# Bayesian optimisation of poloidal field coil positions in tokamaks

Timothy Nunn[1]  Kamran Pentland[1]  Vignesh Gopakumar[1, 2]  James Buchanan[1]

**Abstract.** The tokamak is a world-leading concept for producing sustainable energy via magnetically-confined nuclear fusion. Identifying where to position the magnets within a tokamak, specifically the poloidal field (PF) coils, is a design problem which requires balancing a number of competing economic, physical, and engineering objectives and constraints. In this paper, we show that multi-objective Bayesian optimisation (BO), an iterative optimisation technique utilising probabilistic machine learning models, can effectively explore this complex design space and return several optimal PF coil sets. These solutions span the Pareto front, a subset of the objective space that optimally satisfies the specified objective functions. We outline an easy-to-use BO framework and demonstrate that it outperforms alternative optimisation techniques while using significantly fewer computational resources. Our results show that BO is a promising technique for fusion design problems that rely on computationally demanding high-fidelity simulations.

**Keywords.** Bayesian optimisation ● Poloidal field coils ● Spherical tokamak ● MHD equilibria ● FreeGS

## 1. Introduction

### 1.1. Motivation and aims

A spherical tokamak is a torus-shaped device with a low aspect ratio that uses strong magnetic fields to confine and control a thermonuclear fusion plasma, with the goal of producing fusion energy (Freidberg, 2008). The Spherical Tokamak for Energy Production (STEP), currently in the design phase and targeting completion in 2040 (Chapman et al., 2024), is one of a few ongoing international fusion research and development projects based on the spherical tokamak concept. To deliver fusion power to the grid on such a short timescale, researchers are increasingly designing next-generation tokamaks in silico with multi-physics simulations, many of which require high-performance computing (HPC) resources.

STEP is no exception, with initial concept designs for the tokamak (and the associated plasma) being generated by low-fidelity integrated modelling codes (Muldrew et al., 2024) such as PROCESS (Morris et al., 2024; Muldrew et al., 2020) and Bluemira (Coleman and McIntosh, 2019; Coleman et al., 2025; Franza et al., 2022). These codes use simplified physics and engineering models to produce designs within seconds or minutes. In contrast, more complex medium- to high-fidelity codes, such as JINTRAC (Romanelli et al., 2014),

incorporate more detailed physics models but can require days or weeks to complete a single simulation. These higher-fidelity simulations play a crucial role in refining, integrating, and validating the initial concept design across the entire fusion power plant (Davis et al., 2024). Making the most efficient use of these computationally expensive simulations is critical if we wish to accelerate the design of future fusion power plants like STEP.

Our focus here will be on the design of the poloidal field (PF) coil system, which plays a critical role in controlling the position and shape of the plasma in both the core and divertor regions of a tokamak (Lim et al., 2010). In particular, some coils are crucial for managing the vertical stability of elongated plasmas, such as those in spherical tokamaks, where the higher elongation can lead to larger vertical instability, risking disruption without appropriate control (Anand et al., 2023). By generating poloidal magnetic fields, the PF coil system ensures the plasma remains in equilibrium, balancing the inward-facing magnetic forces produced by the coils against the outward-facing pressure-driven forces generated by the plasma (Wesson and Campbell, 2011, Chp. 3.1). The design of the system—in terms of the coil positions, size, and shape—will have a significant impact on plasma performance and stability and will therefore need to satisfy a number of competing (and often conflicting) constraints. In the plasma, for example, constraints are required to ensure X-points form in specific locations (for stability), strikepoints hit the correct divertor plates (for heat management), and total current density limits on the PF coils are not exceeded. In terms of the tokamak itself, the locations/sizes of the coils will inevitably be constrained by the vacuum vessel, diagnostic systems, and maintenance ports (to name but a few).

In addition to constraints, there will be a number of objectives related to the desired operational plasma conditions that we wish the chosen coil set to minimise or maximise (depending on the objective). This could include minimising the coil size to reduce fabrication, construction, and installation costs or could include minimising current flows to reduce power consumption and structural stresses from forces produced by the coils (Coleman and McIntosh, 2020). Moreover, we may wish to optimise certain properties of the plasma in the divertor chambers in order to minimise heat loads on plasma facing components and improve exhaust performance (Hudoba et al., 2023b). Simultaneously satisfying both the objectives and constraints will require the solution of a complex optimisation problem that needs to be tackled in a systematic, computationally efficient manner.

[1] United Kingdom Atomic Energy Authority, Culham Campus, Abingdon, Oxfordshire, OX14 3DB, United Kingdom (timothy.nunn@ukaea.uk)
[2] Centre for Artificial Intelligence, Department of Computer Science, University College London, London, WC1V 6LJ

In this paper, we will perform multi-objective Bayesian optimisation (BO) on an earlier baseline design of the STEP PF coil system (Hudoba et al., 2023a). Our aims are to:

(i) design and outline an easy-to-use BO framework which is flexible, data efficient (reducing the computational cost of design), and can yield more optimal designs than obtained through other exhaustive optimisation schemes.

(ii) identify a *Pareto front*, i.e. a set of optimal PF coil locations, that outperform the baseline for some given objectives and constraints.

(iii) encourage more widespread adoption of BO for the in silico design of interlinked components on future tokamak devices to save time, minimise financial costs, and improve plasma performance.

We should stress that this work has not had a direct impact on the current design of the STEP PF coil system (Nasr et al., 2024) and is instead a demonstration of a generalisable BO framework for PF coil system design. We do wish to highlight, however, that the framework is completely machine agnostic and can be used with different objectives and constraints to the ones we use here. It is the hope that frameworks such as this will be adopted more regularly within the integrated modelling codes currently used for tokamak design.

## 1.2. Related work

PF coil sets are typically optimised using integrated modelling codes for tokamak power plant design. A common approach is to force the PF coils to lie on a contour "rail" that surrounds the core plasma, reducing the number of degrees of freedom in the optimisation problem (Coleman and McIntosh, 2020; Meneghini et al., 2024). Exclusion zones along the rails enforce engineering constraints, before nonlinear (non-Bayesian) optimisation is performed with respect to some pre-specified objectives and constraints on the plasma boundary shape.

While well-established, rail-based methods can restrict the PF coil design space, often rely on estimated objective function gradients, and can struggle with multiple competing objectives. They are primarily suited to conventional aspect ratio tokamaks, where PF coil rails are placed outside (and close to) the toroidal field (TF) coils, sometimes leading to intersection issues. BO, on the other hand, performs gradient-free global optimisation, can handle diverse constraints, and uses a surrogate model of the multi-output objective function to intelligently guide function evaluations. This helps balance exploration of new designs and exploitation of known optimal designs, leading to high levels of data efficiency.

Despite these advantages, the adoption and application of BO in fusion engineering and design has, so far, remained relatively limited. Brown et al. (2024a,b) use BO to optimise the current-profiles in STEP. Their goal was to improve six key properties of the safety factor profile. They additionally demonstrate that BO performs better than a genetic algorithm with the same number of black-box function evaluations. Mehta et al. (2024) use BO to find the parameters

such as neutral beam injection power, plasma current, and plasma elongation in the DIII-D tokamak that safeguard against disruption during the ramp-down phase. Similarly, Pusztai et al. (2023) use BO to mitigate the impact of disruptions in ITER by exploring how injected deuterium and neon can minimise runaway electron currents, transported heat, and quench time post-disruption. Järvinen et al. (2022) also investigate runaway electron currents using BO as an advanced sampling method to help calibrate uncertainty and minimise the discrepancy between simulations and experimental data. For fusion component design, Humphrey et al. (2023) demonstrates BO to minimise stresses in parametrised divertor monoblock under fusion conditions. The most relevant work to ours is that of Nunn et al. (2023), who use multi-objective BO to optimise TF coil shapes to reduce both financial costs and magnetic ripples (which affect plasma stability and performance). In contrast, our approach deals with more computationally expensive, failure-prone plasma equilibrium simulations without analytic objective/constraint functions, necessitating the use of a classifier alongside the surrogate model (as we will see later on).

The work here is inspired by that of Hudoba et al. (2023a), in which the authors seek to optimise the STEP PF coil system by minimising deviations of key plasma parameters (from a baseline scenario which we adopt) and coil currents, while maximising divertor performance metrics. Using a free-boundary equilibrium solver, thousands of potential PF coil sets are sampled and evaluated (in a Monte Carlo-type approach) before optimal solution sets are identified heuristically. We aim to provide and fully outline an alternative, much more data efficient, framework for carrying out similar multi-objective optimisation that can return a STEP equilibrium similar to the baseline.

There are also a number of areas in fusion design where BO has yet to be applied but could potentially offer significant benefits. For example, parameter scans for optimal magnetic sensor placement, as explored for TCV (Romero and Svensson, 2013) and SPARC (Stewart et al., 2023), could benefit from BO's sample efficiency, saving computational resources and time. Similarly, these benefits could transfer to existing frameworks for stellarator coil design (Giuliani, 2024; Jorge et al., 2024; Kaptanoglu et al., 2025).

## 1.3. Outline

In Section 2, we describe the multi-objective BO problem, the Gaussian process surrogate model, the classifier scheme, and the acquisition function required in the BO loop. We follow this in Section 3 by defining the PF coil design problem in terms of the input space, the objectives we seek to optimise, and the constraints on the plasma and the machine. In addition, we present the simulator used to generate the plasma equilibria for each PF coil set and define cases in which the simulator may fail to produce a valid equilibrium (requiring the classifier). The numerical experiments are detailed and presented in Section 4. To highlight the data efficiency of the BO scheme, we carry out a number of experiments with a fixed computational budget and assess performance against alternative optimisation methods. In Section 5, we

discuss our findings, highlight any major advantages and disadvantages of the BO framework applied to this problem, and propose avenues for future work.

## 2. Multi-objective Bayesian optimisation

BO is a method for performing gradient-free global optimisation of black-box functions, typically utilised when the function is expensive-to-evaluate (Garnett, 2023). This is because practitioners will often want to identify (feasible) optimal points of the function's input/output spaces with as few function evaluations as possible—especially if there is a limited computational budget.

Here, we are interested in optimising the nonlinear function $\boldsymbol{f}: \mathcal{U} \subseteq \mathbb{R}^d \to \mathbb{R}^{l+m}$ that takes in a $d$-dimensional input and returns $l$ objectives and $m$ constraints. More formally, the aim of *multi-objective BO* ($l > 1$), is to solve

$$\operatorname*{argmin}_{\substack{\boldsymbol{x} \in \mathcal{U} \\ \boldsymbol{f}_{[l+1..l+m]} \preccurlyeq 0}} \boldsymbol{f}_{[1..l]}(\boldsymbol{x}), \tag{2.1}$$

where $\boldsymbol{f}_{1..k}$ denotes the first $k$ components of $\boldsymbol{f}$ and $\preccurlyeq$ denotes a component-wise less than or equal to comparison.

Given we need to optimise over multiple competing objectives, problems such as (2.1) will often involve trade-offs where improving one objective may come at the expense of another. The aim is therefore to seek the set of *Pareto optimal* solutions $\mathcal{P}$ that are not *dominated* by any other solutions. A solution $\boldsymbol{x}$ *dominates* another $\boldsymbol{x}'$, denoted $\boldsymbol{x} \prec \boldsymbol{x}'$, if and only if $\boldsymbol{f}(\boldsymbol{x}) \preccurlyeq \boldsymbol{f}(\boldsymbol{x}')$ and $\exists j \in \{1, \ldots, l\}$ such that $\boldsymbol{f}_j(\boldsymbol{x}) < \boldsymbol{f}_j(\boldsymbol{x}')$. In short, a solution $\boldsymbol{x}$ dominates $\boldsymbol{x}'$ if it is at least as good in all objectives and strictly better in at least one. Given a dataset

$$\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{f}(\boldsymbol{x}_i))\}_{i=1}^N,$$

consisting of $N$ evaluations of $\boldsymbol{f}$, the *Pareto set* for (2.1) is defined as

$$\mathcal{P}(\mathcal{D}) \coloneqq \{\boldsymbol{x} \in D \mid \nexists \, \boldsymbol{x}' \in D \text{ s.t. } \boldsymbol{x}' \prec \boldsymbol{x}\}. \tag{2.2}$$

The *Pareto front*, denoted $\mathcal{P}_{\boldsymbol{f}}$, is defined as the image of the Pareto set, i.e. $\mathcal{P}_{\boldsymbol{f}} \coloneqq \{\boldsymbol{f}(\boldsymbol{x}) \mid \boldsymbol{x} \in \mathcal{P}(\mathcal{D})\}$. See Garnett (2023, Chp. 11.7) for an illustration of the Pareto front.

### 2.1. The Bayesian optimisation loop

The key component in BO for identifying feasible and optimal trade-offs between the objectives is a *probabilistic surrogate model*, capable of performing uncertainty based exploration. This model is typically trained on some initial dataset by maximising its marginal likelihood—more details on this surrogate model are given in Section 2.2.

The first stage in BO (refer to Figure 2.1) is to construct this initial dataset (which we will call $\mathcal{D}$) by taking $N$ samples $\boldsymbol{x} \in \mathcal{U}$ and evaluating them all using $\boldsymbol{f}$. One popular method used is Sobol sampling (Sobol, 1967), whereby samples are chosen quasi-randomly with low discrepancy to
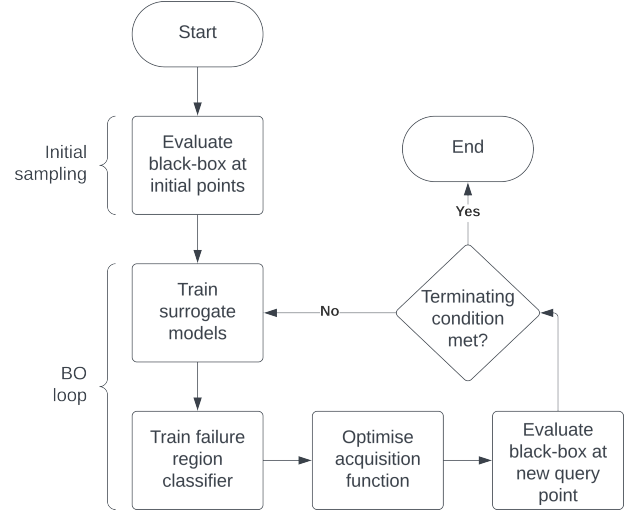


**Figure 2.1** A flowchart illustrating the standard "BO loop" along with an additional failure region classifier step—see text for more details.

achieve approximately uniform coverage of the input space. The number of samples $N$ chosen/required may depend on the size of $d$, the computational budget available, and if parallel processing is available (for the $\boldsymbol{f}$ evaluations). Note that at this point, while we could use $D$ to immediately generate a Pareto set $\mathcal{P}(\mathcal{D})$, this would almost certainly be a poor estimate given a lack of data points and that most would reside in non-optimal regions of the objective space.

It is worth noting that for many black-box functions such as $\boldsymbol{f}$, there will be *failure regions* of the input space that cannot be evaluated. The reasons for failure in our particular setting are discussed further in Section 3.2.5. During construction of the initial dataset, samples that lie in failure regions may be encountered and so we do not wish to include these in the dataset. We do, however, wish to learn from these samples so that we do not encounter similar samples again and therefore we train a *classifier* to predict when this may happen—a similar approach was taken by Hornsby et al. (2024) when generating gyrokinetic simulation datasets for spherical tokamaks. This classifier will be used within the *BO loop*, which can be seen in Figure 2.1 and is now outlined:

**Stage 1:** Generate the initial dataset $\mathcal{D}$ using the Sobol sampling scheme.

**Stage 2:** Train the surrogate model using the dataset $\mathcal{D}$ (see Section 2.2) to obtain a probabilistic model: $\boldsymbol{f}(\boldsymbol{x}) \mid \mathcal{D}$.

**Stage 3:** Train the failure region classifier using the data and the surrogate (see Section 2.3).

**Stage 4:** Maximise the *acquisition function* over the input space to identify the most "informative" point $\boldsymbol{x}^*$ to observe next, given the current surrogate model (and classifier) $\boldsymbol{f}(\boldsymbol{x}^*) \mid \mathcal{D}$ (see Section 2.4).

**Stage 5:** Evaluate $\boldsymbol{f}(\boldsymbol{x}^*)$ and add it to the dataset: $\mathcal{D} \coloneqq \mathcal{D} \cup \{(\boldsymbol{x}^*, \boldsymbol{f}(\boldsymbol{x}^*))\}$.

**Stage 6:** Check whether the terminating condition is met

and if not, return to Stage 2. In our case, we check whether or not the maximum number of iterations $j$ has been exceeded (to remain within our computational budget)[1].

## 2.2. The Gaussian process surrogate

In BO, the most common type of surrogate used is a Gaussian process (GP), which is a probabilistic machine learning model for performing inference on the value of a function $f\colon \mathcal{U} \to \mathbb{R}$ given some training data—see Garnett (2023, Chp. 2). It is characterised by a mean function $m\colon \mathcal{U} \to \mathbb{R}$ and a positive semi-definite covariance function $k\colon \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ (with $k(\cdot, \cdot) \geqslant 0$) such that the prior can be defined as

$$f \sim \mathcal{GP}(m, k). \tag{2.3}$$

The distribution of this prior is the joint distribution of (infinitely) many Gaussian random variables and can be thought of as a distribution over functions. Therefore, at a finite set of evaluation points $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots\} \subset \mathcal{U}$ we have that

$$f(\boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2.4}$$

where $\boldsymbol{\mu} = [m(\boldsymbol{x}_1), m(\boldsymbol{x}_2), \ldots]^{\mathsf{T}}$ is the mean vector and $[\boldsymbol{\Sigma}]_{i,j} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)\ \forall i, j \in \{1, 2, \ldots\}$ is the covariance matrix.

Training a GP requires conditioning the prior (2.3) on the dataset[2] of known function evaluations $\mathcal{D}$ such that we obtain the following posterior distribution

$$f(\boldsymbol{X}) \mid \mathcal{D} \sim \mathcal{N}(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}}). \tag{2.5}$$

This conditioning can be done analytically (see Garnett (2023, Chp. 2.2) for formulae for $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$) and effectively tells the model to assign higher probability to functions that fit the training data well.

The quality of this posterior distribution (in terms of the mean accuracy and variance calibration), however, is highly dependent on the choices made for the functions $m$ and $k$. A typical choice for the mean function is $m \equiv 0$, which assumes no prior knowledge of the function being modelled and ensures model predictions from (2.5) are heavily influenced by the training data. The choice of covariance kernel is formed via our prior belief in the expected behaviour of the true function being modelled (e.g. non-periodicity and smoothness). The covariance function used here is the Matérn-(1/2) (or exponential) kernel

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2}{\ell}\right),$$

where $\|\cdot\|_2$ denotes the Euclidean distance (Pandit and Infield, 2019). In addition, the parameters $\ell$ and $\sigma$ define the input length scale (smaller values produce more 'wiggly'

functions) and the function noise (smaller values lead to lower predictive uncertainty in the function). The covariance kernel encodes the relationship between input points and the resulting covariance matrix quantifies how a change in one point influences changes in another across the domain. The *hyperparameters* $\ell$ and $\sigma$ are tuned (for example, using traditional non-Bayesian optimisation algorithms) to produce the best fit to the training data such that the marginal log-likelihood of the posterior (2.5) is maximised.

It should be noted that while we have described *scalar* output GPs here, in practice we model each output dimension of $\boldsymbol{f}$ using its own scalar GP. This assumes each output of $\boldsymbol{f}$ is uncorrelated (i.e. independent) of one another and means that we require $l + m$ "stacked" GPs to model the joint distribution over $\boldsymbol{f}$. More importantly for BO, it is crucial that the surrogate model is relatively cheap to train and evaluate compared to the cost of evaluating $\boldsymbol{f}$.

## 2.3. The classifier

The aim of classification is to label each data point in the input space as either a failure (0) or non-failure (1). We model the probability of failure by 'wrapping' the GP surrogate model in a logistic function (Williams and Barber, 1998):

$$p(\text{non-failure} \mid \boldsymbol{x}) = \frac{1}{1 + e^{-f(\boldsymbol{x})}} \quad \boldsymbol{x} \in \mathcal{U}.$$

This transforms the GP from a regression model into a classifier model by squishing the GPs prediction into a probability. This probability can be converted into a binary prediction by wrapping it in a shifted Heaviside function (e.g returning 1 if the probability is $\geqslant 0.5$, else 0).

The GP model with classification now has a Bernoulli likelihood $p(\mathcal{D}|f)$, making the calculation of the posterior distribution $p(f|\mathcal{D})$ analytically intractable—unlike in (2.5) where the prior, likelihood, and, therefore, posterior were all Gaussian (see Williams and Rasmussen (2006, Chp. 3.4)). To address this, we approximate the posterior using a variational distribution $q(f|\mathcal{D}; \boldsymbol{\lambda})$, chosen such that its likelihood $q(D|f; \boldsymbol{\lambda})$ is Gaussian—with $\boldsymbol{\lambda}$ parameterising the new distribution (Tran et al., 2015). The parameters $\boldsymbol{\lambda}$ are found by maximising the evidence lower bound

$$\mathcal{L}(\boldsymbol{\lambda}) := \mathbb{E}_{q(D|f; \boldsymbol{\lambda})}[\log p(D|f)] - \mathrm{KL}[q(f; \boldsymbol{\lambda}) \parallel p(f)].$$

The first term represents the expected log likelihood (observing the training data given the probability distribution over functions), while the second term denotes the (non-negative) Kullback–Leibler divergence between the two distributions. Clearly if the KL divergence was zero (the distributions were identical), we would be maximising over the original (log) Bernoulli likelihood. Once the $\boldsymbol{\lambda}$ are found, the GP can be conditioned on the data (as was shown in Section 2.2) using the new Gaussian likelihood $q(D|f; \lambda)$.

In classification, imbalance in the dataset–where one label is significantly more prevalent–can create a poor quality classifier. This results in the classifier being accurate by simply predicting the majority class, rendering it useless

---

[1]Other terminating conditions include stopping once improvements in the acquisition function are below some threshold or when the objectives are deemed to be sufficiently optimal (Ishibashi et al., 2023).

[2]In practice, the data in $\mathcal{D}$ is transformed such that the outputs ($f(\boldsymbol{x})$) are standardised to have a mean of 0 and a variance of 1.

for identifying failure regions. To combat this, we employ *oversampling*, which randomly duplicates samples in the minority class such that both labels are equally represented in the training dataset (Hoens and Chawla, 2013). As a result, the classifier cannot achieve a high accuracy by simply predicting one class and a higher quality model is produced.

## 2.4. The acquisition function

Based on knowledge from the trained GP and classifier, the acquisition function provides us with a way to estimate the informativeness of evaluating $\boldsymbol{f}$ at previously unseen values of $\boldsymbol{x} \in \mathcal{U}$. Depending on the task at hand, there are many possible choices of acquisition functions, each tailored to specific objectives. As mentioned before, the key factor in selecting an appropriate one is that it should be computationally cheap (compared to $\boldsymbol{f}$) to evaluate given the surrogate model.

Here, we use the *expected hypervolume improvement* (EHVI) function, which seeks to quantify the expected improvement in the hypervolume of $\mathcal{P}_{\boldsymbol{f}}$ when adding a new point to the dataset $\mathcal{D}$ (Daulton et al., 2020). EHVI is particularly suited to multi-objective optimization, as it effectively balances exploration and exploitation by focusing on regions of the search space that are both uncertain and potentially optimal. The EHVI function $\alpha_{\mathrm{EHVI}} \colon \mathcal{U} \to \mathbb{R}$ is given by

$$\alpha_{\mathrm{EHVI}}(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{f}}[\mathrm{HV}(\mathcal{P}_{\boldsymbol{f}} \cup \{\boldsymbol{f}(\boldsymbol{x})\}) - \mathrm{HV}(\mathcal{P}_{\boldsymbol{f}})],$$

where $\mathrm{HV}(\mathcal{P}_{\boldsymbol{f}})$ is the hypervolume of the Pareto front (relative to some reference point[3]) and $\mathbb{E}_{\boldsymbol{f}}$ is the expectation operator of (2.5) (with respect to the $l$ objectives, not the constraints).

The goal is to use constrained single-objective (non-Bayesian) gradient-based optimisation[4] to find

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \mathcal{U}} \alpha_{\mathrm{EHVI}}(\boldsymbol{x}).$$

Once found, this point can be evaluated using $\boldsymbol{f}$ and added to the dataset $\mathcal{D}$.

As mentioned before, we have both constraints on the function $\boldsymbol{f}$ and failure regions in the input space (that we wish to avoid). To do this, we can extend EHVI by multiplying it by a feasibility indicator $\Delta(\boldsymbol{x}) \in \{0, 1\}$, such that we obtain the Expected Constrained Hypervolume Improvement (ECHVI) function

$$\alpha_{\mathrm{ECHVI}}(\boldsymbol{x}) = \Delta(\boldsymbol{x})\alpha_{\mathrm{EHVI}}(\boldsymbol{x}).$$

The feasibility indicator $\Delta(\boldsymbol{x})$ returns 1 if the probability of feasibility is above some threshold in $[0, 1]$, otherwise it returns 0 (Gardner et al., 2014). The probability of feasibility is low if a point $\boldsymbol{x}$ is likely to violate any of the $m$ constraints or reside in a failure region. This means that a

point that is Pareto optimal but likely infeasible would yield an ECHVI value of zero and would therefore not maximise the acquisition function.

## 3. The poloidal field coil design problem

The PF coil design problem described here is concerned with identifying the set of PF coil positions that will optimise some aspects of both cost and performance of a STEP-like tokamak, subject to strict design and engineering requirements. In this section, we will describe the inputs, objectives, and constraints required to formulate the optimisation problem as well as the underlying STEP baseline design and the simulator required to calculate plasma equilibria. Throughout, we will be working within a cylindrical coordinate system $(R, \phi, Z)$ which denotes the major radius, the toroidal direction (into the page), and the height, respectively.

### 3.1. The STEP baseline design

We will be working with the initial PF coil setup and limiter geometry from the baseline STEP design presented by Hudoba et al. (2023a). The design is shown in Figure 3.1 and the information available to us from the baseline dataset are the:

○ names, centroid positions $(R^c, Z^c)$, and half width/heights $(\mathrm{d}R, \mathrm{d}Z)$ of the PF coils.
○ permissible zones for each PF coil (i.e. the region of the $RZ$-plane in which each coil can be placed without intersecting the TF coils, diagnostic systems, or other parts of the tokamak).
○ limiter contour that confines the plasma equilibrium[5].
○ strike plate locations (i.e. segments of the limiter in the inner and outer divertor where the legs of the plasma separatrix will strike).
○ separatrix of the plasma equilibrium, the X-points, and the strikepoints.
○ plasma pressure and toroidal magnetic field profiles required to solve for the equilibrium.

### 3.2. Optimisation problem

The mathematical formulation of the PF coil optimisation problem requires stacked scalar inputs, objectives, and constraints so that we can map a vector of PF coil positions to a vector of objective/constraint values.

#### 3.2.1. Input space

As can be seen in Figure 3.1, there are seven up-down symmetric[6] PF coil circuits each with their own $(R^c, Z^c)$ centroid
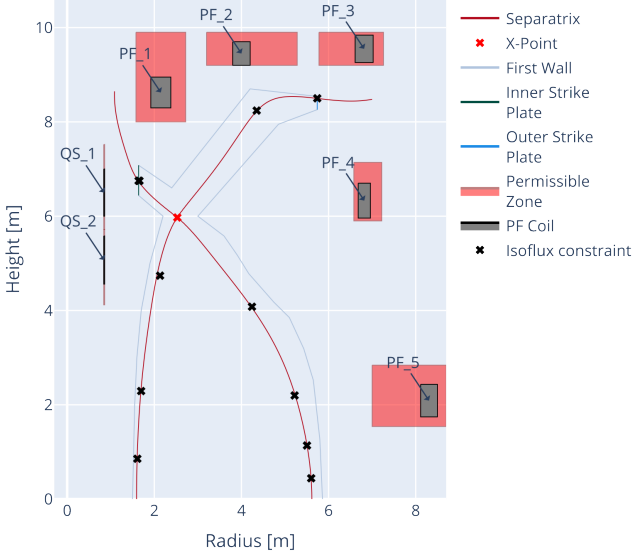
---

**Figure 3.1** The baseline setup from Hudoba et al. (2023a) in the $RZ$-plane (due to vertical symmetry about $Z = 0$ only the top half of the tokamak is shown). The separatrix (red) outlines the shape of the plasma core and the divertor legs which hit the inner (green) and outer (blue) strike plates. The initial location of the seven PF coils (grey rectangles) are displayed within their respective permissible zones (red rectangles). Note the absence of the central solenoid, which is not used in the flat-top phase of operation shown here. The isoflux constraints (black crosses) define locations at which the separatrix should pass through.

coordinate that is allowed to move freely such that no part of the coil leaves the permissible zone. The exceptions are the two $QS$ coils which are only able to move vertically. This results in a twelve dimensional input space for the optimisation problem: five pairs of $(R^c, Z^c)$ coordinates for the $PF$ coils and one $Z^c$ coordinate for each of the two $QS$ coils. We normalise each of the coordinates with respect to their own permissible zones so that we can work with the unit hypercube $[0, 1]^{12}$ as our input space. A more detailed explanation of the normalisation process can be found in Appendix A.

### 3.2.2. Objectives

In this problem, we consider two scalar objective functions that we wish to optimise with multi-objective BO—though we should note that nothing prevents us from adding more objectives.

The first objective is to *minimise* the volumetric sum of the PF coils. This is important as smaller coils require less physical material and therefore weigh less, making the fabrication, transportation, and installation process less arduous and costly. Recall, each PF coil is modelled as a rectangle in the $RZ$-plane and as an annulus in the $R\phi$-plane. The total volume of the seven PF coils (upper and lower components inclusive) can therefore be defined as

$$V = 8\pi \sum_{i=1}^{14} R_i^c \mathrm{d}R_i \mathrm{d}Z_i. \qquad (3.1)$$

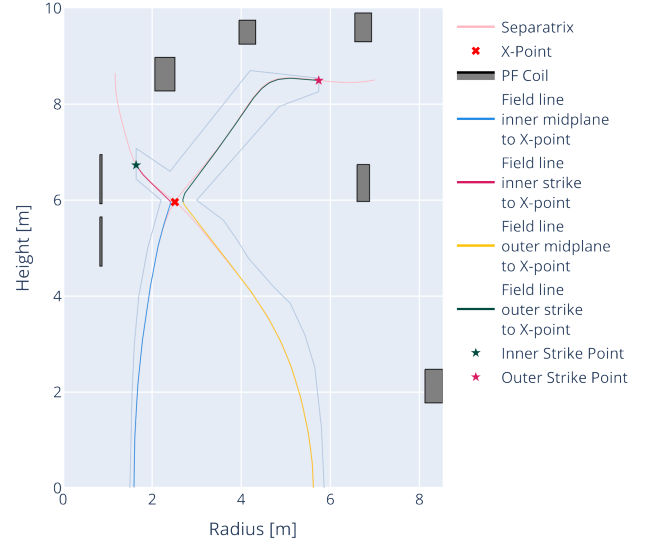where the coil index $i = 1, \ldots, 7$ for upper and $i = 8, \ldots, 14$



**Figure 3.2** The baseline setup from Figure 3.1, this time marked with objective and constraint quantities. Shown are the separatrix (light pink), the upper X-point (red cross), the O-point/magnetic axis (green circle), and the inner (green star) and outer (red star) strikepoints. Also shown are the flux surfaces traced out to calculate the ICL (blue plus red lines) and OCL (yellow plus green lines).

for lower coils. It is important to note that the PF coils do not change shape in the poloidal plane (i.e. cross-sectional areas are fixed, equal to $4\mathrm{d}R\mathrm{d}Z$) and so the volume of each coil will change only when its central radial position $R^c$ changes.

The second objective is to *maximise* the average of the inner connection length (ICL) and the outer connection length (OCL). The connection length is the distance traced out by a helical (i.e. moving both poloidally and toroidally) magnetic field line that starts at the inner (outer) edge of the last closed flux surface (LCFS) at the midplane and ends at the inner (outer) strikepoint (Albanese et al., 2023; Doyle et al., 2021). We refer to these inner and outer midplane points as the IMP and OMP, respectively. Larger connection lengths ensure that hot plasma leaving the core edge region will travel a further distance and therefore cool to more acceptable temperatures before hitting the strike plate. This is an important aspect of managing heat loads in the divertor region.

The ICL and OCL are calculated using a 3D field line tracing algorithm, i.e. by carrying out an integral over the length of the magnetic field line of interest (Kos et al., 2019). By parametrising the position vector $\boldsymbol{r}(\ell) = (R(\ell), \phi(\ell), Z(\ell))$ in terms of a length $\ell$, we can find the connection length $L$ by first solving the ordinary differential equations

$$\frac{\mathrm{d}\boldsymbol{r}(\ell)}{\mathrm{d}\ell} = \boldsymbol{B}(\boldsymbol{r}(\ell)) = \begin{cases} -\frac{1}{R}\frac{\partial\psi(R,Z)}{\partial Z}, \\ \frac{F(\psi(R,Z))}{R}, & \ell \in [0, L], \\ \frac{1}{R}\frac{\partial\psi(R,Z)}{\partial R}, \end{cases} \qquad (3.2)$$

with an initial condition $\boldsymbol{r}(0) = (R(0), \phi(0), Z(0))$. Here, $\boldsymbol{B}$ denotes the magnetic field, $\psi(R, Z)$ the scalar poloidal

magnetic flux, and $F$ the toroidal magnetic field profile. To do this, we use a fourth-order Runge-Kutta method (with appropriately chosen step size $\Delta\ell$) and integrate until some terminating condition is met (see below), recording the value of $L$ obtained (i.e. the total number of steps $\Delta\ell$).

In practice, however, the calculation is split into two stages. The first stage involves integrating from an initial location 3mm radially outside the IMP/OMP and ending at some very small distance away from the upper X-point. The second stage integrates between the inner/outer strikepoint and the point close to the upper X-point. The length of these individual sections is then combined to return the final connection length. The two flux surfaces traced out when calculating the ICL and OCL are visualised in Figure 3.2. These techniques reduce the likelihood of the integrator getting stuck at the exact X-point, travelling around the LCFS (instead of going up into a divertor), and from travelling into the wrong divertor.

### 3.2.3. Constraints

In addition to the objectives, we also have a number of design and engineering constraints that need to be satisfied so that each PF coil set considered in the BO loop produces an equilibrium with key targets that are similar to the baseline equilibrium and does not violate coil current limits. The bounds of the constraints are summarised in Table 3.1.

The first constraint is on the shape of the LCFS, which is defined as the contour of $(R, Z)$ points that pass through the X-point closest to the magnetic axis—recall Figure 3.1. We denote this region of the $RZ$-plane as $\Omega_p$ and quantify the difference between two different regions using

$$\eta\big(\Omega_p^1, \Omega_p^2\big) := \frac{\big|\Omega_p^1 \cup \Omega_p^2\big| - \big|\Omega_p^1 \cap \Omega_p^2\big|}{\big|\Omega_p^1\big| + \big|\Omega_p^2\big|} \in [0, 1],$$

where $|\cdot|$ denotes the cross-sectional area of a region in the poloidal plane (Bardsley et al., 2024). This parameter quantifies the ratio of the total non-overlapping areas and the sum of the two areas. Placing an upper limit on this ratio enables us to constrain the LCFS shape of the new equilibrium ($\Omega_p^2$) to be similar to that of the baseline ($\Omega_p^1$). This helps to ensure the new equilibrium has similar core performance to the baseline.

The second and third constraints place an upper limit on the distance between the strikepoints (i.e. where the separatrix first intersects the limiter geometry at some location[7]) and the centre of the strike plates. The bound is half of the length of the strike plate, with one constraint on each of the inner and outer strike plates.

The fourth constraint will place an upper limit on the distance between the two X-point locations when mirrored about $Z = 0$. This distance should be minimal in a double-null plasma scenario as considered here. See Figure 3.2 for the strikepoint and X-point locations.

| Constraint | Bound [unit] |
|---|---|
| LCFS area ratio | $\leqslant 0.012$ |
| Outer strike distance | $\leqslant 0.14$ [m] |
| Inner strike distance | $\leqslant 0.32$ [m] |
| X-point distance | $\leqslant 0.01$ [m] |
| Maximum current density | $\leqslant 100$ [MA/m$^2$] |

**Table 3.1** Constraint bounds enforced on the equilibria generated by FreeGS for a particular PF coil set.

The final constraint ensures the maximum current density

$$J_{\mathrm{max}} = \frac{1}{4} \max_{i \in [1...7]} \frac{I_i}{\mathrm{d}R_i \mathrm{d}Z_i},$$

in the PF coil set remains below the engineering limit defined in Nasr et al. (2024). Here, $I_i$ denotes the coil current and the denominator is the coil area. This limits stresses in the PF coil structures and helps avoid quench events—a sudden loss of superconductivity which can damage the coils (Coatanea-Gouachet et al., 2015).

### 3.2.4. Equilibrium simulator

In order to calculate the aforementioned objective functions and evaluate whether or not the constraints have been met, we need a simulator that is able to generate a plasma equilibrium using the STEP baseline design and a given PF coil set. For this we use *FreeGS*, a free-boundary static inverse equilibrium solver developed by Dudson (2023). FreeGS will return a plasma equilibrium (in terms of the poloidal flux) and the PF coil currents required to generate it. It uses an optimisation routine to identify the coil currents with respect to some constraints on the chosen plasma shape and a Picard iteration scheme to solve the free-boundary Grad-Shafranov problem (see Song et al. (2024) and Pentland et al. (2024)). The required inputs to solve the equilibrium problem are:

- the STEP baseline parameters and a PF coil set (permissible zones not required).
- two X-point locations, one at $(R^X, Z^X)$ and the other mirrored at $(R^X, -Z^X)$ (as we required an up-down symmetric double-null configuration like the baseline equilibrium).
- 23 isoflux constraints[8] that link poloidal flux values on the core plasma boundary to the X-points and the divertor regions.

Given these inputs, FreeGS will return the coil currents in the PF coils required to generate an equilibrium that (closely) matches the one provided in the baseline. From this equilibrium, we can then calculate the values of the objectives and the constraints. From time to time, however, the simulator may fail to converge on a physically "valid" equilibrium, returning spurious objective and constraint values. This could be for a number of reasons such as solver instability or a

---

[7]In rare cases, an equilibrium may have a separatrix that intersects the limiter multiple times on the same plate.

[8]These are constraints that ensure the poloidal flux $\psi(R, Z)$ at two different locations $(R_1, Z_1)$ and $(R_2, Z_2)$ are the same.
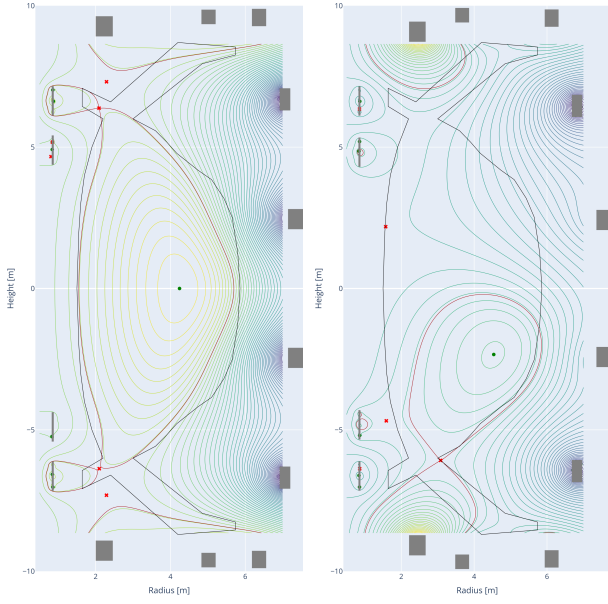
**Figure 3.3** Two examples of invalid equilibria returned by "converged" FreeGS simulations with different PF coil sets.

physically incompatible PF coil set. This requires care and will be discussed in the next section.

### 3.2.5. Failure regions

The equilibrium simulator will either (in rare cases) fail to converge or stop once the relative difference between the poloidal flux at successive iterations is below some tolerance threshold (returning an equilibrium solution). This stopping criteria does not, however, consider the physical validity of the equilibrium identified. In some cases, non-physical equilibria that do not satisfy the X-point and isoflux constraints may be returned. In other cases, we may have an equilibrium for which we either cannot calculate the objectives/constraints or which return spurious objective/constraint values. The regions of the input space for which non-physical (*invalid*) equilibria are returned (or if the simulator outright fails) will be referred to as *failure regions*.

In Figure 3.3, we illustrate two cases of invalid equilibria returned by FreeGS. The left panel depicts an equilibrium in which both X-points have formed inside the divertor regions, far from the desired locations in the core, resulting in the LCFS intersecting the limiter. This is problematic when calculating the ICL, as this calculation assumes the LCFS does not intersect the limiter geometry until hitting the strikeplate, resulting in an unfeasibly small ICL value. The right panel shows a single-null equilibrium with the LCFS, again, intersecting the limiter in the core region. Calculations for both the ICL/OCL and the strike distances are spurious in this case.

To mitigate these issues, we can classify (recall Section 2.3) whether an equilibrium is valid by checking the following conditions:

1. the X-points $(R^X, Z^X)$ and $(R^X, -Z^X)$ must be to within 10cm of the limiter boundary with $R^X \in [2.2, 3]$.
2. the LCFS does not intersect the limiter[9].

By actively avoiding sampling the PF coil sets where the simulator fails or produces such invalid equilibria (via the classifier), we can avoid wasting computational resources on solutions that do not provide any useful information to the BO loop.

## 4. Numerical experiments

In this section, we will perform the design optimisation of the PF coil set problem. The first experiment will use multi-objective BO to find several Pareto optimal PF coil sets that respect the engineering and design constraints in Section 3.2. We analyse two of the Pareto optimal in more detail, highlighting how the BO explores the solution space while respecting the trade-off between the objective functions. To further illustrate the data efficiency of the BO, we compare its performance against two other optimisation methods (simple Sobol sampling and a genetic algorithm) when using both identical and larger computational budgets.

To generate these results, we use the Trieste platform (Berkeley et al., 2024; Picheny et al., 2023) which provides the software implementations for Sobol sampling, acquisition functions, and Gaussian processes (via GPflow (Matthews et al., 2017)). Pygmo2 (Biscani and Izzo, 2020) provides the genetic algorithm which we will use for benchmarking. When evaluating the Sobol samples with FreeGS, we make use of the CSD3 HPC cluster (see Acknowledgements) and the Simvue platform (Lahiff et al., 2024) to monitor simulation progress and store the objective/constraint data.

### 4.1. Stand-alone BO

In this experiment, we will limit ourselves to 128 evaluations of $\boldsymbol{f}$: 64 Sobol samples to build the initial dataset and 64 sequential BO samples to intelligently explore the objective space and identify feasible optimal points.

In Table 4.1, we display the proportions of each sampling set that result in feasible, infeasible (violating one or more constraints), and failed (invalid) PF coil sets. We can see that only 10% of the Sobol samples provide feasible designs and that once the BO loop begins running, we accumulate a much larger proportion of feasible designs (with fewer failures). This shows that the GPs can accurately model the constraint responses and the acquisition function uses this to propose feasible samples. The drop in failure region samples likely results from a combination of explicit classifier intervention and the scarcity of optimal samples near these regions, making them less likely to be chosen by the acquisition function.

---

[9]There is now an extension to FreeGS—*FreeGSNKE* (Amorisco et al., 2024)—which ensures the core remains within the limiter. Use of this equilibrium solver would remove the need for this constraint, however, it was released following the completion of this work.
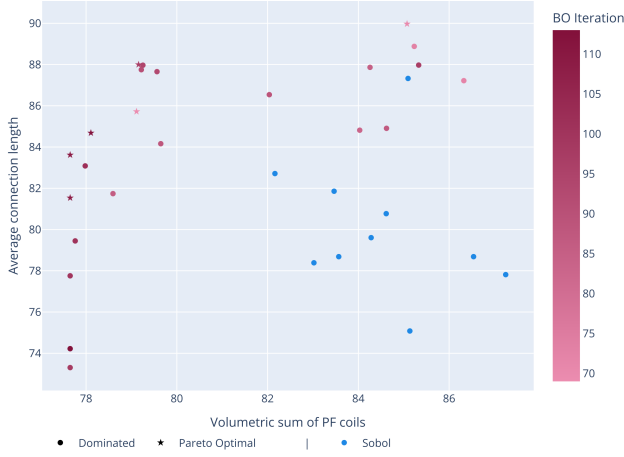
**Figure 4.1** Feasible solutions in the objective space for the 64 Sobol + 64 BO experiment. Shown are the Sobol solutions (blue) and the BO solutions (light pink to red). Pareto optimal solutions are denoted with a star and dominated solutions with a circle, with light pink to red indicating successive BO iterations.
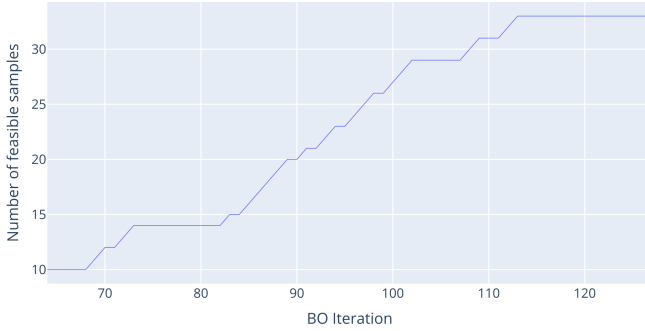


**Figure 4.2** The total number of feasible BO samples at the end of each BO iteration.

The BO loop successfully identifies the Pareto front (see Figure 4.1), capturing the trade-off between the volumetric sum (3.1) and the average connection length (recall (3.2)). Generally, the later BO iterations produce samples that dominate earlier samples, highlighting how BO learns from new data and exploits its new understanding of the functions to produce higher-quality samples.

During the initial BO iterations, the data is sparse, resulting in highly uncertain and inaccurate GPs. As a result, the exact location of the feasible regions is unclear and, as it explores for the first 20 iterations, the BO produces few feasible samples—see Figure 4.2. Following this initial exploration, 30 iterations of exploitation takes place, where BO reliably produces feasible samples (those seen in Figure 4.1).

| Method | Failure | Infeasible | Feasible | Total |
|--------|---------|------------|----------|-------|
| Sobol | 23 (36%) | 31 (48%) | 10 (16%) | 64 |
| BO | 4 (6%) | 37 (58%) | 23 (36%) | 64 |

**Table 4.1** The number (and percentage) of samples from each sampling method that lie in failure, infeasible, or feasible regions.
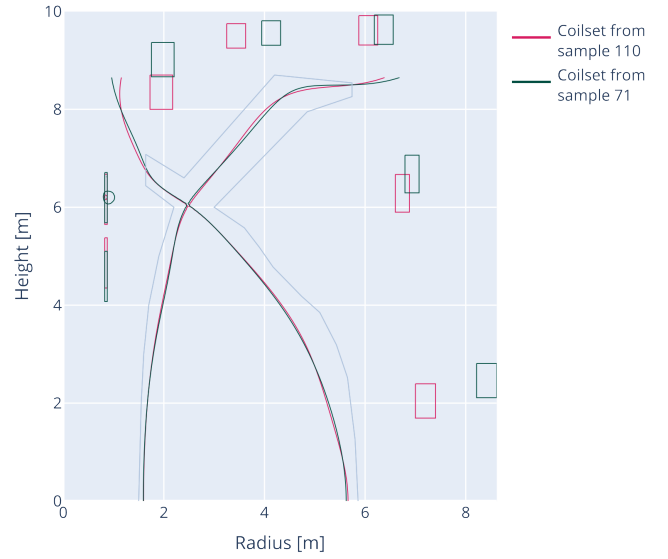


**Figure 4.3** Two Pareto optimal PF coil sets (and the separatrix of the equilibria they generate) returned from the 71st (green) and 110th (green) sampling of the simulator (corresponding to samples from the 7th and 46th BO iteration, respectively).

The final 20 iterations yield no feasible samples, indicating a return to exploration and suggesting there is little scope to find feasible optimal solutions around the current Pareto optimal points.

In Table 4.2, we display the objective/constraint values obtained from two of the Pareto optimal PF coil sets shown in Figure 4.3, with both samples obtained during the BO iterations. The 71st sample yields the the highest average connection length while the 110th sample yields the lowest volumetric sum (i.e. they are the two samples that optimise the marginals of the objective-space). The first three rows of the table show the objective quantities for both samples

| Objective/constraint | Sample 110 | Sample 71 |
|----------------------|------------|-----------|
| $V$ | 77.6 [m$^3$] | 85.1 [m$^3$] |
| ICL | 96.9 [m] | 108 [m] |
| OCL | 70.3 [m] | 71.8 [m] |
| LCFS area ratio | 0.0117 | 0.0103 |
| Outer strike distance | 0.0957 [m] | 0.0962 [m] |
| Inner strike distance | 0.0233 [m] | 0.0305 [m] |
| X-point distance | 0.00394 [m] | 0.000919 [m] |
| $J_{\max}$ | 82.4 [MA/m$^2$] | 98.8 [MA/m$^2$] |
| Inner $\langle B_p \rangle$ | 0.413 [T] | 0.373 [T] |
| Outer $\langle B_p \rangle$ | 0.537 [T] | 0.530 [T] |

**Table 4.2** The objectives and constraint values (to three significant figures) returned for two of the Pareto optimal solutions shown in Figure 4.3. Also shown are the inner and outer line-averaged poloidal magnetic field readings.

while the intermediate five rows show the constraint values. The constraints show that samples 110 and 71 are close to the constraint bounds for the LCFS shape and maximum coil current density, respectively. This could indicate that further optimisation of these samples (and other Pareto optimal samples) is not possible without violating the constraints, hence the lack of feasible samples in the final BO iterations above.

It is clear from Figure 4.3 that the 110th sample has a smaller volumetric sum because PF coils 2, 3, 4, and 5 are closer to the device centreline ($R = 0$). However, the difference in average connection length is less obvious because the separatrices look (qualitatively, at least) very similar. The lower connection length in sample 110 results from a higher poloidal field, causing particles travelling from the midplane into the divertors to move faster, decreasing the number of times (and thus the distance) they orbit the tokamak toroidally. This can be seen in the final two rows of Figure 4.3, which shows the line-averaged poloidal magnetic field $\langle B_p \rangle$ along the inner and outer connection length field lines, respectively.

## 4.2. Comparison between BO, Sobol sampling, and a genetic algorithm

Next, we compare the BO scheme against two other multi-objective optimisation methods. The first method we compare against uses quasi-Monte Carlo sampling via the Sobol method, essentially relying on random chance to sample feasible and optimal coil sets. The second method will use a genetic algorithm, specifically the 'Non-Dominated Sorting Genetic Algorithm II' (NSGA-II) (Deb et al., 2002). NSGA-II, like all genetic algorithms, operates on the principle that combining the inputs of well-performing individuals within a population can produce offspring (new samples) that inherit characteristics from its parents and therefore may perform similarly or better. Starting with an initial population (in our case, Sobol samples), the algorithm generates additional samples through iterative recombination and mutation (applying slight random changes to the inputs), therefore introducing variation to explore the solution space (Verma et al., 2021). Constraints are handled by penalising the objectives according to the number of violated constraints (Kuri-Morales and Gutiérrez-García, 2002). Similarly, the failure region is handled by returning large constants for the objectives, artificially making the sample appear very non-optimal.

We run two sets of experiments with each of these methods, the results of which are presented in Table 4.3. The first two (II and III) use the same computational budget as the BO experiment (I) from Section 4.1, while the second two (IV and V) have a budget 8× larger.

From the results, we can see that experiment I produces better samples than both II and III with a hypervolume at least 20% larger. Recall that a larger hypervolume indicates a feasible objective space with better trade-offs that are further from the (anti-optimal) reference point—the feasible solutions are shown in the objective space in Figure 4.4. The genetic algorithm finds the most feasible samples, outper-

| | Experiment | HV | Feasible |
|---|---|---|---|
| I | 64 Sobol + 64 BO | 1990.9 | 26% |
| II | 128 Sobol | 1626.6 | 18% |
| III | 8 Sobol + 120 NSGA-II | 1659.2 | 49% |
| IV | 1024 Sobol | 1711.5 | 13% |
| V | 32 Sobol + 992 NSGA-II | 1820.3 | 64% |

**Table 4.3** The hypervolume of the feasible solution set and the percentage of total samples taken that were feasible for each of the experiments run. Here, we compare the 64 Sobol + 64 BO experiment from Section 4.1 with pure Sobol sampling and the NSGA-II algorithm, each with the same number of samples (128). We also display Sobol and NSGA-II experiments that use 8× the number of samples (1024). All hypervolumes are calculated with respect to the same reference point.



**Figure 4.4** Feasible solutions in the objective space for experiments I (red), II (blue), and III (green) in Table 4.3. Pareto optimal solutions are denoted with a star and dominated solutions with a circle.

forming BO by nearly 2×. However, the hypervolume of experiment III indicates few of these feasible samples offer any improvement over even quasi-random samples. This illustrates how the BO performs significantly better than the Sobol sampling and the genetic algorithm at finding Pareto optimal PF coil sets. Genetic algorithms find the most feasible PF coil sets, however, all of the samples are of significantly lower quality than those from BO.

BO continues to outperform Sobol sampling and the genetic algorithm even when we increase their computational budgets to 1024 samples. While the hypervolumes returned in experiments IV and V are larger compared to those in II and III (as expected), they still cannot reach the level achieved by the BO (with 1/8th of the data). In Figure 4.5, we again see the majority of Pareto optimal samples coming from the BO with a few being found by the genetic algorithm, with BO finding the best samples in each objective (the samples that optimise the marginals of the objective space). The vast majority of samples taken by the alternative methods are, however, dominated by others from the BO.
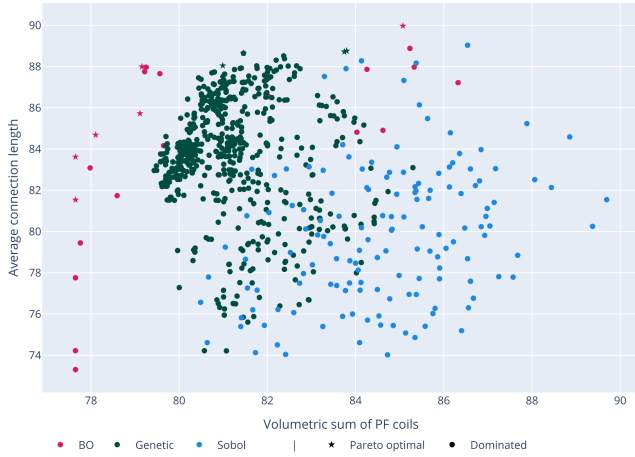
**Figure 4.5** Feasible solutions in the objective space for experiments I (red), IV (blue), and V (green) in Table 4.3. Pareto optimal solutions are denoted with a star and dominated solutions with a circle.

Again, genetic algorithms find the most feasible samples, however, they form a front that underperforms that of BO, particularly in the volumetric sum.

## 5. Discussion and outlook

In this paper, we have demonstrated that BO can successfully identify a set of Pareto optimal PF coil sets in a spherical tokamak. Using underlying probabilistic models, it learns the trade-off between the volume of the PF coil set (i.e. the financial cost) and the average connection length produced by the corresponding equilibrium state, simultaneously respecting several physical plasma and engineering constraints. Compared to some existing optimisation methods, quasi-Monte Carlo (Sobol) sampling and a genetic algorithm (NSGA-II), BO identifies better solutions while using a significantly smaller computational budget, highlighting its effectiveness and data efficiency. Overall, the successful application of BO to a complex tokamak design problem should reinforce its suitability for future fusion power plant design challenges, particularly given the increasing reliance on high-fidelity, high-runtime HPC codes where data efficiency is critical.

The relatively poor performance of the Sobol sampling is expected and can likely be attributed to its sparse quasi-uniform coverage of the sample space. While uniform coverage is good for exploring high dimensional spaces and training emulators (such as the one in our BO loop), the Sobol scheme lacks the ability to hone in on more desirable regions given it is forced to sample inputs within uniformly-spaced partitions of the space. NSGA-II outperforms Sobol sampling, especially when both are afforded moderately high computational budgets, however, has an underwhelming performance against BO. While it excels at finding feasible samples, NSGA-II fails to find samples dominant over BO, even with a significantly higher computational budget. This is likely because the genetic algorithm favours sampling feasible points instead of exploring towards the feasible boundary

and potentially finding a more optimal sample—the cost of infeasibility does not outweigh the reward of slight improvements in the objectives. It is possible more advanced treatments of the constraints (Long, 2014) would improve the genetic algorithm's performance and allow it to explore closer to the feasible boundary, however, that is beyond the scope of this work.

To increase the applicability and extend this BO framework to ongoing and future PF coil design projects, a number of avenues of future work can be considered. For example, incorporating additional objectives and constraints should be a trivial task and could be used to help find PF coil sets that further improve divertor performance (e.g. maximising flux expansion) or help vertically stabilise the core plasma. It could also be of interest to include the PF coil thicknesses (in the input space) so that designers could extract even more financial cost savings by reducing the total cross-sectional area of the PF coils. This is something the current framework could be easily adapted to support. Beyond the current framework, analysis could also be carried out to investigate whether the coil sets found for the flat-top phase of the plasma pulse remain optimal during the ramp-up and ramp-down phases.

## Data availability

The code scripts and data used in this paper will be made available in due course.

## Declarations

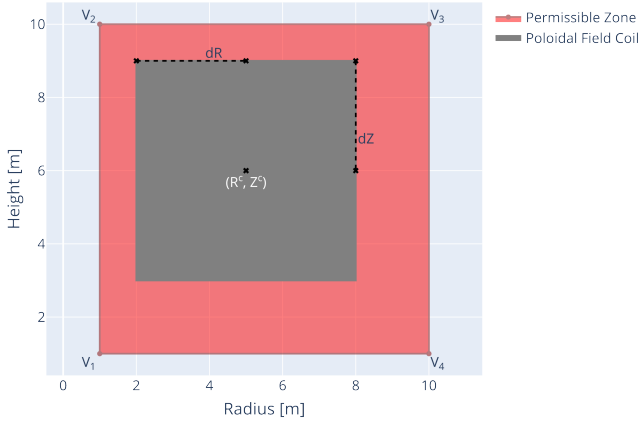The authors have no conflicts of interest to declare.

**Figure A.1** Illustration of a rectangular PF (dark grey), with its centroid and half-width/height marked, and its permissible zone (red), with corner vertices marked.

## A. Coil location normalisation

Here, we outline how to normalise the centroid coordinates of each PF coil with respect to its permissible zone. First, define the lower left and upper right corners of each permissible zone as $\boldsymbol{V}_1 = (R_{\min}, Z_{\min})$ and $\boldsymbol{V}_3 = (R_{\max}, Z_{\max})$, respectively. Given each coil must entirely reside within its permissible zone, we know that the centroid must remain within a half-thickness of the permissible zone:

$$(R^c, Z^c) \in [R_{\min} + \mathrm{d}R, R_{\max} - \mathrm{d}R] \times$$
$$[Z_{\min} + \mathrm{d}Z, Z_{\max} - \mathrm{d}Z].$$

We can then obtain the normalised centroid coordinates (with respect to the permissible zone) by defining

$$\widetilde{R^c} = \frac{R^c - (R_{\min} + \mathrm{d}R)}{(R_{\max} - \mathrm{d}R) - (R_{\min} + \mathrm{d}R)} \in [0, 1],$$

$$\widetilde{Z^c} = \frac{Z^c - (Z_{\min} + \mathrm{d}Z)}{(Z_{\max} - \mathrm{d}Z) - (Z_{\min} + \mathrm{d}Z)} \in [0, 1].$$

An illustration of a PF coil and its permissible zone are shown in Figure A.1.

## References

R. Albanese, A. G. Chiariello, L. E. Di Grazia, A. Iaiunese, R. Martone, M. Mattei, F. Villone, and P. Zumbolo. Three-dimensional evaluation of the connection lengths in a tokamak. *Fusion Engineering and Design*, 192:113622, 2023. doi:10.1016/j.fusengdes.2023.113622.

N. C. Amorisco, A. Agnello, G. Holt, M. Mars, J. Buchanan, and S. Pamela. FreeGSNKE: A python-based dynamic free-boundary toroidal plasma equilibrium solver. *Physics of Plasmas*, 31(4):042517, 2024. doi:10.1063/5.0188467.

H. Anand, O. Bardsley, D. Humphreys, M. Lennholm, A. Welander, Z. Xing, J. Barr, M. Walker, J. Mitchell, and H. Meyer. Modelling, design and simulation of plasma magnetic control for the spherical tokamak for energy production (step). *Fusion Engineering and Design*, 194:113724, 2023. doi:10.1016/j.fusengdes.2023.113724.

O. Bardsley, J. Baker, and C. Vincent. Decoupled magnetic control of spherical tokamak divertors via vacuum harmonic constraints. *Plasma Physics and Controlled Fusion*, 66(5): 055006, 2024. doi:10.1088/1361-6587/ad319d.

J. Berkeley, H. B. Moss, A. Artemev, S. Pascual-Diaz, U. Granta, H. Stojic, I. Couckuyt, J. Qing, N. Loka, A. Paleyes, S. W. Ober, A. Goodall, K. Ghani, and V. Picheny. Trieste (4.2.2), 2024. URL https://github.com/secondmind-labs/trieste.

F. Biscani and D. Izzo. A parallel global multiobjective framework for optimization: pagmo. *Journal of Open Source Software*, 5 (53):2338, 2020. doi:10.21105/joss.02338.

T. Brown, A. Cioba, and I. Bogunovic. Sample-efficient Bayesian optimisation using known invariances, 2024a. arXiv:2410.16972.

T. Brown, S. Marsden, V. Gopakumar, A. Terenin, H. Ge, and F. Casson. Multi-objective Bayesian optimization for design of Pareto-optimal current drive profiles in STEP. *IEEE Transactions on Plasma Science*, pages 1–6, 2024b. doi:10.1109/TPS.2024.3382775.

R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. doi:10.1137/0916069.

I. Chapman, S. Cowley, and H. Wilson. The spherical tokamak for energy production: theme issue introduction, 2024.

M. Coatanea-Gouachet, D. Carrillo, S. Lee, and F. Rodríguez-Mateos. Electromagnetic quench detection in ITER superconducting magnet systems. *IEEE Transactions on Applied Superconductivity*, 25(3):1–7, 2015. doi:10.1109/TASC.2015.2390296.

M. Coleman and S. McIntosh. BLUEPRINT: a novel approach to fusion reactor design. *Fusion Engineering and Design*, 139: 26–38, 2019. doi:10.1016/j.fusengdes.2018.12.036.

M. Coleman and S. McIntosh. The design and optimisation of tokamak poloidal field systems in the BLUEPRINT framework. *Fusion Engineering and Design*, 154:111544, 2020. doi:10.1016/j.fusengdes.2020.111544.

M. Coleman, J. E. Cook, F. Franza, I. A. Maione, S. McIntosh, J. Morris, D. Short, A. I. Blair, M. Bluteau, H. Brooks, I. Chiang, S. Desai, M. Foord, O. Funk, G. A. Graham, J. Hagues, L. Humphrey, M. Johnson, S. Kahn, C. MacMackin, S. Mason, H. Saunders, D. Vaccaro, and O. Wong. BLUEMIRA, 2025. URL https://github.com/Fusion-Power-Plant-Framework/bluemira.

S. Daulton, M. Balandat, and E. Bakshy. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems*, 33:9851–9864, 2020.

A. Davis, C. Waldon, S. I. Muldrew, B. S. Patel, P. Verrier, T. R. Barrett, and G. A. Politis. Digital: accelerating the pathway. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 382(2280): 20230411, 2024. doi:10.1098/rsta.2023.0411.

K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002. doi:10.1109/4235.996017.

S. J. Doyle, D. Lopez-Aires, A. Mancini, M. Agredano-Torres, J. Garcia-Sanchez, J. Segado-Fernández, J. Ayllon-Guerola, M. Garcia-Muñoz, E. Viezzer, C. Soria-Hoyo, et al. Magnetic equilibrium design for the SMART tokamak. *Fusion Engineering and Design*, 171:112706, 2021. doi:10.1016/j.fusengdes.2021.112706.

B. Dudson. FreeGS: Free-boundary Grad–Shafranov solver (0.8.0), 2023. URL https://github.com/freegs-plasma/freegs.

F. Franza, L. Boccaccini, E. Fable, I. Landman, I. Maione, S. Petschanyi, R. Stieglitz, and H. Zohm. Mira: a multi-physics approach to designing a fusion power plant. *Nuclear Fusion*, 62(7):076042, 2022. doi:10.1088/1741-4326/ac6433.

J. P. Freidberg. *Plasma physics and fusion energy*. Cambridge University Press, 2008. doi:10.1017/CBO9780511755705.

J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham. Bayesian optimization with inequality constraints. In *International Conference on Machine Learning*, volume 2014 of *ICML'14*, pages 937–945. JMLR.org, 2014. doi:10.5555/3044805.304499.

R. Garnett. *Bayesian Optimization*. Cambridge University Press, 2023. doi:10.1017/9781108348973.

A. Giuliani. Direct stellarator coil design using global optimization: application to a comprehensive exploration of quasi-axisymmetric devices. *Journal of Plasma Physics*, 90(3): 905900303, 2024. doi:10.1017/S0022377824000412.

T. R. Hoens and N. V. Chawla. *Imbalanced Datasets: From Sampling to Classifiers*, chapter 3, pages 43–59. John Wiley & Sons, Ltd, 2013. ISBN 9781118646106. doi:https://doi.org/10.1002/9781118646106.ch3. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118646106.ch3.

W. Hornsby, A. Gray, J. Buchanan, B. Patel, D. Kennedy, F. Casson, C. Roach, M. Lykkegaard, H. Nguyen, N. Papadimas, et al. Gaussian process regression models for the properties of micro-tearing modes in spherical tokamaks. *Physics of Plasmas*, 31 (1), 2024. doi:10.1063/5.0174478.

A. Hudoba, G. Cunningham, S. Bakes, and STEP Team and others. Magnetic equilibrium optimisation and divertor integration in spherical tokamak reactors. *Fusion Engineering and Design*, 191:113704, 2023a. doi:10.1016/j.fusengdes.2023.113704.

A. Hudoba, S. Newton, G. Voss, G. Cunningham, and S. Henderson. Divertor optimisation and power handling in spherical tokamak reactors. *Nuclear Materials and Energy*, 35:101410, 2023b. doi:10.1016/j.nme.2023.101410.

L. R. Humphrey, A. J. Dubas, L. C. Fletcher, and A. Davis. Machine learning techniques for sequential learning engineering design optimisation. *Plasma Physics and Controlled Fusion*, 66(2):025002, 2023. doi:10.1088/1361-6587/ad11fb.

H. Ishibashi, M. Karasuyama, I. Takeuchi, and H. Hino. A stopping criterion for Bayesian optimization by the gap of expected minimum simple regrets. In F. Ruiz, J. Dy, and J.-W. van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 6463–6497. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/ishibashi23a.html.

A. E. Järvinen, T. Fülöp, E. Hirvijoki, M. Hoppe, A. Kit, J. Åström, and J. Contributors. Bayesian approach for validation of runaway electron simulations. *Journal of Plasma Physics*, 88(6):905880612, 2022. doi:10.1017/S0022377822001210.

R. Jorge, A. Giuliani, and J. Loizu. Simplified and flexible coils for stellarators using single-stage optimization. *Physics of Plasmas*, 31(11):112501, 2024. doi:10.1063/5.0226688.

A. A. Kaptanoglu, M. Landreman, and M. C. Zarnstorff. Optimization of passive superconductors for shaping stellarator magnetic fields, 2025. arXiv:2501.12468.

L. Kos, R. Pitts, G. Simič, M. Brank, H. Anand, and W. Arter. SMITER: A field-line tracing environment for ITER. *Fusion Engineering and Design*, 146:1796–1800, 2019. doi:10.1016/j.fusengdes.2019.03.037.

A. F. Kuri-Morales and J. Gutiérrez-García. Penalty function methods for constrained optimization with genetic algorithms: A statistical analysis. In *Mexican international conference on artificial intelligence*, pages 108–117. Springer, 2002. doi:10.3390/mca10010045.

A. Lahiff, A. Abraham, V. Gopakumar, K. Zarebski, M. Field, and J. Panayis. Simvue (1.1.4), 2024. URL https://github.com/simvue-io/client.

B. Lim, F. Simon, Y. Ilyin, C. Gung, J. Smith, Y. Hsu, C. Luongo, C. Jong, and N. Mitchell. Design of the iter pf coils. *IEEE transactions on applied superconductivity*, 21(3):1918–1921, 2010. doi:10.1109/TASC.2010.2092732.

Q. Long. A constraint handling technique for constrained multi-objective genetic algorithm. *Swarm and Evolutionary Computation*, 15:66–79, 2014. doi:10.1016/j.swevo.2013.12.002.

A. G. d. G. Matthews, M. van der Wilk, T. Nickson, K. Fujii, A. Boukouvalas, P. León-Villagrá, Z. Ghahramani, and J. Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18 (40):1–6, apr 2017. doi:10.5555/3122009.3122049. URL http://jmlr.org/papers/v18/16-537.html.

V. Mehta, J. Barr, J. Abbate, M. D. Boyer, I. Char, W. Neiswanger, E. Kolemen, and J. Schneider. Automated experimental design

of safe rampdowns via probabilistic machine learning. *Nuclear Fusion*, 64(4):046014, 2024. doi:10.1088/1741-4326/ad22f5.

O. Meneghini, T. Slendebroek, B. C. Lyons, K. McLaughlin, J. McClenaghan, L. Stagner, J. Harvey, T. F. Neiser, A. Ghiozzi, G. Dose, J. Guterl, A. Zalzali, T. Cote, N. Shi, D. Weisberg, S. P. Smith, B. A. Grierson, and J. Candy. FUSE (Fusion Synthesis Engine): A next generation framework for integrated design of fusion pilot plants, 2024. arXiv:2409.05894.

J. Morris, S. Muldrew, P. Knight, M. Kovari, A. Pearce, S. Kahn, J. Maddock, T. Nunn, R. Chapman, C. Swanson, H. Lux, J. Foster, K. Ellis, M. Kumar, G. Turkington, J. Lion, S. Chislett-McDonald, A. Brown, J. Edwards, C. Ashe, E. Fable, K. Zarebski, C. Griesel, R. Kemp, S. Gubbins, S. Pickering, J. Matthews, G. Graham, C. Mould, T. Miller, S. Gadgil, D. Short, J. Cook, and P. Lloyd. PROCESS, Mar. 2024. URL https://github.com/ukaea/PROCESS.

S. I. Muldrew, H. Lux, G. Cunningham, T. C. Hender, S. Kahn, P. J. Knight, B. Patel, G. M. Voss, and H. R. Wilson. "PROCESS": Systems studies of spherical tokamaks. *Fusion engineering and design*, 154:111530, 2020. doi:10.1016/j.fusengdes.2020.111530.

S. I. Muldrew, C. Harrington, J. Keep, C. Waldon, C. Ashe, R. Chapman, C. Griesel, A. J. Pearce, F. Casson, S. P. Marsden, et al. Conceptual design workflow for the STEP prototype powerplant. *Fusion Engineering and Design*, 201:114238, 2024. doi:10.1016/j.fusengdes.2024.114238.

E. Nasr, S. C. Wimbush, P. Noonan, P. Harris, R. Gowland, and A. Petrov. The magnetic cage. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 382(2280):20230407, 2024. doi:10.1098/rsta.2023.0407.

T. Nunn, V. Gopakumar, and S. Kahn. Shaping of magnetic field coils in fusion reactors using Bayesian optimisation, 2023. arXiv:2310.01455 NeurIPS 2022 Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems.

R. K. Pandit and D. Infield. Comparative analysis of Gaussian process power curve models based on different stationary covariance functions for the purpose of improving model accuracy. *Renewable Energy*, 140:190–202, 2019. doi:10.1016/j.renene.2019.03.047.

K. Pentland, N. C. Amorisco, O. El-Zobaidi, S. Etches, A. Agnello, G. K. Holt, A. Ross, C. Vincent, J. Buchanan, S. Pamela, G. McArdle, L. Kogan, and G. Cunningham. Validation of the static forward Grad–Shafranov equilibrium solvers in FreeGSNKE and Fiesta using EFIT++ reconstructions from MAST-U. *Physica Scripta*, 2024. doi:10.1088/1402-4896/ada192.

V. Picheny, J. Berkeley, H. B. Moss, H. Stojic, U. Granta, S. W. Ober, A. Artemev, K. Ghani, A. Goodall, A. Paleyes, S. Vakili, S. Pascual-Diaz, S. Markou, J. Qing, N. R. B. S. Loka, and I. Couckuyt. Trieste: Efficiently exploring the depths of black-box functions with TensorFlow, 2023. arXiv:2302.08436.

I. Pusztai, I. Ekmark, H. Bergström, P. Halldestam, P. Jansson, M. Hoppe, O. Vallhagen, and T. Fülöp. Bayesian optimization of massive material injection for disruption mitigation in tokamaks. *Journal of Plasma Physics*, 89(2):905890204, 2023. doi:10.1017/S0022377823000193.

M. Romanelli, G. Corrigan, V. Parail, S. Wiesen, R. Ambrosino, P. D. S. A. Belo, L. Garzotti, D. Harting, F. Koechl, T. Koskela, et al. JINTRAC: a system of codes for integrated simulation of tokamak scenarios. *Plasma and Fusion research*, 9:3403023–3403023, 2014. doi:10.1585/pfr.9.3403023.

J. A. Romero and J. Svensson. Optimization of out-vessel magnetic diagnostics for plasma boundary reconstruction in tokamaks. *Nuclear Fusion*, 53(3):033009, 2013. doi:10.1088/0029-5515/53/3/033009.

I. Sobol. The distribution of points in a cube and the accurate evaluation of integrals (in Russian) zh. *Vychisl. Mat. i Mater. Phys*, 7:784–802, 1967. doi:10.1016/0041-5553(67)90144-9.

X. Song, B. Leard, Z. Wang, S. T. Paruchuri, T. Rafiq, and E. Schuster. Control-oriented free-boundary equilibrium solver for tokamaks. *Plasma*, 7(4):842–857, 2024. doi:10.3390/plasma7040045.

I. G. Stewart, R. S. Granetz, C. E. Myers, C. Paz-Soldan, R. Sweeney, C. J. Hansen, D. T. Garnier, D. J. Battaglia,

A. J. Creely, and M. L. Reinke. Optimization of the equilibrium magnetic sensor set for the SPARC tokamak. *Nuclear Fusion*, 63(12):126014, 2023. doi:10.1088/1741-4326/acf600.

E. Tholerus, F. Casson, S. Marsden, T. Wilson, D. Brunetti, P. Fox, S. Freethy, T. Hender, S. Henderson, A. Hudoba, K. Kirov, F. Koechl, H. Meyer, S. Muldrew, C. Olde, B. Patel, C. Roach, S. Saarelma, G. Xia, and the STEP team. Flat-top plasma operational space of the STEP power plant. *Nuclear Fusion*, 64(10):106030, aug 2024. doi:10.1088/1741-4326/ad6ea2.

D. Tran, R. Ranganath, and D. Blei. The variational gaussian process. *arXiv preprint arXiv:1511.06499*, 2015. doi:10.48550/arXiv.1511.06499. Appears in International Conference on Learning Representations, 2016.

S. Verma, M. Pant, and V. Snasel. A comprehensive review on NSGA-II for multi-objective combinatorial optimization problems. *IEEE access*, 9:57757–57791, 2021. doi:10.1109/ACCESS.2021.3070634.

J. Wesson and D. J. Campbell. *Tokamaks*, volume 149. Oxford university press, 2011.

C. K. Williams and D. Barber. Bayesian classification with gaussian processes. *IEEE Transactions on pattern analysis and machine intelligence*, 20(12):1342–1351, 1998. doi:10.1109/34.735807.

C. K. Williams and C. E. Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006. doi:10.7551/mitpress/3206.001.0001.

K. Yang, M. Emmerich, A. Deutz, and T. Bäck. Multi-objective Bayesian global optimization using expected hypervolume improvement gradient. *Swarm and evolutionary computation*, 44: 945–956, 2019. doi:10.1016/j.swevo.2018.10.007.