Trade-Offs in Bayesian Active Learning for Feasible Region Identification

Ioana Nikova[1,2*], Sebastian Rojas Gonzalez[2], Tom Dhaene[2], Ivo Couckuyt[2]

[1*]Siemens Industry Software, Leuven, Belgium.
[2]Ghent University - imec, Ghent, Belgium.

*Corresponding author(s). E-mail(s): ioana.nikova@siemens.com;
Contributing authors: sebastian.rojasgonzalez@ugent.be;
tom.dhaene@ugent.be; ivo.couckuyt@ugent.be;

Abstract

Typical engineering design problems, such as designing an aeroplane engine or testing an automated driving system, often involve multiple design constraints that define feasible solutions in the design space. Modern data-driven approaches allow for the effective characterisation of the (corresponding) feasible region(s), often by exploring the trade-off in regions of the design space with high expected performance and high uncertainty. Bayesian active learning is a data-efficient method that iteratively learns a surrogate model based on limited input-output data. Acquisition functions select samples based on a trade-off between exploration of the design space and exploitation of the feasible region. In this work, we consider this trade-off as a bi-objective maximization problem and show that existing acquisition functions choose samples on this Pareto front. We introduce two novel acquisition functions based on multi-objective scalarization methods for identifying the feasible region. The acquisition functions are compared against the state-of-the-art on several engineering benchmarks, as well as for testing an automated driving system. The results show that the novel acquisition methods are generally at least as effective as the state-of-the-art, while they select more feasible designs than boundary-focused acquisition functions.

Keywords: active learning, machine learning, design space exploration, feasible designs

1 Introduction

Data-driven design space exploration (DSE) has shown remarkable performance in modern design and optimization tasks (Forrester, Sobester, & Keane, 2008), for example in additive manufacturing (Flores Ituarte, Panicker, Nagarajan, Coatanea, & Rosen, 2023; Xiong et al., 2019; Young, Vondrasek, & Czabaj, 2025), automated driving systems (ADS) such as an Automatic Emergency Braking (AEB) system in a vehicle (Ariyanto, Haryadi, Munadi, Ismail, & Hendra, 2018), and fracture mechanics (Carrara, De Lorenzis, Stainier, & Ortiz, 2020). More specifically, engineers are often confronted with problems such as determining the optimal parameters of a specific part (e.g., length, width, angle, etc.), testing numerous scenarios to determine safety levels of operation, or simply minimizing the total production cost.

While the goal of the decision-maker is often to optimize these problems, learning about the feasibility of the design space is also an important task for engineers (Azzimonti, Ginsbourger, Chevalier, Bect, & Richet, 2021; Gotovos, Casati, Hitz, & Krause, 2013; Knudde, Couckuyt, Shintani, & Dhaene, 2019; Qing, Knudde, Couckuyt, Dhaene, & Shintani, 2020), especially when the design requirements and constraints are numerous, can change or are highly uncertain. For instance, an engineer wants to test an AEB system to see which scenarios result in the safe, thus feasible, behaviour of mitigating a collision between vehicles. There are an infinite number of real-world scenarios in which a vehicle should operate, and thus the system should be tested for (e.g., driving in a city centre versus on a highway and in different countries). Furthermore, in vehicle dynamics (Sobek II, 1996) engineers make critical early decisions about hardpoint locations, which heavily influence chassis design. At this stage, the design space is often constrained, and the focus is on identifying a layout that meets handling metrics rather than committing to an optimal design prematurely. Exploring the feasible region allows flexibility to adapt later when structural analyses may reveal stress-related issues in the initial design. Thus, testing these systems for each possible design is theoretically and practically infeasible, due to cost, time, and computational limits, so there is a need for more intelligent ways of choosing which simulations to run.

A typical way of gaining insight into the feasibility of the problem is using a space-filling Design of Experiments (DoE) like a Latin Hypercube (Park, 1994) or a quasi-random sequence (Lemieux, 2009). However, if the dimensionality of the problem is high, or the simulation is computationally expensive, running a space-filling DoE will likely fail in covering the entire feasible space (Santner et al., 2018). Our interest is not to find an optimum but to identify the feasible region, i.e., the region in the design space in which all designs satisfy the requirements. Note that the problem constraints are often analytically intractable; thus, to understand the feasibility of the problems, data-driven DSE techniques specifically designed for feasible region identification (FRI) are needed.

Bayesian active learning (AL) is a data-driven adaptive sampling approach that can be used for FRI. AL is a model-based methodology that extends the data set intelligently and iteratively (Cohn, Ghahramani, & Jordan, 1996). In the case of Bayesian AL, the surrogate is a probabilistic model that quantifies model uncertainty by returning a predictive distribution on the learned function. The most widely used

probabilistic model for Bayesian AL is a *Gaussian process* (Rasmussen & Williams, 2008). Note that, if we are interested in finding the optimum of a function, the iterative approach is called Bayesian optimization (BO) (Frazier, 2018). The samples are selected according to an acquisition function, where the next sampling point is the maximizer of this function. This function balances a trade-off between exploring regions that can improve the model's accuracy by sampling where the predicted uncertainty is high and exploiting regions with high performance in terms of feasibility.

Recently in the field of BO (see e.g., De Ath, Everson, Rahat, and Fieldsend 2021; Leite Richardson, De Ath, and Chugh 2024; Rojas Gonzalez, Branke, and Van Nieuwenhuyse 2025) researchers have looked at the trade-off between model prediction and uncertainty in well-known acquisition functions. In this work, we follow this perspective to approach acquisition functions in the field of FRI, where our objective is to learn the constraint functions that limit the design possibilities. We look at common AL acquisition functions specifically designed for FRI, and study whether their maximizer lies on the Pareto front (i.e., the optimal trade-off) of the exploration and exploitation objectives. Furthermore, we propose a new type of scalarization-based (Miettinen & Mäkelä, 2002) acquisition functions that explicitly balance said trade-off. More specifically, our contributions are as follows:

- We break down the building blocks of several state-of-the-art FRI acquisition functions to the fundamental measures for uncertainty exploration and feasibility exploitation.
- We investigate the exploration-exploitation trade-off front. This offers a unique comparison between the acquisition functions and gives more insight into the sample selection strategy of each acquisition function.
- We propose two novel scalarization-based acquisition functions for FRI, which are explicitly designed to increase the quality and quantity of feasible designs in the sampling set (as opposed to traditional boundary-based acquisition functions).
- We emphasize the broad range of use cases that can benefit from these techniques in an extensive empirical study on multiple real-world engineering examples. These include the design of two cantilever beam problems, an automated driving system (ADS) testing problem, a process flow sheeting problem, a spring design, and the design of a speed reducer in an aeroplane engine.

The rest of the paper is organized as follows. We start by introducing related work in Section 2, which covers a wide range of acquisition function approaches. Section 3 defines the constrained problem setting we consider for FRI. Next, Section 4 provides more background information. In Section 5 we focus on the considered acquisition functions. After, we have a closer look at the exploration-exploitation trade-off from the bi-objective optimization perspective in Section 6. This leads us to the introduction of the scalarization-based acquisition functions in Section 6.1. The engineering problems that are used for the benchmark are highlighted in Section 7, followed by a discussion of the results in Section 8.

2 Related Work

In the broad range of contributions to the field of FRI, we predominantly focus on research about acquisition function design. Acquisition functions can be classified according to different criteria. While there exist model-independent strategies that exploit advanced discretization techniques (e.g., Singh, van der Herten, Deschrijver, Couckuyt, and Dhaene 2017), most often a surrogate model is required for the computation of the acquisition function. The model can either be a classifier (Houlsby, Huszár, Ghahramani, & Lengyel, 2011), in which case the boundary between one (or more) classes needs to be identified, or a regression to learn the continuous constraints (Knudde et al., 2019). Note that classification leads to loss of information about the decision boundary, that is captured by a regression model.

Acquisition functions can be constructed in a *greedy* manner where each step gives immediate improvement (see e.g., Rahat and Wood 2020; Ranjan, Bingham, and Michailidis 2008), or *look-ahead*, in which case future acquisitions are taken into consideration (see e.g., Letham, Guan, Tymms, Bakshy, and Shvartsman 2022). There are two main challenges with look-ahead acquisition functions. Firstly, there is a significant part of these functions that are based on integral sampling. Stepwise Uncertainty Reduction (SUR) (Chevalier et al., 2014) and entropy reduction (Marques, Lam, & Willcox, 2018) methods typically rely on integration approximation. Hence, these acquisition functions are computationally more complex, usually not scalable to high dimensions, and require a set of integral points provided by the user. Secondly, as future states are approximated, an additional level of uncertainty is introduced, which can lead to reduced effectiveness.

Although acquisition functions are intended to balance both exploration and exploitation, the approach to achieving this balance differs. In SUR methods for example, the choice of the uncertainty measure guides the search (Chevalier et al., 2014; Picheny, Ginsbourger, Roustant, Haftka, & Kim, 2010); indeed, as the name suggests, SUR methods focus on uncertainty measures for exploration, and aim to learn a contour, i.e., boundary. More recently in Booth, Renganathan, and Gramacy (2025), the trade-off between entropy and (predictive) uncertainty is studied, where the entropy is seen as exploitative of the boundary region. However, as we discuss further in Section 4.3, entropy can be reduced to a function of model uncertainty, making it inherently exploratory. While these acquisition functions demonstrate competitive performance, they do not explicitly address flexibility or changes in the trade-off focus – an aspect that is often desirable in practical engineering applications. This aspect is incorporated in the scalarization-based acquisition functions we propose in Section 6.

Another difference among acquisition functions lies in their sampling strategy. The vast majority focuses on sampling near the boundary of the feasible region, i.e., locating the contours of the feasible region (see e.g., Marques et al. 2018). We want to address this gap because engineers often emphasize the importance of populating the core of the feasible region rather than its periphery. For example, in the automotive industry, CAD robustness studies ensure designs remain feasible by identifying valid ranges for the interdependent geometric variables. This approach prevents error-prone designs, guides search algorithms to explore only feasible solutions, and improves design efficiency and accuracy. Thus, we focus on greedy approaches that rely on

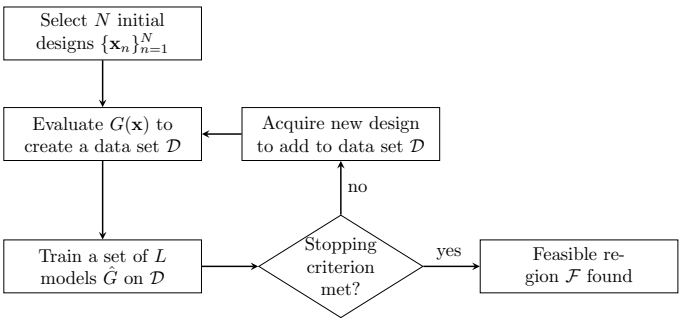


Fig. 1 Constrained problems and their feasible region can be learned with Bayesian active learning. Starting with an initial set, new designs are acquired and added to the data set. Models of the constraints are trained each iteration until the stopping criterion has been met.

regression models, like the ones proposed by Bichon, Eldred, Swiler, Mahadevan, and McFarland (2008); Kaintura et al. (2018); Knudde et al. (2019); Rahat and Wood (2020); Ranjan et al. (2008). We examine these in more detail in Section 5, and in Section 6 we propose to meet the engineers' request to populate the feasible region more in a multi-objective fashion.

3 Problem Definition

Consider an engineering problem with L constraints in a design space $\mathcal{X} \in \mathbb{R}^d$. The constrained problem $G(\mathbf{x})$ with design vector $\mathbf{x}$ can then be defined as:

$$G(\mathbf{x}) = (g_1(\mathbf{x}), \ldots, g_L(\mathbf{x}))^\top \leq \mathbf{t}, \text{with } \mathbf{t} = (t_1, \ldots, t_L)^\top, \tag{1}$$

where, $g_l : \mathbb{R}^d \to \mathbb{R}$ is the l-th constraint function with t_l its threshold for feasibility. In this study, for simplicity, $\mathbf{t}$ is chosen to be $(0, \ldots, 0)^\top$. Note that any inequality constraint can be converted to this form and that equality constraints can also be converted to two inequality constraints with a small constant ϵ.

Each constraint function $g_l(\mathbf{x})$ separates the design space in a feasible $\mathcal{F}_l \subseteq \mathcal{X}$ and infeasible $\mathcal{I}_l = \mathcal{X} \setminus \mathcal{F}_l$ space. Thus, the feasible space of the problem $G(\mathbf{x})$ is at the intersection of the feasible sets for all constraints: $\mathcal{F} = \bigcap_{l=1}^L \mathcal{F}_l$. Consequently, the infeasible set for the problem becomes $\mathcal{I} = \bigcup_{l=1}^L \mathcal{I}_l$.

Figure 1 illustrates how AL can be used for identifying the feasible region $\mathcal{F}$ of the problem $G(\mathbf{x})$. The first step is to select an initial design set $\{\mathbf{x}_n\}_{n=1}^N$. Often a space-filling design set is taken. Fuhg, Fau, and Nackenhorst (2021) mention the trade-off between a small set, where the surrogate models are lacking in knowledge, and a large set with high computational cost. Although the choice for the initial DoE size can potentially influence the performance of the surrogates, in practice the number of points is based on recommendations in the literature based on empirical evidence (e.g., $N = 11d - 1$ by Jones, Schonlau, and Welch (1998)). The constraints are then evaluated on these N initial points; for the l-th constraint function the data set is defined as $\mathcal{D}_l = \{(\mathbf{x}_n, g_l(\mathbf{x}_n))\}_{n=1}^N$ and the full data set is $\mathcal{D} = \{(\mathbf{x}_n, G(\mathbf{x}_n))\}_{n=1}^N$.

Subsequently, a surrogate model is trained for each constraint. In line with the AL literature, in this work we use Gaussian process (GP) models as surrogates. Therefore, in a problem with L constraints, L GP regression models are trained and $\mathcal{D}_l$ is the training data for the l-th model. We denote the set of L models as $\hat{G} = \{\hat{g}_1, \ldots, \hat{g}_L\}$.

After the models are trained, there is a check whether the stopping criterion has been satisfied. There exist various stopping criteria in the literature, which can be roughly divided into four categories (Fuhg et al., 2021): (i) a performance metric of the trained models is used, (ii) a certain budget of simulation time is defined, (iii) the number of maximum designs is constrained, or (iv) the relative correction between consecutive iterations is used. In this paper, we use the number of designs as the stopping criterion; this choice is intuitive in our context, as the simulation budget often determines the extent of the analysis.

In case the stopping criterion is not met, the AL method proceeds with expanding the data set based on the acquisition function. The acquisition function measures how informative a new candidate sample is (Qing, Knudde, Couckuyt, Dhaene, & Shintani, 2020). Once the newly added sample is added to the data set, it needs to be evaluated. After that, the models are re-trained on the expanded data set, and the stopping criterion is checked again.

When the stopping criterion is reached, the GP models and respective data set provide valuable information that can be exploited by the engineer during the post-processing, for example, to learn more about where the feasible regions are in the design space. While the surrogate models can be used for further exploration or optimization, the feasible designs can serve as starting points for optimizing a design at a later stage.

4 Background

4.1 Gaussian Process Regression

A GP is a non-parametric Bayesian model that can be considered as a distribution over functions $f : \mathcal{X} \to \mathbb{R}$. A GP is defined by a mean function $m : \mathcal{X} \to \mathbb{R}$ and a covariance function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Hence, an unknown function $f(\mathbf{x})$ can be written in the stochastic form as: $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$.

While $m(\cdot)$ is often assumed zero, there are many choices for $k(\cdot, \cdot)$. Common choices are the squared exponential (Gaussian) kernel and the Matérn kernel (Frazier, 2018), which we will use later in the experiments. The hyperparameters, like signal variance, and length scales, are optimized using maximum likelihood estimation (MLE) (Mardia & Marshall, 1984).

Given training samples $\mathcal{D}_{tr} = (\mathbf{X}_{tr}, \mathbf{y}_{tr})$, the posterior belief can be derived. This posterior belief is Gaussian distributed as follows:

$$\mathbf{Y}_* | \mathcal{D}_{tr}, \mathbf{X}_* \sim \mathcal{N}(K_*^\top K^{-1} \mathbf{y}_{tr}, K_{**} - K_*^\top K^{-1} K_*), \tag{2}$$

where $\mathbf{X}_*$ denotes the testing samples and $\mathbf{Y}_*$ the predicted values, K is the correlation matrix between training samples $\mathcal{D}_{tr}$, K_* is the correlation vector calculated for all

pairs of training and test samples, and K_{**} is the correlation matrix between the test samples. Using this posterior, new samples can be drawn.

As the constraints in the constrained problem $G(\mathbf{x})$ (see Equation 1) are assumed independent, each constraint can be modelled independently with a GP. All constraint models are considered together (Rahat & Wood, 2020), so the posterior predictive distribution is a multi-variate Gaussian distribution with mean prediction vector $\boldsymbol{\mu}(\mathbf{x}) = (\mu_1(\mathbf{x}), \ldots, \mu_L(\mathbf{x}))^\top$ and predictive covariance matrix $\Sigma(\mathbf{x}) = diag(\sigma_1^2(\mathbf{x}), \ldots, \sigma_L^2(\mathbf{x}))$. That is:

$$p(\hat{G}|\mathbf{x}, \mathcal{D}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{x}), \Sigma(\mathbf{x})) = \prod_{l=1}^{L} p(\hat{g}_l|\mathbf{x}, \mathcal{D}_l), \tag{3}$$

where,

$$p(\hat{g}_l|\mathbf{x}, \mathcal{D}_l) = \mathcal{N}(\mu_l(\mathbf{x}), \sigma_l^2(\mathbf{x})|\mathbf{x}, \mathcal{D}_l). \tag{4}$$

To know whether a design vector $\mathbf{x}$ is feasible we use the following definition:

$$\begin{aligned} \mathbf{x} \in \mathcal{F} &\iff (\boldsymbol{\mu}(\mathbf{x}) \leq \mathbf{t}) = 1 \\ &\iff (\mu_1(\mathbf{x}) \leq t_1) \wedge \cdots \wedge (\mu_L(\mathbf{x}) \leq t_L). \end{aligned} \tag{5}$$

4.2 Probability of Feasibility

In constrained AL, acquisition functions often measure exploitation by a probability which answers the question: how likely will my prediction be below the constraint threshold? In order to know if a point is feasible, the probability of feasibility (PoF) can be used, and is defined as follows (Forrester & Keane, 2009):

$$\begin{aligned} p(\mathbf{x} \in \mathcal{F}) &= \prod_{l=1}^{L} p(\mathbf{x} \in \mathcal{F}_l) = \prod_{l=1}^{L} \Phi(\tau_l) \\ &= \prod_{l=1}^{L} p(p(\hat{g}_l|\mathbf{x}, \mathcal{D}_l) \leq t_l), \end{aligned} \tag{6}$$

where $\tau_l = \frac{t_l - \mu_l(\mathbf{x})}{\sigma_l(\mathbf{x})}$, and $\Phi(\cdot)$ is the cumulative Gaussian distribution function. Then, the probability of infeasibility $p(\mathbf{x} \in \mathcal{I}) = 1 - p(\mathbf{x} \in \mathcal{F})$ due to existing symmetry.

4.3 Model Uncertainty

A typical way of quantifying predictive uncertainty in acquisition functions is to straightforwardly use the predictive variance σ^2 (see Equation 2). In the case of information-theoretic approaches, the differential entropy $\mathbb{H}(\mathbf{x} \mid \hat{g}_l)$ of a Gaussian distribution is used. For a single constraint l we have

$$\mathbb{H}(\mathbf{x} \mid \hat{g}_l) = \frac{1}{2}\, ln\big(2\pi e \sigma_l^2(\mathbf{x})\big). \tag{7}$$

Table 1 Comparing the properties of the different acquisition functions in terms of sampling, exploration and exploitation strategy.

α	Sampling	Exploration method	Exploitation method
PoFV	Feasible region	Variance	Probability of feasibility
EF	Boundary	Differential Entropy	Log Probability of being at the boundary
PBE	Boundary	Differential Entropy	Probability of being at the boundary
B	Boundary	Standard Deviation	Predictive distance from the boundary
R	Boundary	Variance	Predictive distance from the boundary

Considering the multi-variate Gaussian distribution of $\hat{G}$, the differential entropy $\mathbb{H}(\mathbf{x} \mid \hat{G})$ is defined as (Rahat & Wood, 2020):

$$\mathbb{H}(\mathbf{x} \mid \hat{G}) = \frac{L}{2} ln(2\pi e) + \frac{1}{2} ln\big(det(\Sigma)\big) \propto ln\left(\prod_{l=1}^{L} \sigma_l^2(\mathbf{x})\right). \tag{8}$$

The entropy can thus be reduced to a function of the predictive variance.

An information-theoretic approach searches for a sample that will lead to the largest information gain. In other words, we want to find the sample that reduces the entropy of the posterior distribution the most. This coincides with finding the sample with the highest uncertainty in terms of the entropy function. So the reduction in entropy of the quantity of interest θ can then be written as follows (Hernández-Lobato, Gelbart, Hoffman, Adams, & Ghahramani, 2015):

$$\begin{aligned} EntropyLoss_\theta &= \mathbb{H}[p(\theta \mid \mathcal{D}_l)] - \mathbb{E}_{p(\hat{g}_l \mid \mathcal{D}_l, \mathbf{x})}(\mathbb{H}[p(\theta \mid \mathcal{D}_l \cup \{(\mathbf{x}, \hat{g}_l)\})]) \\ &= \mathbb{H}[p(\hat{g}_l \mid \mathbf{x}, \mathcal{D}_l)] - \mathbb{E}_{p(\theta \mid \mathcal{D}_l)}(\mathbb{H}[p(\hat{g}_l \mid \mathbf{x}, \mathcal{D}_l, \theta)]). \end{aligned} \tag{9}$$

5 Acquisition Functions for FRI

The acquisition function $\alpha(\mathbf{x}, \hat{G}, \mathbf{t})$ defines which point in the design space is most promising as the next sample. By optimizing the acquisition function, the optimizer $\mathbf{x}^*$ becomes the next sample. That is $\mathbf{x}^* = argmax_{\mathbf{x}}\ \alpha(\mathbf{x}, \hat{G}, \mathbf{t})$. Note that the acquisition function uses the knowledge of the surrogate models $\hat{G}$ to find $\mathbf{x}^*$. It is only after $\mathbf{x}^*$ has been found that the expensive simulations are run to know the actual value $G(\mathbf{x}^*)$ and expand the data set $\mathcal{D}$. Table 1 summarizes the properties of the state-of-the-art acquisition functions that we consider in this work.

5.1 Probability of Feasibility and Variance

The *Probability of Feasibility and Variance (PoFV)* used by Kaintura et al. (2018) is simply the product of the PoF and the predictive variance of the model, as shown in Equation 10 (for the case of a single constraint). Note that by using PoFV, the sampled points are also selected inside the feasible region, which in turn leads to a larger number of feasible designs relative to all the other acquisition functions considered

here.

$$\alpha_{PoFV[l]}(\mathbf{x}, \hat{g}_l, t_l) = p(\mathbf{x} \in \mathcal{F}_l)\, \sigma_l^2(\mathbf{x}) \tag{10}$$

To make this acquisition function work for more than one constraint, Nikova, Dhaene, and Couckuyt (2023) extends Equation 10 as follows:

$$\begin{aligned} \alpha_{PoFV}(\mathbf{x}, \hat{G}, \mathbf{t}) = p(\mathbf{x} \in \mathcal{F})\, det(\Sigma) &= \left(\prod_{l=1}^{L} p(\mathbf{x} \in \mathcal{F}_l)\right)\left(\prod_{l=1}^{L} \sigma_l^2(\mathbf{x})\right) \\ &= \prod_{l=1}^{L} p(\mathbf{x} \in \mathcal{F}_l)\, \sigma_l^2(\mathbf{x}) = \prod_{l=1}^{L} \alpha_{PoFV[l]}(\mathbf{x}, \hat{g}_l, t_l). \end{aligned} \tag{11}$$

5.2 Probability of Boundary and Entropy

Assuming the boundary is critical for identifying the feasible region accurately, some acquisition functions focus directly on locating the boundary $\beta \in \mathbb{R}^d$, between feasible and infeasible solutions. For example, Rahat and Wood (2020) introduce the probability of being at the boundary which exploits the knowledge about the boundary, i.e., where the model predictions change from feasible to infeasible values, and thus the feasible region it encapsulates. The probability provides information both on the feasible and infeasible regions as given in Equation 12, which uses the PoF.

$$p(\mathbf{x} \in \beta) = p(\mathbf{x} \in \mathcal{F})\, p(\mathbf{x} \in \mathcal{I}) \tag{12}$$

$p(\mathbf{x} \in \beta) = 0$ if a design is certainly feasible (or infeasible). Around the boundary, where there is more uncertainty about the feasibility of a design, the probability of being at the boundary becomes greater than zero. Hence, when maximizing this probability, the maximizer will be close to the predicted boundary.

Rahat and Wood (2020) created the *Probability of Boundary and Entropy (PBE)* acquisition function directly with multiple constraints in mind. By simply taking the product of the probability of being at the boundary and the differential entropy, this resulted in the following acquisition function:

$$\begin{aligned} \alpha_{PBE}(\mathbf{x}, \hat{G}, \mathbf{t}) &= p(\mathbf{x} \in \beta)\mathbb{H}(\mathbf{x}|\hat{G}) \\ &= p(\mathbf{x} \in \mathcal{F})p(\mathbf{x} \in \mathcal{I})\mathbb{H}(\mathbf{x}|\hat{G}) \\ &= p(\mathbf{x} \in \mathcal{F})(1 - p(\mathbf{x} \in \mathcal{F}))\mathbb{H}(\mathbf{x}|\hat{G}), \end{aligned} \tag{13}$$

where $\mathbb{H}(\mathbf{x}|\hat{G})$ is approximated as shown in Equation 8.

By using an acquisition function that jointly considers all constraints, there is no explicit model selection. PBE looks at the true boundary β instead of considering the separate constraint boundaries. Thus, this acquisition function is expected to perform better than acquisition functions that combine single-constraint versions.

5.3 Entropy Feasible

Entropy Feasible (EF) maximizes the information for both the infeasible and feasible regions (Knudde et al., 2019; Qing, Knudde, Couckuyt, Dhaene, & Shintani, 2020; Qing, Knudde, Couckuyt, Spina, & Dhaene, 2020; Qing et al., 2022). The information is measured using predictive entropy. This acquisition function has a closed-form expression as opposed to other information-theoretic approaches commonly used in BO (Hernández-Lobato et al., 2015). The original form of EF is defined for a lower and upper boundary per constraint ($a < g_l(\mathbf{x}) < b$). Therefore, the acquisition function computes the entropy loss in each of the three regions $g_l(\mathbf{x}) > b$, $a < g_l(\mathbf{x}) < b$, and $g_l(\mathbf{x}) < a$. When the acquisition function is adapted to only an upper boundary ($-\infty < g_l(\mathbf{x}) < t_l$), it can be simplified to the following formula (Rahat & Wood, 2020):

$$\begin{aligned}\alpha_{EF[l]}(\mathbf{x}, \hat{g}_l, t_l) =& EntropyLoss_{\hat{g}_l > t_l} + EntropyLoss_{-\infty < \hat{g}_l < t_l} + EntropyLoss_{\hat{g}_l < -\infty} \\ =& \, 3\mathbb{H}[p(\hat{g}_l \mid \mathbf{x}, \mathcal{D}_l)] \\ & - \mathbb{H}[p(\hat{g}_l \mid \mathbf{x}, \mathcal{D}_l, \hat{g}_l > t_l)] \\ & - \mathbb{H}[p(\hat{g}_l \mid \mathbf{x}, \mathcal{D}_l, -\infty < \hat{g}_l < t_l)] \\ & - \mathbb{H}[p(\hat{g}_l \mid \mathbf{x}, \mathcal{D}_l, \hat{g}_l < -\infty)] \\ =& \frac{1}{2} ln(2\pi e \sigma_l^2(\mathbf{x})) - ln(\Phi(\tau_l)(1 - \Phi(\tau_l))),\end{aligned} \tag{14}$$

where the first term denotes the differential entropy of a Gaussian distribution and the second term is the natural logarithm of the probability of being at the boundary. To achieve this simplification[1], the differential entropy of a truncated Gaussian distribution is used. Knudde et al. (2019) explain that this acquisition function is more explorative while the predictive variance is large. When the predictive variance decreases, there is more exploitation of the boundary region. The derivation for multiple constraints is shown by Qing et al. (2022) and consists of taking the sum of the individual logarithmic terms, as shown in Equation 15.

$$\alpha_{EF}(\mathbf{x}, \hat{G}, \mathbf{t}) = \sum_{l=1}^{L} \alpha_{EF[l]}(\mathbf{x}, \hat{g}_l, t_l) \tag{15}$$

5.4 Expected Feasibility

Lastly, we consider two closely related acquisition functions that were originally introduced for reliability studies. The first one is introduced by Bichon et al. (2008), while the other one is proposed by Ranjan et al. (2008). We will further refer to them as the *Bichon (B)* and *Ranjan (R)* acquisition functions, respectively. Both acquisition functions choose the next samples in the area of the boundary β of a single constraint as they calculate an average between the uncertainty and the predictive distance from

[1] In our implementation EF is computed exactly as in Knudde et al. (2019) and not using the approximation in Rahat and Wood (2020).

β. In order for these acquisition functions to work for feasibility, Rahat and Wood (2020) adapted both functions as follows:

$$\alpha_{B[l]}(\mathbf{x}, \hat{g}_l, t_l) = \sigma(\mathbf{x})\Big[\tau^+\Phi(\tau^+) + \tau^-\Phi(\tau^-) + \phi(\tau^+) + \phi(\tau^-) - 2\tau\Phi(\tau) - 2\phi(\tau)\Big], \quad (16)$$

$$\alpha_{R[l]}(\mathbf{x}, \hat{g}_l, t_l) = \sigma^2(\mathbf{x})\Big[\tau^2(\Phi(\tau^-) - \Phi(\tau^+)) + \tau^+\phi(\tau^-) - \tau^-\phi(\tau^+)\Big], \quad (17)$$

where $\tau^+ = \tau + 1$, $\tau^- = \tau - 1$, and $\phi(\cdot)$ is the probability density function of a univariate Gaussian distribution.

In the case of multiple constraints, Rahat and Wood (2020) reformulated an approach where only a single model with the best individual mean prediction is selected per iteration. This is called a composite criterion approach and is often used for system failure problems (Yang, Mi, Deng, & Liu, 2019). However, it does not make sense to use this approach for FRI because of three shortcomings. Firstly, this approach does not account for the boundary β on the full problem G, but for individual boundaries. Although a chosen sample for one constraint is predicted to be feasible, chances are high that it will violate another constraint. This can lead to redundant samples in the infeasible space. Secondly, this model selection will not consider prediction uncertainty. Lastly, if the constraints all have a different scale, the relative importance can be disturbed. Therefore, we opt for using the product of all constraints as follows:

$$\alpha_{B/R}(\mathbf{x}, \hat{G}, \mathbf{t}) = \prod_{l=1}^{L} \alpha_{B[l]/R[l]}(\mathbf{x}, \hat{g}_l, t_l) \quad (18)$$

Thus, the redundancy in the feasible space is reduced, and the prediction uncertainty captured in the individual acquisition functions is also captured in the product.

6 Bi-Objective Optimization of the Exploration-Exploitation Trade-Off

Recall that exploration can be reduced to the uncertainty of a model that is represented by the predictive variance $\sigma^2(\mathbf{x})$, while exploitation is based on the prediction being below a threshold (i.e., the PoF, see Equation 6). We will further denote this as a function of $\mathbf{x}$: $F(\mathbf{x}) = p(\mathbf{x} \in \mathcal{F})$. Note that this trade-off forms a bi-objective optimization problem that needs to be maximized:

$$max_{\mathbf{x}\in\mathcal{X}}\ (F(\mathbf{x}), \sigma^2(\mathbf{x})). \quad (19)$$

As both exploration and exploitation are important, these can be competing objectives. Only designs that can increase both objectives are preferred. This means we are searching for the so-called *non-dominated* or *Pareto-optimal* designs (i.e., the set of solutions that reveal the optimal trade-offs). A design $\mathbf{x}$ dominates $\mathbf{x}'$, iff $F(\mathbf{x}) \geq F(\mathbf{x}')$

and $\sigma^2(\mathbf{x}) \geq \sigma^2(\mathbf{x}')$, and they are not equal on both (Rojas Gonzalez & Van Nieuwenhuyse, 2020). The Pareto set is the set of non-dominated points, while the Pareto front consists of the Pareto set in the objective space: $\{(F(\mathbf{x}), \sigma^2(\mathbf{x})) \mid \mathbf{x} \text{ non-dominated}\}$. In what follows, we first introduce the novel acquisition functions based on scalarization, followed by a description of the proposed sampling policy.

6.1 Scalarization-Based Acquisition Functions

Multi-objective optimization problems can be solved by using scalarization methods (Miettinen & Mäkelä, 2002). By using such a method, the multiple objectives are decomposed into multiple single objective problems (i.e., each single-objective scalarization solves for a specific solution on the Pareto front). Often, the scalarization function depends on a set of weights that will determine the specific single-objective problem(s) to be solved, depending on the decision-maker preferences. Thus, a single objective optimization solver can be used to find an optimum of a given scalarization. Here we employ two well-known scalarization functions: the Augmented Tchebysheff (Equation 20) and the Augmented Achievement Scalarizing Function (Equation 21). These functions not only offer adequate convergence guarantees (Miettinen, 1999), but have also shown excellent empirical performance in the literature (see e.g., Knowles 2006; Rojas Gonzalez, Jalali, and Van Nieuwenhuyse 2020).

$$max_{i=1}^{M}\Big[w_i(f_i - z_i)\Big] + \gamma \sum_{i=1}^{M}\Big(w_i(f_i - z_i)\Big) \tag{20}$$

$$max_{i=1}^{M}\Big[\frac{f_i - z_i}{w_i}\Big] + \gamma \sum_{i=1}^{M}\Big(\frac{f_i - z_i}{w_i}\Big) \tag{21}$$

Note both equations 20 and 21 are defined for a minimization problem, where $\mathbf{f} = (f_i, \ldots, f_M)$ is the objective vector of size M. The weight vector $\mathbf{w}$ is chosen such that $\sum_{i=1}^{M} w_i = 1$, while γ is a positive value we set to 10^{-4}. Furthermore, both methods rely on a reference point $\mathbf{z}$, which is often the ideal point (i.e., the global optimum for each objective). For minimization this corresponds to $\mathbf{z} = (0, 0)$ in a normalized bi-objective space (as in Equations 20 and 21). For the maximization (see Equation 19), we reformulate it as follows (see also Mandow, Martín-Albob, and Perez-de-la Cruza 2023):

$$ATCH(\mathbf{x}, \mathbf{f}) = min_{i=1}^{M}\Big[w_i(f_i - z_i)\Big] + \gamma \sum_{i=1}^{M}\Big(w_i(f_i - z_i)\Big), \tag{22}$$

$$AASF(\mathbf{x}, \mathbf{f}) = min_{i=1}^{M}\Big[\frac{f_i - z_i}{w_i}\Big] + \gamma \sum_{i=1}^{M}\Big(\frac{f_i - z_i}{w_i}\Big), \tag{23}$$

with $\mathbf{z} = (1, 1)$. The scalarization-based acquisition functions are then defined as:

$$\alpha_{ATCH}(\mathbf{x}, \hat{G}, \mathbf{t}) = ATCH\Big(\mathbf{x}, (F(\mathbf{x}), \sigma^2(\mathbf{x}))\Big), \tag{24}$$

$$\alpha_{AASF}(\mathbf{x}, \hat{G}, \mathbf{t}) = AASF\Big(\mathbf{x}, (F(\mathbf{x}), \sigma^2(\mathbf{x}))\Big). \tag{25}$$

In order to use these acquisition functions for problems with multiple constraints, the objectives are calculated as follows:

$$(F(\mathbf{x}), \sigma^2(\mathbf{x})) = (p(\mathbf{x} \in \mathcal{F}), \bar{\sigma}^2_L(\mathbf{x})) = \Big(\prod_{l=1}^{L} p(\mathbf{x} \in \mathcal{F}_l), \frac{1}{L}\sum_{l=1}^{L} \sigma_l^2(\mathbf{x})\Big). \tag{26}$$

Hence, the product of PoF for each constraint and the mean predictive variance for each constraint after using min-max scaling to $[0, 1]$.

The weights $\mathbf{w}$ in the scalarization methods drive the selection on the Pareto front of the two objectives, and are often sampled uniformly distributed in the weight-space to explicitly show that the method can populate the entire Pareto front (Knowles, 2006). For our scalarization-based acquisition functions, we build a set of uniform weights a priori which is larger than the number of AL iterations. When the acquisition function is being optimized at each iteration, the weights are sampled from this set without repetition, such that we explore different trade-offs on the Pareto front. Note that uniform weight selection is an automatic procedure that does not require expert knowledge from the user. Other weight strategies, which can incorporate expert knowledge and user preference, are left for future research.

6.2 Exploration-Exploitation Trade-Off Front

Figure 2 illustrates how the Pareto front of the exploration-exploitation trade-off looks like for the cantilever beam (CB) problem (see 7.1 and A.1). The choice for the next sample depends on the acquisition function. If the acquisition function only considers one of the objectives (either PoF or VAR), the selection is straightforward: the maximum of that objective is taken. This coincides with the two endpoints of the Pareto front. As mentioned before, only the PoFV method is focused on sampling inside the feasible region, the other commonly used acquisition functions sample along the boundary. However, this distinction is not clearly visible in the trade-off plot. PoFV chooses a point somewhere in the middle range of both objectives as it maximizes their product. The boundary methods (PBE, B, R) will choose a point with higher variance and smaller PoF. Only EF chooses a point with higher PoF and lower variance.

The two scalarization-based acquisition functions ATCH and AASF choose a point depending on the weights. For this plot, we compare two weights with both functions. Because of the nature of the chosen scalarization methods, where ATCH multiplies by the weights $\mathbf{w}_i$, while AASF divides by these weights, we notice that both acquisition functions sample opposite of each other. The benefit of using the weights is that the full range of the Pareto front can be sampled.

Note that this plot looks different at every iteration. When the model is trained on a larger data set, and thus predicts more accurately, the Pareto front will tend

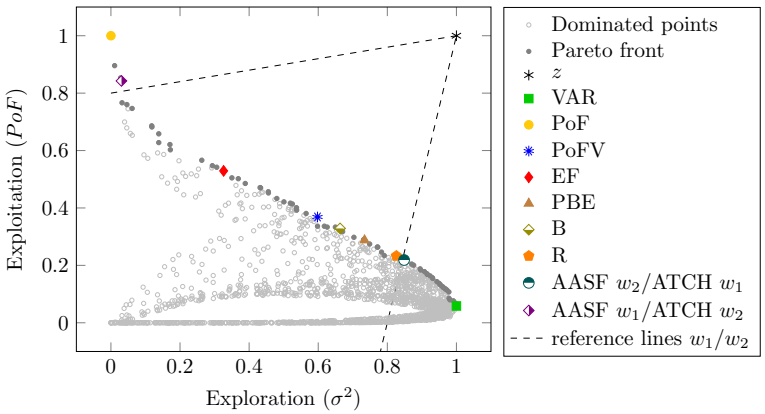


Fig. 2 Exploration-exploitation objective space for the cantilever beam problem with one constraint created with a Halton sampling set of 2000 samples. Depending on which acquisition function is optimized a different sample on the front is chosen. For ATCH and AASF the chosen point depends on the weights. Two weight vectors were used for this plot: $\mathbf{w}_1 = (\frac{1}{6}, \frac{5}{6})$, $\mathbf{w}_2 = (\frac{5}{6}, \frac{1}{6})$. By changing the weight, the reference lines also change, and the point on the front closest to the reference lines is chosen.

be discontinuous. A better model prediction means the PoF will be closer to the two extremes for most of the designs. In that case, the front will consist of fewer points in the middle range of PoF and the various acquisition functions will converge to choose the same sample as the next one.

7 Experiments

We compare the performance of the proposed scalarization-based acquisition functions against the state-of-the-art on several real-world engineering design examples. The problems range from two-dimensional with two constraints up to a seven-dimensional design space with seven constraints. This selection of problems illustrates the purpose of AL in the design process of various engineering disciplines with varying complexity. Furthermore, we evaluate an ADS testing example in which the safety of an AEB system is under test. Following the overview of Wotawa et al. (2021), our AL approach falls under the learning-based methods for finding safety-critical scenarios. By formulating the testing problem as an engineering problem with a safety measure as a constraint, we can borrow the concepts of FRI to identify scenarios of interest.

Decision-makers either use the surrogate model or rely on the final data set for further design analysis. As both are influenced by the acquisition function, we evaluate the acquisition function by looking at the performance of the constraint surrogate models and the number of acquired feasible designs in the final data set.

7.1 Real-world Engineering Benchmarks

We have chosen five analytically defined real-world engineering problems as the benchmark. The diverse selection of applications emphasizes the versatile adoption of the

AL methods for FRI. Please refer to Appendix A for the analytical formulation of the problems.

- The first engineering problem is the design of a cantilever beam (CB) (Engineering Toolbox, 2013) in which the shape of the beam is defined by two parameters while two constraints are limiting the stress (g_1) and displacement (g_2) of the beam. g_1 is the most restrictive constraint and defines the boundary of the feasible region. The full problem definition is given in A.1.
- The second problem comes from chemical engineering and is called the process flow sheeting problem (PFS) (Kumar et al., 2020). Originally the problem consists of three variables and three constraints, forming a non-convex constrained optimization problem. Here we reduce the problem to two variables and two constraints, which is described in A.2. Both g_1 and g_2 define the boundary of the feasible region. Notably, g_2 is the most restrictive. The PFS problem has the lowest feasibility ratio ρ, i.e., the percentage of feasible designs in the final data set, of all considered problems.
- We use the Nowacki beam (NB) (Nowacki, 2005) problem as adapted to FRI by Knudde et al. (2019). The complexity of this problem lies in the fact that it has five constraints. We are interested in finding the breadth and height of the beam constrained by the area, the tip deflection, and other stresses. Because of all these constraints, $\rho = 11.0\%$. The complete problem is described in A.3. Experiments show that constraint g_4 and g_5 are not crucial for the solution, while g_1 is the most restricting.
- We consider the design of a tension/compression spring (TCS) (Kumar et al., 2020). It is a three-dimensional problem in which the variables represent the diameter of the wire, the coil, and the active coils. This problem is restricted by three constraints as defined in A.4. Of the three constraints, g_3 is the most restrictive, followed by g_1 and then g_2. This brings the feasibility ratio to 33.65%. Although ρ is much higher compared to the previous two problems, the higher dimensionality and additional constraints make it more difficult.
- The last real-world problem is taken from aerospace engineering, i.e., a seven-dimensional speed reducer (SR) (Kumar et al., 2020; Ray, 2003). The problem focuses on designing a speed reducer situated in a small aircraft engine. This is an example of a very highly-constrained use case with 11 constraints. However, we select 7 constraints for our experiments which are given in A.5 like in Nikova et al. (2023). Preliminary experiments for the test set show that constraints g_3, g_4, and g_5 are not restricting the problem. The most restrictive constraint is g_7. Although this problem has the highest feasibility ratio, namely $\rho = 56.54\%$, the dimensionality and the number of constraints make this problem difficult to learn.

7.2 Testing Automatic Emergency Braking System

Simulations of driving scenarios generate time-dependent data, also referred to as parameter trajectories i.e., from the start of the scenario until a collision happens. Following the testing strategies as proposed by Zhang et al. (2021), identification

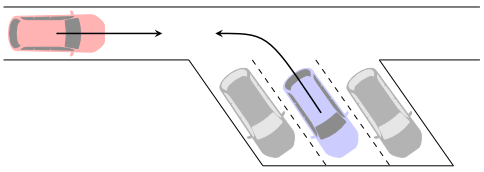


Fig. 3 AEB system testing scenario. The host vehicle is the red one driving on the road to the right. The host vehicle is the car with the AEB system under test. There are three vehicles parked along the road. The vehicle that is parked in the middle (blue) is reversing out of the parking spot. Because the vehicle is occluded by another parked vehicle, the host vehicle is not able to immediately notice the vehicle reversing. Depending on the velocity of both vehicles, a collision could occur or be avoided.

methods can either make use of the parameter trajectories or not. Often the full trajectory is not needed to know whether a scenario is safe or critical. Learning-based identification methods, like the ones considered in this work, belong to the category of *exploring logical scenarios without parameter trajectories*. This means that the scenario is parametrized, and the trajectories are not considered. Instead, information is extracted from the trajectories to create meaningful parameters, e.g. the minimum time to collision or impact velocity at collision time. Consequently, the extracted parameters can serve as safety-critical measures or are used to calculate other more complex safety-critical measures.

The input of the simulation is also given by parameters. We choose the inputs to define the initial state of the scenario. A scenario instance is a concrete combination of the parameters assigned to the scenario. The way the scenario progresses beyond the initial state is then defined by models for the host vehicle with the ADS under test as well as for the other vehicles in the scenario.

We follow the definitions of a functional, logical, and concrete scenario as given in Menzel, Bagschik, and Maurer (2018) to create the testing environment. The functional scenario of our application is depicted in Figure 3 and consists of a one-lane road with angled parking spots. The moving objects in this scenario are the host vehicle which includes the system under test and a vehicle reversing from a parking spot that cuts into the lane of the host vehicle.

For the purpose of the experiments, the logical scenario consists of two-parameter ranges, namely, the host and reversing velocities: $v_{host} \in [5, 15] m/s$ and $v_{reverse} \in [0, 5] m/s$. The host vehicle keeps a constant speed as defined by v_{host}, until the AEB is triggered, while the reversing vehicle follows a trapezoidal speed profile. The vehicle accelerates when the action is triggered in time until $v_{reverse}$ is reached, at a certain location on the road the reversing velocity is decelerated until the vehicle is standing still. By choosing a value for the design parameters, a concrete scenario is simulated. The design vector, i.e. a sample in the design space, consists of the parameter values that define the concrete scenario.

The chosen safety-critical constraint is based on the severity of a possible collision or emergency braking event. If there is no collision and AEB is not activated, the severity is -1. A severity between -1 and 0 indicates a collision was avoided by the AEB, where a smaller value means more safety margin. If the severity is above 0 up to 1, a collision occurred with reduced impact because the AEB was activated. Else

Table 2 Overview of the real-world problems. d and L stand for the number of variables and the number of constraints of the problem, respectively. ρ is the feasibility ratio in the design space. The column depicted by N is the size of the initial data set. q is the number of AL steps. This is equal to the number of samples that are selected adaptively after the initial data set.

Problem	d	L	$\rho(\%)$	N	q
Cantilever Beam (CB)	2	2	9.42	6	20
Process Flow Sheeting (PFS)	2	2	0.28	6	20
Nowacki Beam (NB)	2	5	11.0	6	20
Tension/Compression Spring (TCS)	3	3	33.65	9	30
Speed Reducer (SR)	7	7	56.54	21	70
Automatic Emergency Braking System (AEB)	2	1	43.8	20	100

if a collision happened without the AEB being activated, the severity is 1. Note that, the AEB not being activated can also be because the impact was outside the sensors' field of view. Nonetheless, it is a critical scenario.

Although there is only one constraint defined in a two-dimensional design space, the feasible region, i.e., the region where no collisions occur, is difficult to identify. Firstly, the severity constraint creates a discontinuous function around the boundary which is caused by the fact that the severity is calculated differently depending on whether there was a collision or not. Secondly, the feasible region consists of multiple regions, and we want to identify all of them and not miss a region.

7.3 Experimental Setup

All experiments on the benchmark problems were conducted in Python. The AEB problem was set up using Simcenter Prescan[2] for the simulation of the scenario and system under test, and Simcenter HEEDS[3] for automating the workflow between the simulation tool and the AL Python code.

The Bayesian AL framework and the acquisition functions are implemented using Trieste (Moss et al., 2024), while the GPs are constructed and trained using GPflow (Matthews et al., 2017). For all experiments the GP is constructed with a Matérn 5/2 kernel. The initial designs are sampled using a Halton sequence of size N. This size depends on the problem as shown in Table 2. The stopping criterion is set to the maximum number of iterations q, as given in Table 2. All results are taken from running the experiments 10 times with different random seeds. The default setup for optimizing the acquisition functions was used, namely a standard multi-start parallel L-BFGS-B optimizer. For the scalarization-based acquisition functions a set of uniform weights of size $50 \times d$ is taken, which is more than the number of iterations.

As more accurate GP predictions lead to better design selection by the acquisition function, we evaluate the GP performance at each iteration against a test set. The test sets consist of 5000 data points for all problems, except for the AEB problem where evaluations are expensive, and a smaller test set of 1000 data points was used. The

[2]https://plm.sw.siemens.com/en-US/simcenter/autonomous-vehicle-solutions/prescan/
[3]https://plm.sw.siemens.com/en-US/simcenter/integration-solutions/heeds/

test set is created using a Halton sequence in the design space. All the methods are also compared against a baseline method. The baseline uses the predictive variance as acquisition function. This function only consists of an explorative part and is not exploitative. We refer to this baseline method as VAR.

The performance of the constraint surrogates will be measured every time a new design is acquired in order to see how adding the design to the training data influences the model performance. Although the surrogates are regression models, the classification problem with (in)feasible classes is also considered. To obtain the classes, the predicted values are compared against the thresholds $\mathbf{t} = (0, \ldots, 0)^\top$ as shown in Equation 5. Previous work uses the F_1-score (Knudde et al., 2019) and the *informedness* (Rahat & Wood, 2020) as the performance metric. However, these metrics are not suitable for feasible regions. Because the number of feasible designs can be much lower than the number of infeasible designs, there could be a large class imbalance. Table 2 lists the feasibility ratio ρ which is computed as the ratio of actual feasible designs in the test set against the full test set. This ratio shows the class imbalance in the different problems. Luque, Carrasco, Martín, and de las Heras (2019) show that the F_1-score is not suitable for class imbalanced problems, as it is heavily biased by the class imbalance and is not symmetrical, i.e., the choice for the positive and negative class influences the result.

Thus, in this study, we use a different metric called *Matthews correlation coefficient (MCC)* (Boughorbel, Jarray, & El-Anbari, 2017). MCC is a balanced correlation coefficient between the observed and predicted binary classifications. It captures the quality of the classification predictions of the surrogate model. As all four categories of the confusion matrix (TP, TN, FP, FN) are used in the computations, it can take both the successes and the classification errors into account. MCC is computed as follows:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{27}$$

and can take any value in the range $[-1, 1]$, where 1 corresponds with perfect prediction, 0 is equivalent to a random guess, and -1 is total disagreement.

We propose to use the *Negative Log Likelihood (NLL)*, a measure to capture how well a model's predicted probability distribution matches the actual data. For this we compare the final regression models for each constraint against the actual binary feasibility label f^{true}. In order to get the predicted feasibility probability f^{pred}, a softmax activation function is applied to the continuous predicted values. The lower the NLL value is, the better the model fits the data. For a test set of size N_{test}, the NLL is computed as follows:

$$NLL = -\sum_{i=1}^{N_{test}} f_i^{true}\, ln(f_i^{pred}) + (1 - f_i^{true})\, ln(1 - f_i^{pred}). \tag{28}$$

From an engineer's perspective, the predictive capability of the surrogate model is not the sole priority. As the design process progresses, the engineer requires a sufficient

number of feasible designs that serve as strong candidates for further development and/or implementation. Consequently, it is crucial to ensure that the dataset obtained after AL includes an ample selection of feasible designs for the engineer to evaluate and refine. Therefore, for each problem, we also compare the various acquisition functions based on how many feasible designs they select. Hence, giving preference to functions that consistently acquire feasible designs while better learning the feasible region.

8 Results

8.1 Real-world Engineering Examples

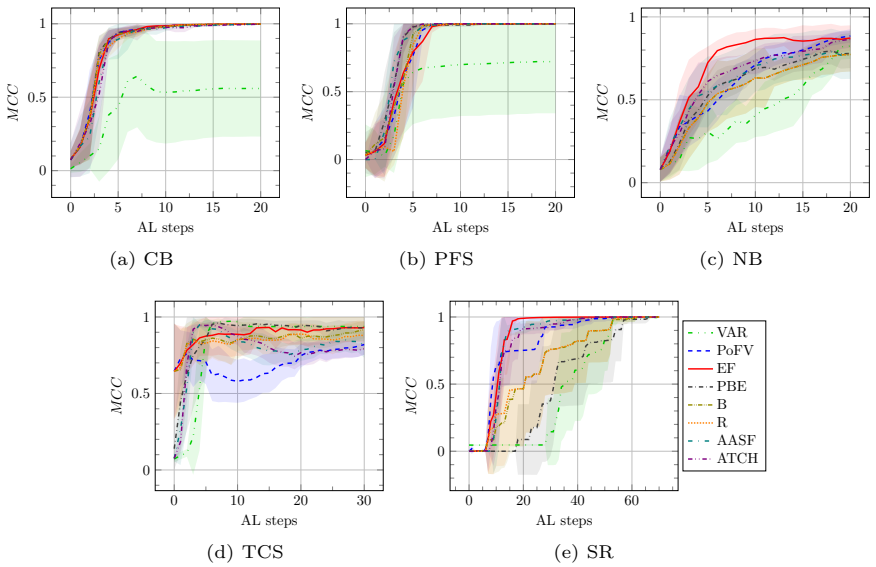
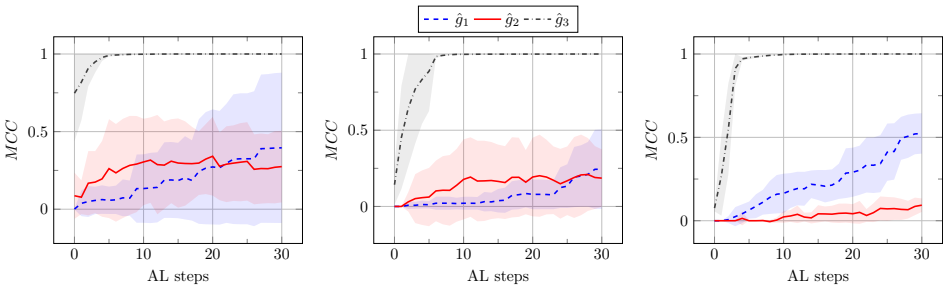


Fig. 4 The 5 acquisition functions (PoFV, EF, PBE, B, R) and 2 scalarization-based acquisition functions (AASF, ATCH) are compared against the baseline (VAR) for the different real-world engineering problems in terms of model performance. These plots show the mean MCC and one standard deviation of the 10 runs for all constraints together at each AL steps after initialization.

The main results begin with Figure 4, which depicts the performance of the competing acquisition functions for the five constrained engineering problems; note that we include supplementary results in the appendix. In this figure, the results are shown for the full problem, so we are not considering the constraints individually. Every method, except the baseline VAR, is able to learn the feasible region with a similar amount of data for the CB problem. For the PFS problem, the scalarization-based acquisition functions ATCH and AASF show better performance early on. With only 16 samples the MCC has already converged to a nearly perfect score, while VAR has a lower MCC and much larger variance. As both these two-dimensional problems are relatively easy to learn, this behaviour is expected.

We see a larger difference between the acquisition functions when looking at the plots for the NB, TCS and SR problems. These problems have more constraints (and dimensions). For the NB problem, EF reaches the highest MCC first. AASF, ATCH and PoFV follow and reach comparable final performance. In the case of TCS, PoFV performs the worst, even below VAR. AASF and ATCH have a good start, but their performance makes a drop to end up in the same range as PoFV. All other methods have a more stable performance. For the SR problem, the EF method has remarkable performance. PBE has a similar performance to the baseline VAR method and needs around 60 iterations to reach the highest MCC.

In general, Figure 4 shows that by increasing dimensionality, not all methods scale equally well. While for most benchmark problems EF and the proposed scalarization-based acquisition functions achieve the best average performance, AASF and ATCH allow more control for the engineer to explore the different trade-offs at a given stage of the decision-making process.



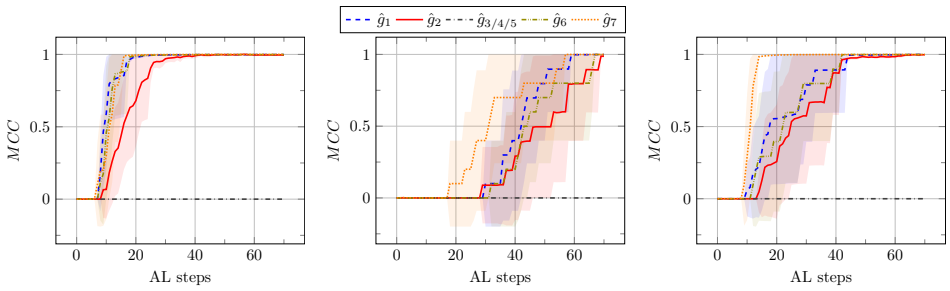
(a) EF (b) PBE (c) AASF

Fig. 5 Performance of the individual constraint models $\hat{g}_l$ in the tension/compression spring problem using the EF, PBE and AASF acquisition functions. The figures show the MCC mean score and one standard deviation of 10 runs at each AL step.

Figures 5 and 6 make a comparison of the MCC for the individual constraints of the TCS and SR problem, respectively, between the EF, PBE and AASF acquisition functions. The results with the other four acquisition functions are depicted in Figures B2 and B3. Figure B1 shows all acquisition functions for the NB problem.

We observe that the most restricting constraint (g_3 for TCS and g_7 for SR) is learned first and well, which reflects the good overall performance. The explanation behind this is that in order to learn and find the feasible region, the constraints that define the region need to be learned best. Scalarization-based acquisition functions, together with EF, overall lead to the fastest convergence of the most restricting constraint.

The models for the other constraints take longer to learn or are lacking. For the TCS problem, constraints g_1 and g_2 are better learned by some methods (PoFV, AASF, ATCH), while for others there remains a high uncertainty around them (EF, PBE, B, R). For the SR problem, EF is able to learn all restricting constraints almost equally fast. The other acquisition functions also learn all constraints, but there is a



(a) EF (b) PBE (c) AASF

Fig. 6 Performance of the individual constraint models $\hat{g}_l$ in the speed reducer problem depending on the acquisition function. The figures show the MCC mean score and one standard deviation of 10 runs for the EF, PBE and AASF acquisition functions at each AL step.

clear order: the restrictiveness of the constraint influences the convergence rate. The slowest convergence is observed for PBE. There are also three constraints that remain at $MCC = 0$ for all methods. These are the constraints that are not restricting the design, and hence always predict feasible, which is correct. This is also visible in Table B1, which show the mean and standard deviation values of MCC and NLL for each final constraint model in each problem.

Also from the results in Table B1, we observe that the two-dimensional CB and PFS problems result in similar performance for all constraints and all methods. Adding more constraints leads to a selection of designs that improve only the constraint models that define the boundary of the feasible region. Remarkably, a clear link between the method performance and the feasibility ratio ρ is not evident. In terms of NLL (see Table B2), the best method varies for each problem and constraint. Overall, it seems ATCH and AASF more often lead to better models as the NLL is the lowest.

In Table 3 we look at the number of feasible designs in the final data set without considering the initial designs. As PoFV is focused on sampling inside the feasible region, evidently it most often returns the highest number of feasible designs. Incidentally, the boundary methods (EF, PBE, B, R) yield the worst performance on this metric. The proposed scalarization-based acquisition functions are in between, but often closer to PoFV. This is a desired aspect and a key feature of the proposed acquisition functions, as they achieve a good balance between exploiting the feasible region for competitive samples and exploring the full design space to reach good model performance.

From the running times in Table 4, it is evident that the best overall performing methods (i.e., EF, AASF and ATCH) require more computational effort than the other acquisition functions. However, in practice this trade-off is relative to the underlying simulation used: if the cost of evaluating the simulation is 10^2 or 10^3 times more expensive than the optimization of the acquisition functions, then the differences shown in Table 4 become irrelevant. For example, the simulation of the AEB problem runs for $2-3$ minutes per design, in which case the acquisition function optimization runtime can be significant. However, there also exist expensive simulations, like in the

Table 3 Number of feasible designs (mean ± standard deviation of 10 runs) in the final data set of size q (listed in Table 2). The initial data set is not considered here. The highest number of feasible designs for each problem is underlined.

	PoFV	EF	PBE	B	R	AASF	ATCH
CB	$\underline{14.8 \pm 1.2}$	5.1 ± 1.2	4.8 ± 0.7	5.0 ± 0.0	5.0 ± 0.0	9.5 ± 1.3	9.7 ± 2.1
PFS	$\underline{15.5 \pm 0.9}$	1.1 ± 0.8	1.0 ± 0.9	1.8 ± 0.7	1.3 ± 0.8	7.3 ± 1.6	7.5 ± 1.7
NB	12.3 ± 1.2	8.8 ± 1.6	1.8 ± 1.1	1.7 ± 1.1	1.7 ± 1.1	12.0 ± 1.7	$\underline{12.7 \pm 1.8}$
TCS	$\underline{26.4 \pm 1.0}$	15.1 ± 4.1	11.7 ± 2.5	13.1 ± 3.0	13.2 ± 3.2	18.6 ± 1.9	19.1 ± 1.7
SR	$\underline{57.1 \pm 4.6}$	16.1 ± 5.2	39.9 ± 4.1	39.8 ± 3.2	40.1 ± 3.1	55.4 ± 2.7	53.1 ± 2.3
AEB	$\underline{68.2 \pm 3.1}$	33.7 ± 2.6	39.5 ± 2.1	37.3 ± 1.9	40.9 ± 2.3	54.1 ± 2.6	54.0 ± 2.6

Table 4 Average running time (in seconds) of acquiring a new data sample using different acquisition functions. We compare the times for a problem with one constraint (CB, average of 20 iterations) and a problem with seven constraints (SR, average of 70 iterations).

Problem	L	PoFV	EF	PBE	B	R	AASF	ATCH
CB	1	0.19s	0.59s	0.90s	0.10s	0.09s	2.19s	2.71s
SR	7	2.15s	26.49s	6.80s	0.24s	0.22s	23.69s	19.99s

field of computational fluid dynamics, which can run for hours or even days (Aultman, Wang, Auza-Gutierrez, & Duan, 2022).

8.2 Testing Automatic Emergency Braking System

The results for the AEB problem (Figure 7, Table 3 and Table B1) show that the single constraint in this problem is actually much more difficult to capture than the other problems with multiple constraints. The MCC curves depicted in Figure 7 confirm that, even after 100 steps, convergence is difficult for all acquisition functions, with EF exhibiting the greatest variance and fluctuation. This lower performance can be attributed to the increased complexity introduced by the discontinuity in the severity constraint, which complicates learning for the GP. This discontinuity is evident in the contour plot in Figure 8, highlighting the fragmented nature of the feasible regions. There are two disconnected feasible regions: the upper left and a band near the x-axis. The largest discontinuity is observed near the x-axis. When $v_{reverse}$ is close to 0, AEB is not activated and no collision happens. A slightly higher $v_{reverse}$, however, does lead to a collision with a severity larger than 0.5. This behaviour is observed for all values of v_{host}. Another area in the design space with a large discontinuity is around $(6, 1.5)$ where there are sudden peaks.

Overall we observe that considering the exploration-exploitation trade-off explicitly, can lead to more feasible designs while maintaining a good model performance. This is important, because we want to be able to identify as many different states of the system under test as possible, within the simulation budget. This provides a clearer understanding of the safety-critical behaviour of the AEB system.

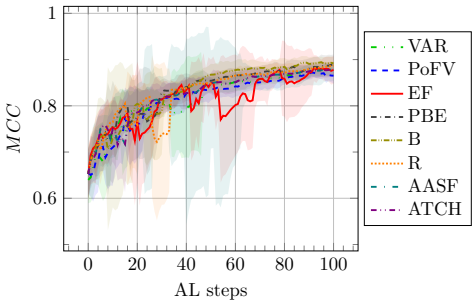


Fig. 7 All the acquisition functions are compared against the baseline function VAR for the AEB example. These plots show the mean MCC and one standard deviation of the 10 runs for the severity constraint at each AL step.

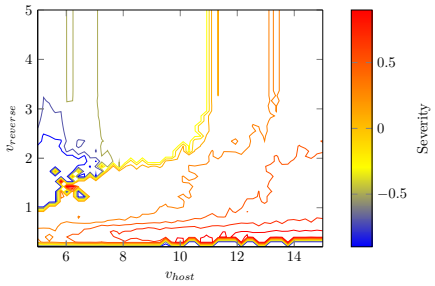


Fig. 8 Contour plot of severity constraint in the design space of the AEB system testing problem. The contour plot is created by running 2500 simulations in a grid. The contour lines are more concentrated near the boundaries between feasible and infeasible according to the severity measure.

8.3 Ablation Study

We study the balance of exploration and exploitation of various acquisition functions, arguably, the most important part of the proposed framework. Hence, besides the PoFV acquisition function, we also examine the VAR (only exploration) and PoF (only exploitation) as separate acquisition functions. Lastly, we compare these methods with the proposed scalarization-based acquisition functions to show the influence of adding weights and flexibility in the design selection.

Figure 9 shows the MCC performance for all benchmark problems. The VAR only focuses on exploration and, as expected, shows the overall worst performance; while PoF can outperform VAR, the proposed scalarization-based acquisition functions remain superior on average. The expected exploitative behaviour of PoF is to choose designs near known feasible designs. Hence, in the design space the choices made by PoF create a trail of only feasible designs, which is in general an undesired behaviour, as it often leads to the discovery of only a very small part of the feasible region.

We observe the exploitative strategy works well in the case of the PFS problem due to the very small feasible region. For the higher-dimensional SR problem, PoF has many directions to choose from, which leads to a more explorative approach than expected. De Ath et al. (2021) also observe this behaviour and confirm that for higher dimensional problems exploitative strategies unintentionally become explorative. While these results demonstrate that focusing solely on exploitation or exploration may work in specific cases, a balanced combination consistently yields superior outcomes. Moreover, the proposed acquisition functions provide engineers with greater control over the trade-off between objectives, a critical aspect for practical decision-making.

In addition to evaluating model performance, we also examine the selections made by each acquisition function on the Pareto Front. Figure 10 illustrates how the different

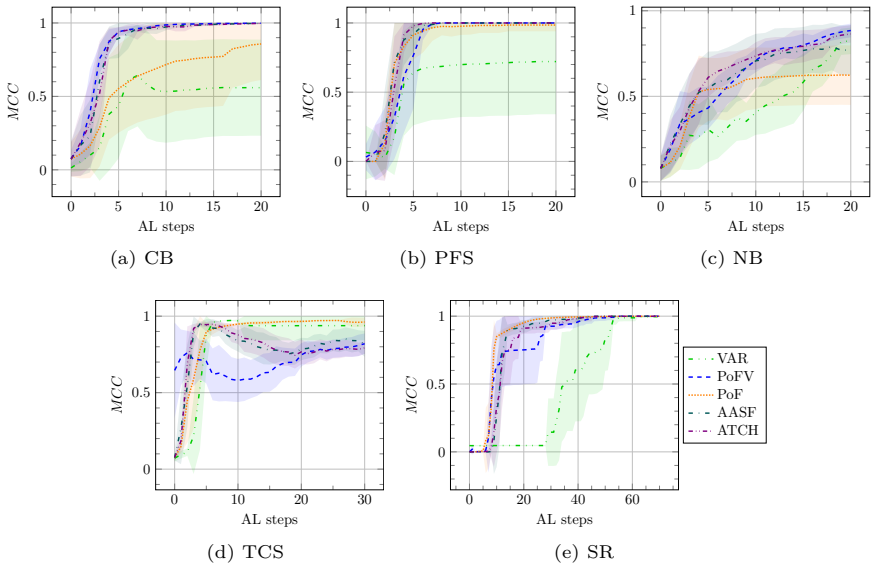


Fig. 9 Comparison of pure exploration (VAR), pure exploitation (PoF), a combination (PoFV) and the scalarization-based (AASF, ATCH) acquisition functions on the benchmark problems. The MCC mean and one standard deviation of 10 runs are shown for all constraints. A better balance between exploration and exploitation leads to faster convergence on all problems.

acquisition functions choose points over 10 iterations for the CB problem with a single constraint. As anticipated, VAR and PoF consistently remain in the extremes throughout all iterations. Only acquisition functions that balance both objectives result in a broader distribution of points along the front, with only AASF and ATCH achieving full coverage across both objectives.

9 Conclusion and Future Work

We scrutinized several common acquisition functions used in AL for FRI. Most of them consist of an exploitation part measured by the PoF, and an exploration part, as quantified by the variance of a GP model. The proposed idea of looking at the exploration-exploitation trade-off from a multi-objective standpoint provides new insights and opportunities for acquisition function design in the field of FRI. In contrast to existing methods, we propose two novel acquisition functions that explicitly model this trade-off. By considering exploration and exploitation as a bi-objective maximization problem, we propose two scalarization-based acquisition functions. The proposed acquisition functions effectively balance exploiting the feasible region for competitive samples and exploring the entire design space to enhance model performance. Additionally, they offer engineers greater control over the Pareto-optimal trade-off between the objectives, a crucial feature desired in practical settings.

We evaluated all methods across multiple engineering design problems to demonstrate the versatility of intelligent design space exploration. The results show that

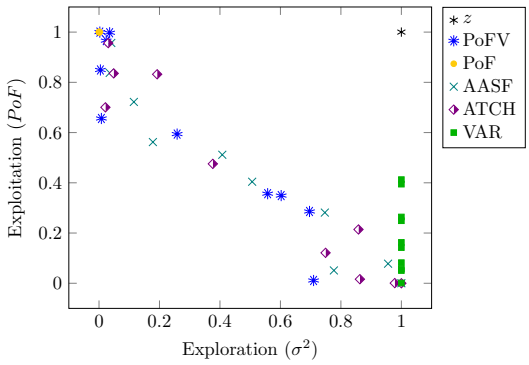


Fig. 10 Acquired designs according to the acquisition functions in exploration-exploitation objective space. Each acquisition function selects 10 designs iteratively on the CB problem with one constraint. Acquisition functions considering both objectives select designs on a broader range of the objectives.

EF excels as a boundary sampling method, while PoFV generates the most feasible samples. The proposed scalarization-based acquisition functions strike a desired balance, achieving competitive performance for most problems, both in terms of MCC and feasible sample count. The sampling behaviour of the proposed methods, driven by an explicit trade-off, alternates between exploration and exploitation, resulting in the best overall NLL performance. Naturally, the experiments also highlight several limitations. Key issues include the challenges GPs face with discontinuous functions, poor extrapolation near design space edges, and computational inefficiency in high dimensions. In this regard, more advanced models should be employed, which can, for example, handle and visualize high-dimensional problems (van der Maaten & Hinton, 2008; Xiao et al., 2023, 2024). Furthermore, acquisition functions focusing on the feasible boundary often yield fewer useful designs for engineers, while scalarization-based methods depend heavily on weight selection, which can impact performance if not carefully chosen. Additionally, as models improve, the Pareto front can become more discontinuous, leading to repeated sampling at extremes.

Future work will focus on dealing with the aforementioned limitations. More specifically, the exploration of methods to enhance acquisition strategies, even in the presence of dynamic and challenging Pareto fronts. One promising direction in this regard is optimizing weight selection. The current approach relies on uniformly distributed weights for scalarizing functions, making the weight selection automatic. This could be extended to incorporate specific engineer preferences (Gaudrie, Le Riche, Picheny, Enaux, & Herbert, 2020; González, Dai, Damianou, & Lawrence, 2017; Xin et al., 2018). For example, if one objective is preferred over the other, weights should be chosen accordingly without completely disregarding the other objective, as that would lead to less beneficial behaviour. Further experiments are needed to identify the best strategy for adding preference through weight control. Nevertheless, preference learning would be a valuable addition to scalarization-based acquisition functions. These functions could also be adapted for batch settings, where a set of weights is selected

per iteration. Furthermore, the trade-off could be expanded to include additional objectives, such as cost or design diversity, which are critical for engineers.

Acknowledgements. The conducted work is part of the Baekeland Research Project HBC.2021.0841 on "Data-efficient and Explainable Engineering Design". Ioana Nikova gratefully acknowledges the Flemish Agency on Innovation and Entrepreneurship (VLAIO) for the financial support. This work has also been supported by the Flemish Government under the Flanders Artificial Intelligence Research program. Sebastian Rojas Gonzalez is funded by grant #12AZF24N from the Research Foundation Flanders (FWO).

Funding. This work is funded by the Flemish Agency on Innovation and Entrepreneurship (VLAIO) as part of the Baekeland Research Project HBC.2021.0841 and by the Research Foundation Flanders (FWO) under grant #12AZF24N.

Data Availability. The data generated on the AEB system testing application belongs to *Siemens Industry Software*, and they have not given their permission for researchers to share the data publicly. However, the data for the other benchmarks can be artificially generated with the equations provided in Appendix A.

Declarations

Competing Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Appendix A Benchmark Problem Definitions

The benchmark of constrained engineering problems that were used to illustrate the Bayesian AL methods are described here.

A.1 Cantilever Beam (CB)

$$\begin{aligned} g_1(\mathbf{x}) &= \frac{P\lambda x_1}{2I} - 5000 \leq 0, \\ g_2(\mathbf{x}) &= \frac{P\lambda^3}{3EI} - 0.1 \leq 0, \end{aligned} \tag{A1}$$

with

$$\begin{gathered} I = \frac{1}{12}b(x_1 - 2h)^3 + 2(\frac{1}{12}x_2h^3 + \frac{1}{4}x_2h(x_1 - h)^2), \\ P = 1000,\ \lambda = 36,\ E = 1.0e7,\ h = 0.1,\ b = 0.25, \end{gathered}$$

where $3 \leq x_1 \leq 7$ and $2 \leq x_2 \leq 12$.

Note that Figure 2 was created when only considering the g_1 constraint.

A.2 Process Flow Sheeting (PFS)

$$\begin{aligned} g_1(\mathbf{x}) &= -exp(x_1 - 0.2) - x_2 \leq 0, \\ g_2(\mathbf{x}) &= x_2 + 2.1 \leq 0, \end{aligned} \tag{A2}$$

where $0.2 \leq x_1 \leq 1$ and $-2.22554 \leq x_2 \leq -1$.

A.3 Nowacki Beam (NB)

$$\begin{aligned} g_1(\mathbf{x}) &= x_1x_2 - 2500 \leq 0, \\ g_2(\mathbf{x}) &= \frac{P\lambda^3}{3EI_y} - 5 \leq 0, \\ g_3(\mathbf{x}) &= \frac{6P\lambda}{x_1x_2^2} - 240 \leq 0, \\ g_4(\mathbf{x}) &= \frac{1.5P}{x_1x_2} - 120 \leq 0, \\ g_5(\mathbf{x}) &= 2P - 4/\lambda^2\sqrt{\frac{GI_TEI_z}{1-\nu^2}} \leq 0, \end{aligned} \tag{A3}$$

with

$$\begin{gathered} I_y = \frac{x_1x_2^3}{12},\ I_z = \frac{x_1^3x_2}{12},\ I_T = I_y + I_z, \\ \lambda = 500,\ P = 5000,\ E = 216620,\ G = 86650,\ \nu = 0.27, \end{gathered}$$

where $10 \leq x_1 \leq 100$ and $20 \leq x_2 \leq 250$.

A.4 Tension/Compression Spring (TCS)

$$
\begin{aligned}
g_1(\mathbf{x}) &= \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0, \\
g_2(\mathbf{x}) &= 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0, \\
g_3(\mathbf{x}) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0,
\end{aligned}
\tag{A4}
$$

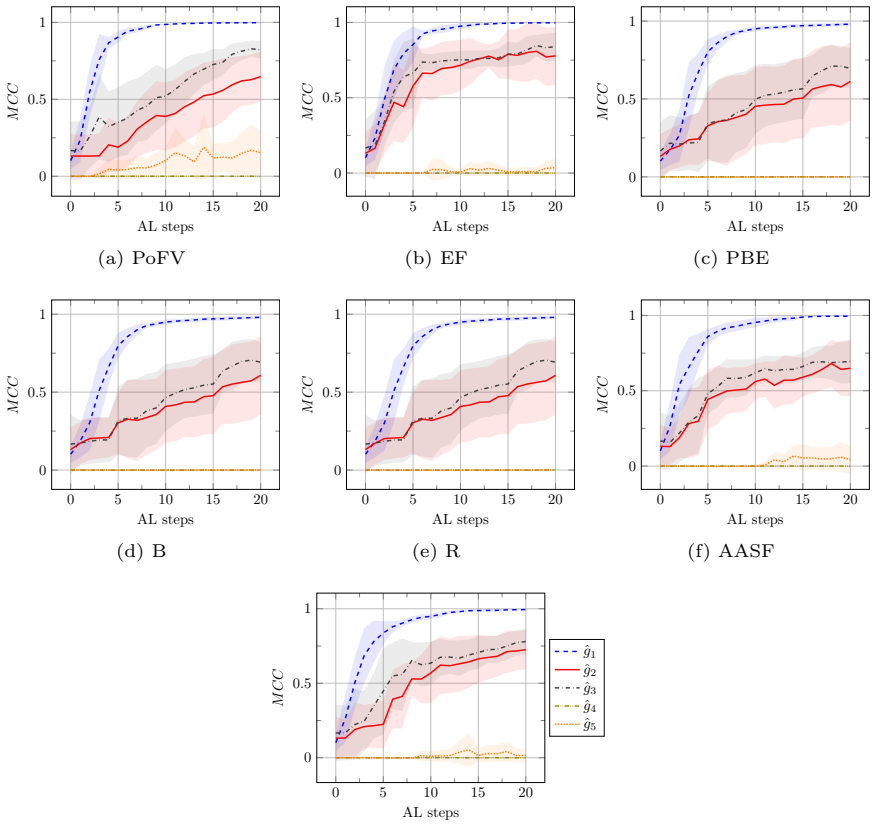
where $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$ and $2 \leq x_3 \leq 15$.

A.5 Speed Reducer (SR)

$$
\begin{aligned}
g_1(\mathbf{x}) &= 27 - x_1x_2^2x_3 \leq 0, \\
g_2(\mathbf{x}) &= 1.93 - \frac{x_2x_6^4x_3}{x_4^3} \leq 0, \\
g_3(\mathbf{x}) &= 1.93 - \frac{x_2x_7^4x_3}{x_5^3} \leq 0, \\
g_4(\mathbf{x}) &= x_2x_3 - 40 \leq 0, \\
g_5(\mathbf{x}) &= \frac{x_1}{x_2} - 12 \leq 0, \\
g_6(\mathbf{x}) &= 1.5x_6 - x_4 + 1.9 \leq 0, \\
g_7(\mathbf{x}) &= 1.1x_7 - x_5 + 1.9 \leq 0,
\end{aligned}
\tag{A5}
$$

where $2.6 \leq x_1 \leq 3.6$, $0.7 \leq x_2 \leq 0.8$, $17 \leq x_3 \leq 28$, $7.3 \leq x_4, x_5 \leq 8.3$, $2.9 \leq x_6 \leq 3.9$, and $5 \leq x_7 \leq 5.5$.

Appendix B Additional Results

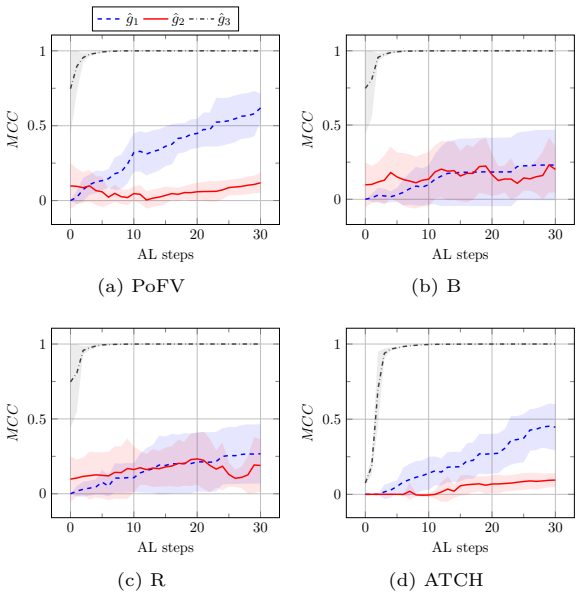


(a) PoFV (b) EF (c) PBE

(d) B (e) R (f) AASF

(g) ATCH

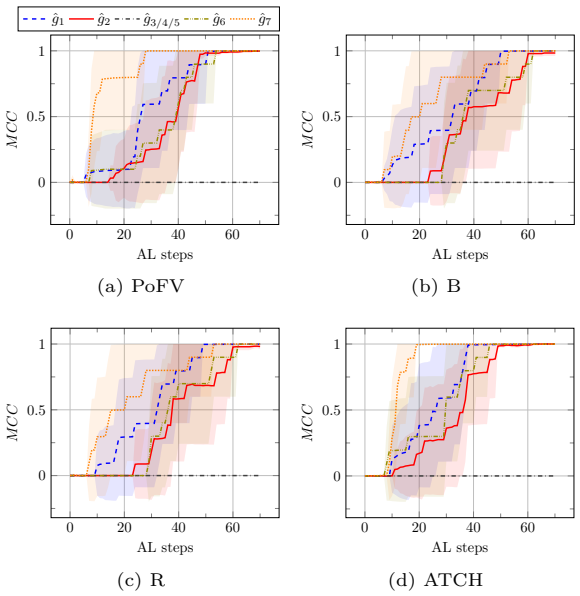
Fig. B1 Performance of the individual constraint models $\hat{g}_l$ in the Nowacki beam problem depending on the acquisition function. The figures show the MCC mean score and one standard deviation of 10 runs. EF manages to converge for all constraints relatively quickly. PBE, B and R fail to learn that g_5 is restrictive in the design space.



(a) PoFV (b) B

(c) R (d) ATCH

Fig. B2 Performance of the individual constraint models $\hat{g}_l$ in the tension/compression spring problem depending on the acquisition function. The figures show the MCC mean score and one standard deviation of 10 runs for the PoFV, B, R and ATCH acquisition functions. All acquisition functions converge the fastest on the most restrictive constraint.



(a) PoFV (b) B

(c) R (d) ATCH

Fig. B3 Performance of the individual constraint models $\hat{g}_l$ in the speed reducer problem depending on the acquisition function. The figures show the MCC mean score and one standard deviation of 10 runs for the PoFV, B, R and ATCH acquisition functions. Note that the results for constraints g_3, g_4 and g_5 coincide.

References

Ariyanto, M., Haryadi, G.D., Munadi, M., Ismail, R., Hendra, Z. (2018). Development of low-cost autonomous emergency braking system (AEBS) for an electric car. *2018 5th International Conference on Electric Vehicular Technology (ICEVT)* (pp. 167–171). IEEE.

Aultman, M., Wang, Z., Auza-Gutierrez, R., Duan, L. (2022). Evaluation of CFD methodologies for prediction of flows around simplified and complex automotive models. *Computers & Fluids*, *236*, 105297, https://doi.org/10.1016/j.compfluid.2021.105297

Azzimonti, D., Ginsbourger, D., Chevalier, C., Bect, J., Richet, Y. (2021). Adaptive design of experiments for conservative estimation of excursion sets. *Technometrics*, *63*(1), 13–26, https://doi.org/10.1080/00401706.2019.1693427

Bichon, B.J., Eldred, M.S., Swiler, L.P., Mahadevan, S., McFarland, J.M. (2008). Efficient global reliability analysis for nonlinear implicit performance functions. *AIAA journal*, *46*(10), 2459–2468, https://doi.org/10.2514/1.34321

Booth, A.S., Renganathan, S.A., Gramacy, R.B. (2025). Contour location for reliability in airfoil simulation experiments using deep gaussian processes. *The Annals of Applied Statistics*, *19*(1), 191–211, https://doi.org/10.1214/24-AOAS1951

Boughorbel, S., Jarray, F., El-Anbari, M. (2017). Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PloS one*, *12*(6), e0177678, https://doi.org/10.1371/journal.pone.0177678

Carrara, P., De Lorenzis, L., Stainier, L., Ortiz, M. (2020). Data-driven fracture mechanics. *Computer Methods in Applied Mechanics and Engineering*, *372*, 113390, https://doi.org/10.1016/j.cma.2020.113390

Chevalier, C., Bect, J., Ginsbourger, D., Vazquez, E., Picheny, V., Richet, Y. (2014). Fast parallel kriging-based stepwise uncertainty reduction with application to the identification of an excursion set. *Technometrics*, *56*(4), 455–465, https://doi.org/10.1080/00401706.2013.860918

Cohn, D.A., Ghahramani, Z., Jordan, M.I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research*, *4*, 129–145, https://doi.org/10.1613/jair.295

De Ath, G., Everson, R.M., Rahat, A.A., Fieldsend, J.E. (2021). Greed is good: Exploration and exploitation trade-offs in Bayesian optimisation. *ACM Transactions on Evolutionary Learning and Optimization*, *1*(1), 1–22, https://doi.org/10.1145/3425501

Engineering Toolbox (2013). *Cantilever Beams - Moments and Deflections.* Retrieved from https://www.engineeringtoolbox.com/cantilever-beams-d_1848.html (Accessed 2 March 2023)

Flores Ituarte, I., Panicker, S., Nagarajan, H.P., Coatanea, E., Rosen, D.W. (2023). Optimisation-driven design to explore and exploit the process–structure–property–performance linkages in digital manufacturing. *Journal of Intelligent Manufacturing*, *34*(1), 219–241, https://doi.org/10.1007/s10845-022-02010-2

Forrester, A., & Keane, A. (2009). Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, *45*(1-3), 50-79, https://doi.org/10.1016/j.paerosci.2008.11.001

Forrester, A., Sobester, A., Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide.* John Wiley & Sons.

Frazier, P.I. (2018). A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, , https://doi.org/10.48550/arXiv.1807.02811

Fuhg, J.N., Fau, A., Nackenhorst, U. (2021). State-of-the-art and comparative review of adaptive sampling methods for kriging. *Archives of Computational Methods in Engineering*, *28*(4), 2689–2747, https://doi.org/10.1007/s11831-020-09474-6

Gaudrie, D., Le Riche, R., Picheny, V., Enaux, B., Herbert, V. (2020). Targeting solutions in bayesian multi-objective optimization: sequential and batch versions. *Annals of Mathematics and Artificial Intelligence*, *88*(1), 187–212, https://doi.org/10.1007/s10472-019-09644-8

González, J., Dai, Z., Damianou, A., Lawrence, N.D. (2017). Preferential Bayesian optimization. *International Conference on Machine Learning* (pp. 1282–1291).

Gotovos, A., Casati, N., Hitz, G., Krause, A. (2013). Active learning for level set estimation. *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence* (pp. 1344–1350). AAAI Press.

Hernández-Lobato, J.M., Gelbart, M., Hoffman, M., Adams, R., Ghahramani, Z. (2015). Predictive entropy search for Bayesian optimization with unknown constraints. *International Conference on Machine Learning* (pp. 1699–1707).

Houlsby, N., Huszár, F., Ghahramani, Z., Lengyel, M. (2011). Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, ,

Jones, D.R., Schonlau, M., Welch, W.J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, *13*(4), 455–492, https://doi.org/10.1023/A:1008306431147

Kaintura, A., Foss, K., Couckuyt, I., Dhaene, T., Zografos, O., Vaysset, A., Soree, B. (2018). Machine learning for fast characterization of magnetic logic devices. *2018 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS)* (pp. 1–3). IEEE.

Knowles, J. (2006). ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE transactions on evolutionary computation*, *10*(1), 50–66, https://doi.org/0.1109/TEVC.2005.851274

Knudde, N., Couckuyt, I., Shintani, K., Dhaene, T. (2019). Active Learning for Feasible Region Discovery. *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)* (pp. 567–572). IEEE.

Kumar, A., Wu, G., Ali, M.Z., Mallipeddi, R., Suganthan, P.N., Das, S. (2020). A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation*, *56*, 100693, https://doi.org/10.1016/j.swevo.2020.100693

Leite Richardson, F., De Ath, G., Chugh, T. (2024). Is greed still good in multi-objective Bayesian optimisation? *Proceedings of the genetic and evolutionary computation conference companion* (pp. 2103–2106).

Lemieux, C. (2009). *Monte carlo and quasi-monte carlo sampling*. Springer Science & Business Media.

Letham, B., Guan, P., Tymms, C., Bakshy, E., Shvartsman, M. (2022). Look-ahead acquisition functions for Bernoulli level set estimation. *International Conference on Artificial Intelligence and Statistics* (pp. 8493–8513).

Luque, A., Carrasco, A., Martín, A., de las Heras, A. (2019). The impact of class imbalance in classification performance metrics based on the binary confusion

matrix. *Pattern Recognition*, *91*, 216–231, https://doi.org/10.1016/j.patcog.2019.02.023

Mandow, L., Martín-Albob, S., Perez-de-la Cruza, J.-L. (2023). Multi-objective bandit algorithms with Chebyshev scalarization.

Mardia, K.V., & Marshall, R.J. (1984). Maximum likelihood estimation of models for residual covariance in spatial regression. *Biometrika*, *71*(1), 135-146, https://doi.org/10.1093/biomet/71.1.135

Marques, A.N., Lam, R.R., Willcox, K.E. (2018). Contour location via entropy reduction leveraging multiple information sources. *Proceedings of the 32nd international conference on neural information processing systems* (p. 5223–5233). Curran Associates Inc.

Matthews, A.G.d.G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., ... Hensman, J. (2017). GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, *18*(40), 1-6,

Menzel, T., Bagschik, G., Maurer, M. (2018). Scenarios for development, test and validation of automated vehicles. *2018 IEEE Intelligent Vehicles Symposium (IV)* (pp. 1821–1827). IEEE.

Miettinen, K. (1999). *Nonlinear multiobjective optimization* (Vol. 12). Springer Science & Business Media.

Miettinen, K., & Mäkelä, M.M. (2002). On scalarizing functions in multiobjective optimization. *OR spectrum*, *24*, 193–213, https://doi.org/10.1007/s00291-001-0092-9

Moss, H., Picheny, V., Stojic, H., Ober, S.W., Artemev, A., Paleyes, A., ... others (2024). Trieste: Efficiently exploring the depths of black-box functions with TensorFlow. *Neurips 2024 workshop on bayesian decision-making and uncertainty.*

Nikova, I., Dhaene, T., Couckuyt, I. (2023). Cost-aware active learning for feasible region identification. *Proceedings of the Companion Conference on Genetic and Evolutionary Computation* (pp. 2286–2288).

Nowacki, H. (2005). Modelling of design decisions for cad. *Computer Aided Design Modelling, Systems Engineering, CAD-Systems: CREST Advanced Course*

Darmstadt, 8.–19. September 1980, 177–223,

Park, J.-S. (1994). Optimal Latin-hypercube designs for computer experiments. *Journal of Statistical Planning and Inference*, *39*(1), 95–111, https://doi.org/10.1016/0378-3758(94)90115-5

Picheny, V., Ginsbourger, D., Roustant, O., Haftka, R.T., Kim, N.-H. (2010). Adaptive designs of experiments for accurate approximation of a target region. *Journal of Mechanical Design*, *132*(7), 071008, https://doi.org/10.1115/1.4001873

Qing, J., Knudde, N., Couckuyt, I., Dhaene, T., Shintani, K. (2020). Batch Bayesian active learning for feasible region identification by local penalization. *2020 Winter Simulation Conference (WSC)* (pp. 2779–2790). IEEE.

Qing, J., Knudde, N., Couckuyt, I., Spina, D., Dhaene, T. (2020). Bayesian active learning for electromagnetic structure design. *2020 14th European Conference on Antennas and Propagation (EuCAP)* (pp. 1–5). IEEE.

Qing, J., Knudde, N., Garbuglia, F., Spina, D., Couckuyt, I., Dhaene, T. (2022). Adaptive sampling with automatic stopping for feasible region identification in engineering design. *Engineering with Computers*, *38*(S3), 1955–1972, https://doi.org/10.1007/s00366-021-01341-7

Rahat, A., & Wood, M. (2020). On bayesian search for the feasible space under computationally expensive constraints. *International conference on machine learning, optimization, and data science* (pp. 529–540).

Ranjan, P., Bingham, D., Michailidis, G. (2008). Sequential Experiment Design for Contour Estimation From Complex Computer Codes. *Technometrics*, *50*(4), 527–541, https://doi.org/10.1198/004017008000000541

Rasmussen, C.E., & Williams, C.K.I. (2008). *Gaussian processes for machine learning* (3. print ed.). MIT Press.

Ray, T. (2003). Golinski's speed reducer problem revisited. *AIAA journal*, *41*(3), 556–558,

Rojas Gonzalez, S., Branke, J., Van Nieuwenhuyse, I. (2025). Bi-objective ranking and selection using stochastic kriging. *European Journal of Operational Research*, *322*(2), 599–614, https://doi.org/10.1016/j.ejor.2024.11.008

Rojas Gonzalez, S., Jalali, H., Van Nieuwenhuyse, I. (2020). A multiobjective stochastic simulation optimization algorithm. *European Journal of Operational Research*, *284*(1), 212–226, https://doi.org/10.1016/j.ejor.2019.12.014

Rojas Gonzalez, S., & Van Nieuwenhuyse, I. (2020). A survey on kriging-based infill algorithms for multiobjective simulation optimization. *Computers & Operations Research*, *116*, 104869, https://doi.org/10.1016/j.cor.2019.104869

Santner, T.J., Williams, B.J., Notz, W.I., Santner, T.J., Williams, B.J., Notz, W.I. (2018). Space-filling designs for computer experiments. In (pp. 145–200). Springer.

Singh, P., van der Herten, J., Deschrijver, D., Couckuyt, I., Dhaene, T. (2017). A sequential sampling strategy for adaptive classification of computationally expensive data. *Structural and Multidisciplinary Optimization*, *55*(4), 1425–1438, https://doi.org/10.1007/s00158-016-1584-1

Sobek II, D.K. (1996). A set-based model of design. *mechanical Engineering*, *118*(7), 78,

van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, *9*(86), 2579–2605,

Wotawa, F., Klück, F., Zimmermann, M., Nica, M., Felbinger, H., Tao, J., Li, Y. (2021). Recent verification and validation methodologies for advanced driver-assistance systems. *Autonomous driving and advanced driver-assistance systems (adas)* (pp. 295–318). CRC Press.

Xiao, Z., Tong, H., Qu, R., Xing, H., Luo, S., Zhu, Z., ... Feng, L. (2023). CapMatch: Semi-supervised contrastive transformer capsule with feature-based knowledge distillation for human activity recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 1-15, https://doi.org/10.1109/TNNLS.2023.3344294

Xiao, Z., Xu, X., Xing, H., Zhao, B., Wang, X., Song, F., ... Feng, L. (2024). DTCM: Deep transformer capsule mutual distillation for multivariate time series classification. *IEEE Transactions on Cognitive and Developmental Systems*, *16*(4), 1445-1461, https://doi.org/10.1109/TCDS.2024.3370219

Xin, B., Chen, L., Chen, J., Ishibuchi, H., Hirota, K., Liu, B. (2018). Interactive multiobjective optimization: A review of the state-of-the-art. *IEEE Access*, *6*, 41256-41279, https://doi.org/10.1109/ACCESS.2018.2856832

Xiong, Y., Duong, P.L.T., Wang, D., Park, S.-I., Ge, Q., Raghavan, N., Rosen, D.W. (2019). Data-driven design space exploration and exploitation for design for additive manufacturing. *Journal of Mechanical Design*, *141*(10), 101101, https://doi.org/10.1115/1.4043587

Yang, X., Mi, C., Deng, D., Liu, Y. (2019). A system reliability analysis method combining active learning Kriging model with adaptive size of candidate points. *Structural and Multidisciplinary Optimization*, *60*(1), 137–150, https://doi.org/10.1007/s00158-019-02205-x

Young, D., Vondrasek, B., Czabaj, M.W. (2025). Machine learning guided design of experiments to accelerate exploration of a material extrusion process parameter space. *Journal of Intelligent Manufacturing*, *36*(1), 491–508, https://doi.org/10.1007/s10845-023-02255-5

Zhang, X., Tao, J., Tan, K., Törngren, M., Sánchez, J.M.G., Ramli, M.R., ... others (2021). Finding critical scenarios for automated driving systems: A systematic literature review. *arXiv preprint arXiv:2110.08664*, ,

Table B1 Results of the final constraint models for each problem, trained on $N + q$ samples. For each acquisition method, the mean ± standard deviation of MCC of all 10 runs are shown. The best values for each model are underlined.

		MCC						
	$\hat{g}_l$	PoFV	EF	PBE	B	R	AASF	ATCH
CB	$\hat{g}_1$	$\underline{0.998 \pm 0.002}$	$\underline{0.998 \pm 0.002}$	0.997 ± 0.002	$\underline{0.998 \pm 0.002}$	$\underline{0.998 \pm 0.002}$	0.997 ± 0.002	0.997 ± 0.004
	$\hat{g}_2$	0.97 ± 0.01	$\underline{0.996 \pm 0.004}$	0.994 ± 0.003	0.993 ± 0.003	0.994 ± 0.003	0.95 ± 0.04	0.97 ± 0.02
PFS	$\hat{g}_1$	0.991 ± 0.008	$\underline{1.0 \pm 0.0}$	0.996 ± 0.002	$1.0 \pm 4e{-4}$	$1.0 \pm 4e{-4}$	$1.0 \pm 4e{-4}$	$1.0 \pm 4e{-4}$
	$\hat{g}_2$	$0.999 \pm 8e{-4}$	$\underline{1.0 \pm 0.0}$	0.999 ± 0.001	$\underline{1.0 \pm 0.0}$	$\underline{1.0 \pm 0.0}$	$1.0 \pm 5e{-4}$	$1.0 \pm 4e{-4}$
NB	$\hat{g}_1$	$\underline{0.998 \pm 8.7e{-4}}$	0.997 ± 0.001	0.980 ± 0.01	0.980 ± 0.01	0.980 ± 0.01	0.995 ± 0.002	0.994 ± 0.004
	$\hat{g}_2$	0.65 ± 0.16	$\underline{0.78 \pm 0.20}$	0.61 ± 0.25	0.61 ± 0.25	0.61 ± 0.25	0.65 ± 0.18	0.73 ± 0.13
	$\hat{g}_3$	0.82 ± 0.06	$\underline{0.84 \pm 0.09}$	0.70 ± 0.13	0.69 ± 0.13	0.69 ± 0.13	0.69 ± 0.15	0.78 ± 0.09
	$\hat{g}_4$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	$\hat{g}_5$	$\underline{0.15 \pm 0.14}$	0.04 ± 0.06	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.04 ± 0.09	0.01 ± 0.05
TCS	$\hat{g}_1$	$\underline{0.62 \pm 0.10}$	0.40 ± 0.48	0.24 ± 0.26	0.23 ± 0.24	0.27 ± 0.20	0.52 ± 0.12	0.45 ± 0.15
	$\hat{g}_2$	0.12 ± 0.07	$\underline{0.27 \pm 0.23}$	0.19 ± 0.19	0.20 ± 0.16	0.19 ± 0.18	0.09 ± 0.04	0.09 ± 0.05
	$\hat{g}_3$	$\underline{1.0 \pm 0.0}$	$\underline{1.0 \pm 0.0}$	$1.0 \pm 4e{-4}$	$\underline{1.0 \pm 0.0}$	$\underline{1.0 \pm 0.0}$	$\underline{1.0 \pm 0.0}$	$\underline{1.0 \pm 0.0}$
SR	$\hat{g}_1$	0.998 ± 0.002	1.0 ± 0.002	0.999 ± 0.001	0.999 ± 0.002	0.999 ± 0.002	1.0 ± 0.001	$\underline{1.0 \pm 0.0}$
	$\hat{g}_2$	0.998 ± 0.006	0.996 ± 0.008	0.988 ± 0.01	0.983 ± 0.02	0.981 ± 0.02	1.0 ± 0.01	$\underline{1.0 \pm 0.0}$
	$\hat{g}_3$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	$\hat{g}_4$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	$\hat{g}_5$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	$\hat{g}_6$	0.999 ± 0.001	0.999 ± 0.001	$0.999 \pm 8e{-4}$	0.999 ± 0.001	0.999 ± 0.001	$\underline{1.0 \pm 8e{-4}}$	$\underline{1.0 \pm 8e{-4}}$
	$\hat{g}_7$	$\underline{1.0 \pm 2e{-4}}$	$\underline{1.0 \pm 2e{-4}}$	$\underline{1.0 \pm 2e{-4}}$	$1.0 \pm 4e{-4}$	$1.0 \pm 4e{-4}$	$1.0 \pm 3e{-4}$	$1.0 \pm 3e{-4}$
AEB	$\hat{g}_1$	0.87 ± 0.02	0.88 ± 0.03	0.89 ± 0.02	$\underline{0.89 \pm 0.01}$	$\underline{0.89 \pm 0.01}$	0.87 ± 0.02	0.88 ± 0.02

Table B2 Results of the final constraint models for each problem, trained on $N + q$ samples. For each acquisition method, the mean ± standard deviation of NLL of all 10 runs are shown. The best values for each model are underlined. Note that the NLL value is influenced by the range of the constraint function values.

		NLL						
	$\hat{g}_l$	PoFV	EF	PBE	B	R	AASF	ATCH
CB	$\hat{g}_1$	0.41 ± 0.04	0.45 ± 0.03	0.51 ± 0.04	0.48 ± 0.03	0.48 ± 0.04	$\underline{0.003 \pm 0.004}$	0.004 ± 0.01
	$\hat{g}_2$	$\underline{0.39 \pm 0.04}$	0.40 ± 0.04	0.47 ± 0.05	0.44 ± 0.04	0.44 ± 0.04	0.64 ± 0.003	0.64 ± 0.002
PFS	$\hat{g}_1$	$\underline{0.33 \pm 0.03}$	0.36 ± 0.03	0.38 ± 0.02	0.42 ± 0.02	0.42 ± 0.02	$0.52 \pm 6e{-6}$	$0.52 \pm 9e{-5}$
	$\hat{g}_2$	$\underline{0.30 \pm 0.05}$	0.56 ± 0.04	0.40 ± 0.03	0.51 ± 0.05	0.51 ± 0.05	$0.49 \pm 5e{-6}$	$0.49 \pm 6e{-6}$
NB	$\hat{g}_1$	$\underline{0.004 \pm 0.004}$	0.006 ± 0.008	1.04 ± 1.26	1.03 ± 1.26	1.04 ± 1.26	0.03 ± 0.02	0.16 ± 0.37
	$\hat{g}_2$	0.44 ± 0.29	0.41 ± 0.50	$\underline{0.18 \pm 0.15}$	0.21 ± 0.25	0.21 ± 0.25	0.82 ± 0.76	0.39 ± 0.33
	$\hat{g}_3$	$\underline{3.3 \pm 2.8}$	4.4 ± 5.2	3.6 ± 2.8	4.1 ± 4.3	4.1 ± 4.3	12.6 ± 11.5	4.7 ± 4.7
	$\hat{g}_4$	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0
	$\hat{g}_5$	$1.1e2 \pm 2.7e2$	$2.6e2 \pm 4.3e2$	$1.0e2 \pm 64.3$	$\underline{93.2 \pm 67.6}$	$\underline{93.2 \pm 67.6}$	$1.8e2 \pm 2.2e2$	$4.4e2 \pm 9.6e2$
TCS	$\hat{g}_1$	0.60 ± 0.03	0.59 ± 0.11	0.69 ± 0.07	0.71 ± 0.06	0.71 ± 0.06	$\underline{0.34 \pm 0.01}$	$\underline{0.34 \pm 0.01}$
	$\hat{g}_2$	0.85 ± 0.05	$\underline{0.62 \pm 0.13}$	0.73 ± 0.06	0.75 ± 0.07	0.76 ± 0.07	2.42 ± 1.40	2.84 ± 1.73
	$\hat{g}_3$	0.34 ± 0.02	$\underline{0.30 \pm 0.04}$	0.38 ± 0.07	0.34 ± 0.06	0.34 ± 0.06	$0.53 \pm 1e{-5}$	$0.53 \pm 3e{-5}$
SR	$\hat{g}_1$	0.74 ± 0.02	0.40 ± 0.04	0.74 ± 0.06	0.73 ± 0.05	0.73 ± 0.05	$0.02 \pm 1e{-5}$	$\underline{0.02 \pm 8e{-6}}$
	$\hat{g}_2$	0.77 ± 0.02	0.61 ± 0.04	0.79 ± 0.04	0.78 ± 0.05	0.78 ± 0.05	$\underline{0.12 \pm 7e{-5}}$	$0.12 \pm 4e{-5}$
	$\hat{g}_3$	0.64 ± 0.03	0.54 ± 0.03	0.78 ± 0.03	0.76 ± 0.04	0.76 ± 0.04	$\underline{2e{-8} \pm 5e{-11}}$	$\underline{2e{-8} \pm 5e{-11}}$
	$\hat{g}_4$	0.75 ± 0.02	1.34 ± 0.09	0.82 ± 0.05	0.85 ± 0.06	0.85 ± 0.06	$\underline{9e{-10} \pm 6e{-13}}$	$\underline{9e{-10} \pm 6e{-13}}$
	$\hat{g}_5$	0.76 ± 0.03	0.92 ± 0.07	0.86 ± 0.06	0.78 ± 0.05	0.78 ± 0.05	$\underline{4e{-4} \pm 2e{-8}}$	$\underline{4e{-4} \pm 2e{-8}}$
	$\hat{g}_6$	0.70 ± 0.02	0.76 ± 0.05	0.69 ± 0.04	0.69 ± 0.04	0.69 ± 0.05	$\underline{0.39 \pm 9e{-6}}$	$\underline{0.39 \pm 9e{-6}}$
	$\hat{g}_7$	0.52 ± 0.02	$\underline{0.37 \pm 0.02}$	0.43 ± 0.01	0.43 ± 0.01	0.42 ± 0.01	$0.56 \pm 2e{-5}$	$0.56 \pm 1e{-5}$
AEB	$\hat{g}_1$	0.60 ± 0.06	$\underline{0.21 \pm 0.04}$	0.33 ± 0.04	0.33 ± 0.04	0.34 ± 0.04	0.32 ± 0.02	0.35 ± 0.03