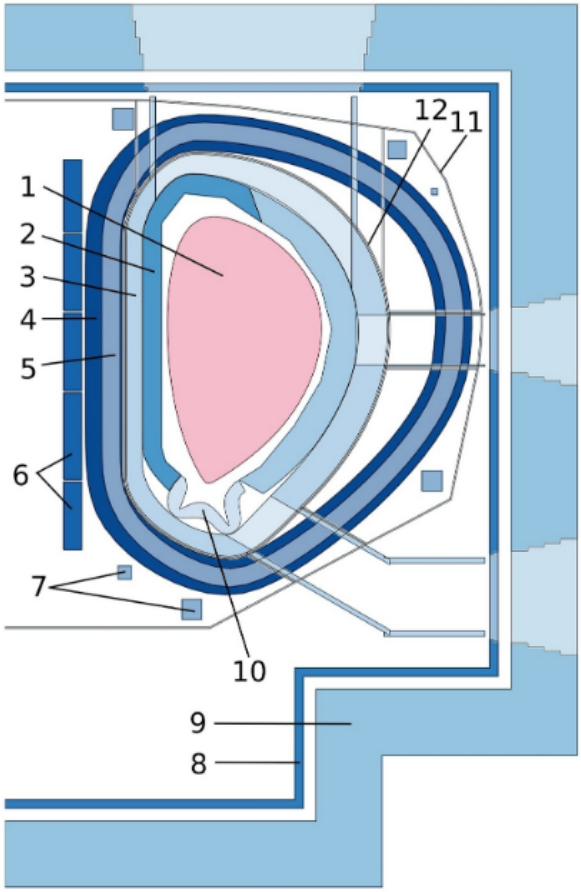


Figure 1: Typical single-null reactor configuration as specified by BLUEPRINT [6]: 1 — plasma, 2 — breeding blankets

2 Data Preprocessing

Data

2.1 Data Description and Initial Sampling

2.2 Dimensionality Reduction

Model training over high-dimensional parameter spaces may be facilitated by carefully reducing the number of variables used to describe the space. For many applications, feature selection strategies succeed in identifying a sufficiently representative subset of the original input variables; however, all given variables were assumed to be physically relevant to the MC TBR model. Feature extraction methods, on the other hand, aim to identify a transformation from the parameter space which decreases dimensionality; even if no individual parameter is separable from the space, some linear combinations of parameters or nonlinear functions of parameters may be.

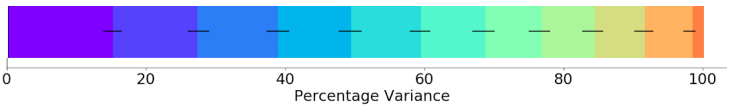


Figure 2: Cumulative variance for optimal features identified by PCA

2.2.1 Principal Component Analysis

To pursue linear feature extraction, principal component analysis (PCA) was performed via scikit-learn on a set of 300,000 uniform samples of the MC TBR model.

Figure 7 shows the resultant cumulative variance of the feature vectors identified by PCA. The similar share of variance among all eleven features demonstrated irreducibility of the TBR model by linear methods.

2.2.2 Variogram Computations

Kriging, or Gaussian process regression, is a geostatistical surrogate modelling technique which relies on correlation functions over distance (lag) in the paramater space [2]. Although kriging performed poorly for our use case due to high dimensionality, these correlation measures gave insight into similarities between discrete-parameter slices of the data.

Figure 3 shows the Matheron semivariance [3] for three discrete slices with coolant material varied, but all other discrete parameters fixed. Fits [4] to the Matérn covariance model confirmed numerically that the coolant material is the discrete parameter with the greatest distinguishability in the MC TBR model.

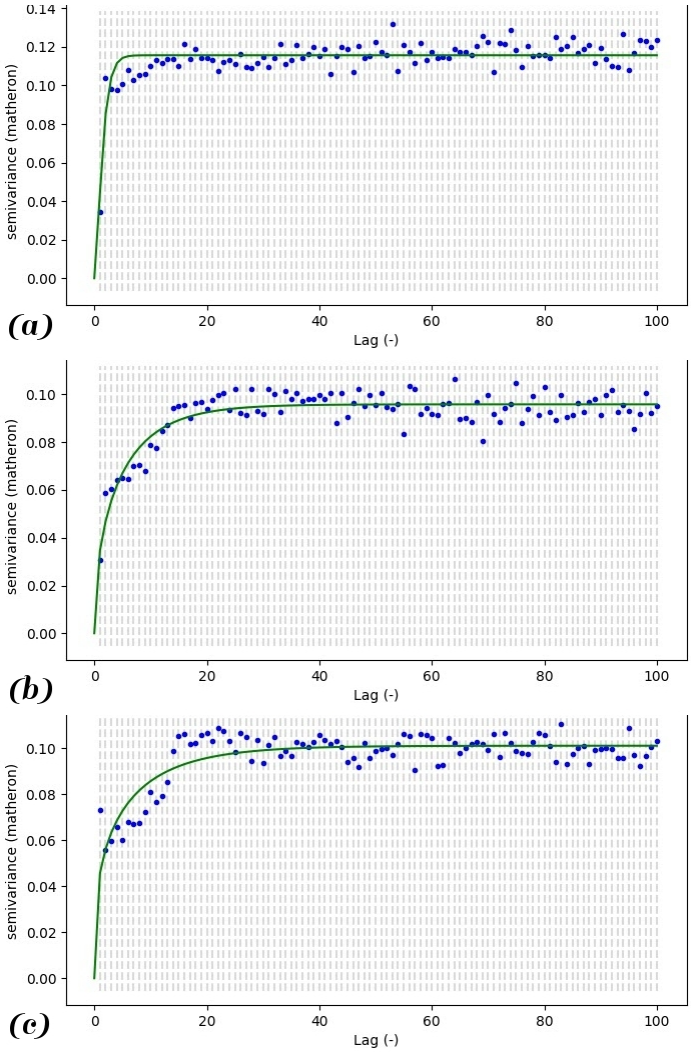


Figure 3: Semivariograms for MC TBR data with coolant materials: (a) He, (b) H_2O, (c) D_2O

** Make sure discrete slices are mentioned before this **

2.2.3 Autoencoders

Question: is it accurate to say this represents nonlinear feature extraction? re my intro paragraph

3 Methodology

Assuming that input has been appropriately treated to eliminate redundant features, we may turn to characterise proposed surrogate models and the criteria used for their evaluation. The task all presented surrogates strive to solve can be formulated using the language of conventional regression problems. In the scope of this work, we explore various possible choices available to us in the scheme of supervised and unsupervised learning.

Labeling the expensive Monte Carlo simulation $f(x)$, a surrogate is a mapping $\hat{f}(x)$ that yields similar images as $f(x)$. In other words, $f(x)$ and $\hat{f}(x)$ minimise a selected similarity metric. Furthermore, in order to be considered *viable*, surrogates are required to achieve expected evaluation time that does not exceed the expected evaluation time of $f(x)$.

In the supervised learning setting, we first gather a sufficiently large training set of samples $\mathcal{T} = \left\{\left(x^{(i)}, f\left(x^{(i)}\right)\right)\right\}_{i=1}^{N}$ to describe the behaviour of $f(x)$ across its domain. Depending on specific model class and appropriate choice of its hyperparameters, surrogate models $\hat{f}(x)$ are trained to minimise empirical risk with respect to $\mathcal{T}$ and a model-specific loss function $\mathcal{L}$, where empirical risk is defined as $R_{\text{emp.}}(\hat{f} \mid \mathcal{T}, \mathcal{L}) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(\hat{f}(x^{(i)}), f(x^{(i)})\right)$.

The unsupervised setting can be viewed as an extension of this method. Rather than fixing the training set $\mathcal{T}$ for the entire duration of training, multiple sets $\{\mathcal{T}_k\}_{k=0}^{K}$ are used, such that $\mathcal{T}_{k-1} \subset \mathcal{T}_k$ for all $k > 1$. The first set $\mathcal{T}_0$ is initialised randomly to provide a *burn-in*, and is repeatedly extended in epochs, whereby each epoch trains a new surrogate on $\mathcal{T}_k$ using the supervised learning procedure, evaluates its performance, and forms a new set $\mathcal{T}_{k+1}$ by adding more samples to $\mathcal{T}_k$. This permits the learning algorithm to condition the selection of new samples by the results of evaluation in order to focus on improvement of surrogate performance in complex regions within the domain.

3.1 Metrics

Aiming to provide objective comparison of a diverse set of surrogate model classes, we define a multitude of metrics to be tracked during experiments. Following the motivation of this work, two desirable properties of surrogates arise: (i) their capability to approximate the expensive model well and (ii) their time of evaluation. An ideal surrogate would maximise the former while minimising the latter.

Table 1 provides exhaustive listing and description of metrics recorded in the experiments. For regression performance analysis, we include a selection of absolute metrics to assess the approximation capability of surrogates, and set practical bounds on the expected accuracy of their predictions. In addition, we also track relative measures that are better-suited for model comparison between works as they maintain invariance with respect to the selected domain and image space. For analysis of evaluation time, surrogates are assessed in terms of wall time elapsed during training and predicition. This closely models the practical use case, in which they are trained and used as drop-in replacements for the expensive model. Since training set sizes remain to be determined, all times are reported per a single sample. Even though some surrogates support acceleration by means of parallelisation, measures were taken to ensure sequential processing of samples to achieve comparability between considered models.

Regression performance metrics	Mathematical formulation / description	Ideal value	
Mean absolute error (MAE)	$\sum_{i=1}^{N} \lvert y^{(i)} - \hat{y}^{(i)} \rvert / N$	0	[TBR]
Standard error of regression S	$\mathrm{StdDev}_{i=1}^{N} \left\{ \lvert y^{(i)} - \hat{y}^{(i)} \rvert \right\}$	0	[TBR]
Coefficient of determination R^2	$1 - \sum_{i=1}^{N} \left(y^{(i)} - \hat{y}^{(i)} \right)^2 / \sum_{i=1}^{N} \left(y^{(i)} - \overline{y} \right)^2$	1	[rel.]
Adjusted R^2	$1 - (1 - R^2)(N-1)/(N-P-1)$	1	[rel.]
Evaluation time metrics			
Mean training time $\bar{t}_{\text{trn.}}$	(wall training time of $\hat{f}(x)$)$/N_0$	0	[ms]
Mean prediction time $\bar{t}_{\text{pred.}}$	(wall prediction time of $\hat{f}(x)$)$/N$	0	[ms]
Relative speedup	(wall evaluation time of $f(x)$)$/(N\bar{t}_{\text{pred.}})$	$\to \infty$	[rel.]

Table 1: Metrics recorded in supervised learning experiments. In formulations, we work with training set of size N_0 and testing set of size N, TBR values $y^{(i)} = f(x^{(i)})$ and $\hat{y}^{(i)} = \hat{f}(x^{(i)})$ denote images of the ith testing sample in the expensive model and the surrogate respectively. Furthermore, the mean $\overline{y} = \sum_{i=1}^{N} y^{(i)}/N$ and P is the number of input features.

To prevent undesirable bias in results due to training set selection, all metrics are collected in the scheme of k-fold cross-validation with a standard choice of $k = 5$. Herein, a sample set is subdivided into 5 disjoint folds which are repeatedly interpreted as training and testing sets, maintaining a constant ratio of samples between the two. In each such interpretation experiments are repeated, and the overall value of each metric of interest is reported as the mean across all folds.

3.2 Supervised Learning Experiments

3.2.1 Considered Surrogates

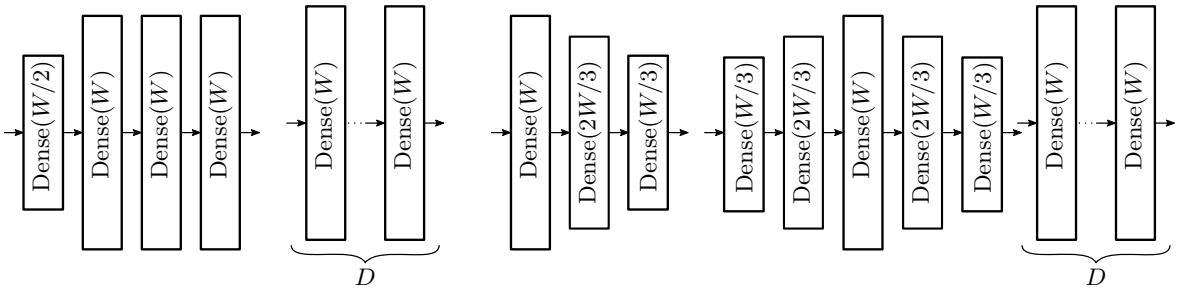
Surrogate	Acronym	Implementation	Hyperparameters
Support vector machines	SVM	SciKit Learn	TODO
Gradient boosted trees	GBT	SciKit Learn	TODO
Extremely random trees	ERT	SciKit Learn	TODO
AdaBoost	AB	SciKit Learn	TODO
Gaussian process regression	GPR	SciKit Learn	TODO
k nearest neighbours	KNN	SciKit Learn	TODO
Neural networks	NN	Keras (TensorFlow)	TODO
Inverse distance weighing	IDW	SMT	TODO
Radial basis functions	RBF	SMT	TODO
Stochastic gradient descent	SGD	SciKit Learn	TODO
Ridge regression	RR	SciKit Learn	TODO
Kriging	KRG	SMT	TODO

Table 2: TODO

TODO: describe classes of surrogates

TODO: reference implementations and original papers

TODO: define NN architectures



(a) 1h3f(W) (b) df(D, W) (c) 3pyr(W) (d) 5diam(W) (e) 6sym(W)

Figure 4: TODO

3.2.2 Experiments

TODO: single slice hyperparameter optimisation

TODO: multislice (mixed) hyperparameter optimisation

TODO: retraining for scaling benchmark

TODO: retraining on large set

3.3 Adaptive Sampling

All of the surrogate modelling techniques studied in this project face a shared challenge: their accuracy is limited by the quantity of training samples which are available from the expensive MC model. Adaptive sampling procedures can improve upon this limitation by taking advantage of statistical information which is accumulated during the training of any surrogate model. Rather than training the surrogate on a single sample set generated according to a fixed strategy, sample locations are chosen incrementally so as to best suit the model under consideration.

Adaptive sampling techniques are widespread in the literature and have been specialised for surrogate modelling. Garud's [1] "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 theorised 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 parameter space. MCMC produces a progressive chain of sample points, each drawn according to the same symmetric proposal distribution from the prior point. These sample points will converge to a desired posterior distribution, so long as the acceptance probability for these draws has a particular functional dependence on that posterior value (see [2] for a review).

Many researchers have embedded surrogate methods into MCMC strategies for parameter optimisation [3][4], in particular the ASMO-PODE algorithm [5] which makes use of MCMC-based adaptive sampling to attain greater surrogate precision around prospective optima. Our novel 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) to iteratively increment the training/test set with sample points which maximise surrogate error and minimise a crowding distance metric (CDM) [6] in parameter space. On each iteration following an initial training of the surrogate on N uniformly random samples, the surrogate was trained and AE calculated for . MCMC was then performed on 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.

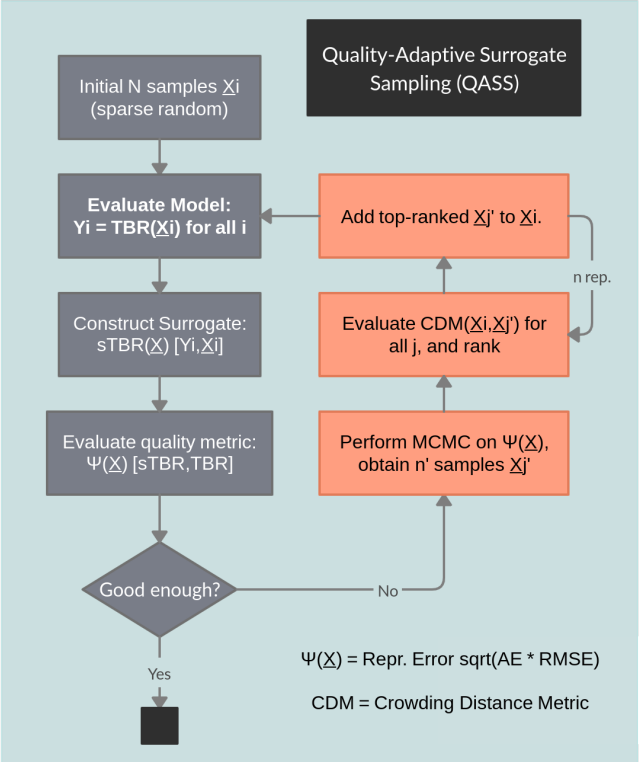


Figure 5: Schematic of QASS algorithm

4 Results

4.1 Results of Model Comparisons

4.2 Results of Adaptive Sampling

Define sinusoidal toy model and justify

Explain hyperparameter tests: initsamples, stepsamples, MCMC length

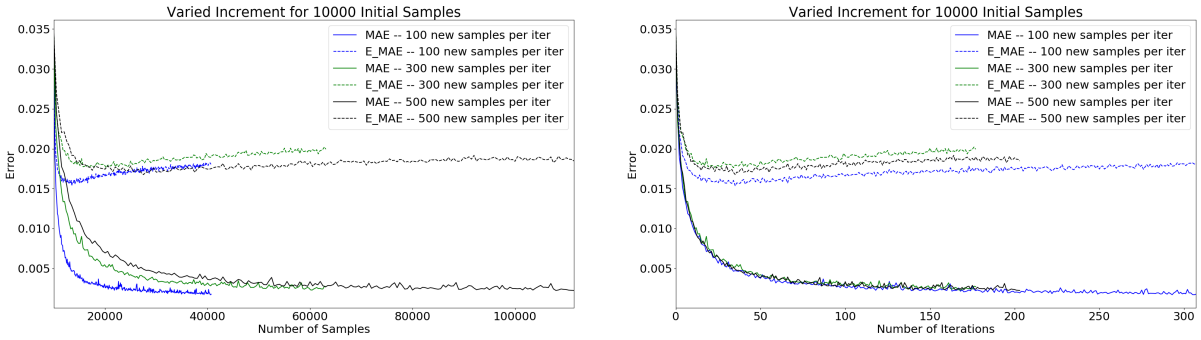


Figure 6: QASS absolute training error over total sample quantity (left) and number of iterations (right)

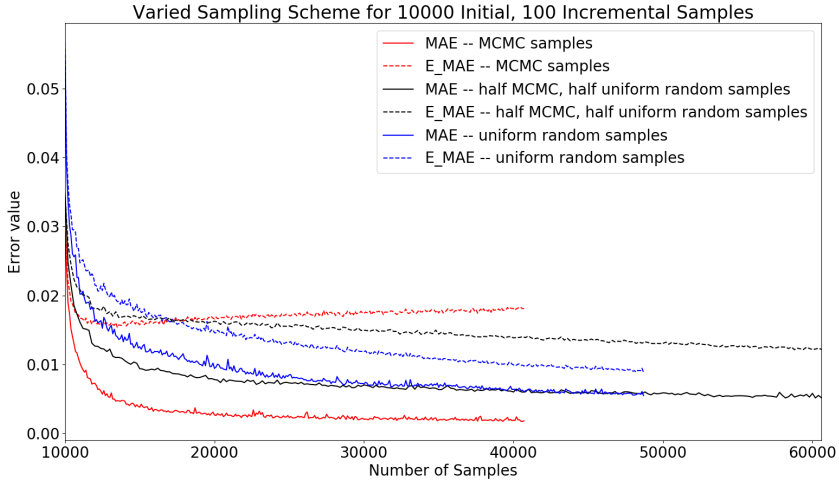


Figure 7: Absolute training error for QASS, uniform random scheme, and mixed scheme

5 Conclusion

Conclusion