

Tuning particle accelerators with safety constraints using Bayesian optimization

Johannes Kirschner¹,* Mojmir Mutný, and Andreas Krause¹

Department of Computer Science, ETH Zurich, 8092 Zurich, Switzerland

Jaime Coello de Portugal², Nicole Hiller, and Jochem Snuverink²†

Paul Scherrer Institut, 5232 Villigen PSI, Switzerland

 (Received 4 May 2021; revised 20 December 2021; accepted 18 March 2022; published 28 June 2022)

Tuning machine parameters of particle accelerators is a repetitive and time-consuming task that is challenging to automate. While many off-the-shelf optimization algorithms are available, in practice their use is limited because most methods do not account for safety-critical constraints in each iteration, such as loss signals or step-size limitations. One notable exception is safe Bayesian optimization, which is a data-driven tuning approach for global optimization with noisy feedback. We propose and evaluate a step-size limited variant of safe Bayesian optimization on two research facilities of the PSI: (a) the SwissFEL and (b) HIPA. We report promising experimental results on both machines, tuning up to 16 parameters subject to 224 constraints.

DOI: [10.1103/PhysRevAccelBeams.25.062802](https://doi.org/10.1103/PhysRevAccelBeams.25.062802)

I. INTRODUCTION

Particle accelerators are complex machines consisting of many elements and are typically built according to an idealized design. However, in reality, there are always systematic and time-varying errors that will reduce the performance with respect to the design performance. These errors need to be corrected by parameter tuning to achieve the optimal performance. Therefore, empirical parameter tuning is a required and often reoccurring task for any particle accelerator. Common tuning objectives include the beam shape, beam trajectory, beam loss minimization, or a combination of multiple objectives.

In this paper, we present a method that allows for safe optimization of particle accelerators. Our approach follows recent work by Kirschner *et al.* [1] on safe Bayesian optimization. The main idea behind Bayesian optimization is to compute regression estimates of the target and loss functions using the data collected during optimization [2,3]. The estimates and associated statistical uncertainty are then used to systematically explore the parameter space, while guaranteeing that the query points are safe. Our main contribution is an experimental evaluation of safe Bayesian optimization on the High Intensity Proton Accelerator

(HIPA) and the Swiss Free Electron Laser (SwissFEL). Further, we address practically relevant challenges in the context of tuning particle accelerators, including step-size constraints and user feedback for hyper-parameter tuning.

Tuning of accelerators is a challenging task for several reasons. Typically, simulation models are not accurate enough to allow off-line optimization, and the parameters have to be tuned empirically by evaluating different settings on the machine. Among many other considerations, the following points are of particular importance, including on the machines we used for experimentation.

Safety.—Beam optimization is a delicate task because improper settings can cause beam losses that potentially damage the machine. While modern machines have safety measures implemented that are designed to prevent hardware damage by stopping the beam, it is desirable to avoid triggering such safety measures. On one hand, restarting the machine is time consuming and breaching the safety level is often considered off limit. Further, an effective method should concentrate on sampling feasible points for fast convergence.

Step-size control.—Classical optimization methods are often based on the principle of making small adjustments to an incumbent solution, such as a small step in the gradient direction. Global search methods are an attractive alternative since they do not require gradient estimates and make less restrictive assumptions such as convexity of the target functions. Popular choices include Nelder-Mead [4], CMA-ES [5] and Particle-Swarm optimization [6]. However, these methods do *not* impose a step-size constraint. Large changes to input parameters of particle accelerators can be

*jkirschn@ualberta.ca
†jochem.snuverink@psi.ch

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

problematic because feedback systems are often in place to stabilize the beam, which might not be able to follow abrupt changes. This limitation can be circumvented by slowly changing the machine parameters to the requested target values. Such a process, however, increases the measurement time on the machine and slows down the optimization.

User feedback.—Optimization methods are usually not designed to provide user feedback other than the progress on the target value. In particular, it is difficult to monitor the optimization trajectory and the user essentially has to trust the method to not induce an unwanted state on the machine.

As our main contribution and addressing the challenges outlined above, we demonstrate feasibility and efficacy of *safe Bayesian optimization* for parameter tuning on particle accelerators. Our approach is based on the LineBO method of Kirschner *et al.* [1], and we develop a version of the same method with step-size control. The LineBO approach combines Bayesian optimization with a line-search technique, thereby allowing to scale the method to high-dimensional settings. We compare performance with the nonsafe global search methods, CMA-ES [5] and NELDERMead [4]. Our claims are corroborated with experimental data on two machines: The High Intensity Proton Accelerator (HIPA, introduced in Sec. IB) and the Swiss Free Electron Laser (SwissFEL, introduced in Sec. IC).

A. A brief overview on optimization methods

For many accelerators, parameter tuning has been automated to achieve significant speedups compared to manual operator tuning [7–9]. Since only point evaluations of the objective function are available, one has to rely on so-called *zero-order* or *black-box* optimization methods. There are a large variety of optimization methods in use. Due to their simplicity, the NELDERMead (Simplex) [4] algorithm and random walk optimizers have become popular choices to assist operator tuning [8,10,11]. Further methods that have been successfully applied on accelerators include Extremum Seeking [12–14] and robust conjugate direction search (RCDS) [15,16]. The advantage of local descent methods is their simplicity, although convergence guarantees only apply under strong assumptions like convexity. Other gradient-based optimization algorithms are more difficult to deploy as additional samples are required to estimate the gradient, e.g. via finite differences.

Another class of zero-order optimization methods are evolutionary algorithms such as Particle-Swarm optimization [16] and genetic algorithms [17–19]. A popular choice due to its simplicity is CMA-ES [5], which is shown to achieve competitive performance on standardized benchmarks [e.g., [20]], although we are not aware of an application on particle accelerators.

As an alternative, *Bayesian optimization* [3,21] has recently gained interest in the accelerator community [22–25]. Bayesian optimization is a framework for zero-order optimization that deals with observational noise in a principled way, allows the use of prior data or knowledge about the objective, and comes with theoretical convergence guarantees for some variants [21].

When tuning particle accelerators, there are often safety-relevant constraints, e.g., physical or safety limits that cannot be violated, of which the typical optimization algorithm is unaware. For example, at the Swiss Free Electron Laser the pulse energy should be maximized while keeping low undulator losses. At the High Intensity Proton Accelerator the situation is more rigid: It is important to keep the overall losses as low as possible while not exceeding any of the individual loss limits; violating the limit of a single loss monitor triggers the safety system, causes downtime, or possible damages to the machine.

It is natural to formulate this setup as a *constrained optimization* problem, where the goal is to optimize a target function subject to a feasibility condition. Established algorithms for the constrained setting include sequential quadratic programming [26] and interior point methods [27]. Variants for constrained Bayesian optimization have been proposed in, e.g., [28–30]. Gradient descent methods can be applied in the constrained setting via a Lagrangian (primal-dual) formulation of the objective that penalize infeasible points [31]. We emphasize that the methods for constrained optimization ought to return a feasible *final* solution but evaluating *infeasible* parameters is tolerated during optimization; hence these methods do *not* address the safety requirement for tuning particle accelerators. This is opposed to *safe optimization*, where the complete optimization trajectory has to satisfy the feasibility condition. To achieve *safe optimization* that guarantees feasibility for all evaluation points with high probability, Sui *et al.* [32] proposed *safe Bayesian optimization*. We describe this method and extensions in detail in Sec. II. A variant of safe Bayesian optimization was evaluated on SwissFEL in previous work [1].

B. HIPA

The high intensity proton accelerator (HIPA) provides the primary beams to Paul Scherrer Institut’s (PSI’s) versatile experimental facilities. HIPA generates a proton beam with 590 MeV kinetic energy and presently 1.3 MW average beam power [33]. This corresponds to a proton current of about 2.2 mA. The HIPA accelerator consists of a Cockcroft-Walton preaccelerator and a chain of two isochronous cyclotrons, the injector II and the ring cyclotron. The beam is produced in continuous wave (cw) mode at a frequency of 50.6 MHz. The high intensity proton beam is used to produce pions and muons by interaction with two graphite targets. After collimation the remaining beam with roughly 1 MW is then used to produce neutrons in a

spallation target. A pulsed source for ultracold neutrons is also in operation [34].

In practice, the performance is limited by the beam losses at the extraction of the ring cyclotron and in the high energy beam line downstream. In the ring cyclotron longitudinal space charge leads to an increase in energy spread that transforms into transverse beam tails, which reduces the turn separation at extraction [33]. Furthermore, significant emittance growth occurs in the graphite targets and results in unavoidable collimation loss in specially shielded collimators. These losses are controlled by several rf and orbit feedback systems to keep the machine stable and by empirical operator tuning.

C. SwissFEL

The Swiss free electron laser (SwissFEL) is PSI’s state of the art source for ultrashort x-ray pulses [35,36]. A photocathode is used to generate short electron pulses which are then brought up to 6 GeV by a series of linear accelerator modules and compressed down to a pulse length of only a few femtoseconds in two bunch compressor chicanes. Inside an undulator line, the electron pulses self-modulate and emit x-ray pulses with laserlike properties by a process called self-amplified spontaneous emission. These ultrashort x-ray pulses are then used for various user experiments (e.g., pump probe experiments, x-ray spectroscopy). At SwissFEL, there are two undulator lines—Aramis and Athos—that can be used to generate photons for different end stations where the user experiments take place.

The settings of the accelerator (e.g., the photon energy or the pulse compression) are often adjusted to fit requests of the users. Typically subsequent parameter optimization is necessary in order to maximize the photon pulse energy at the experiment. The pulse energy is measured with a gas detector [37] that provides readings on a shot-to-shot basis. When performing the optimization, various tuning parameters (~ 100) can have a strong impact on the resulting beam quality, but also on the losses that can cause radiation damage to delicate equipment. Especially, in the undulator region, losses should be kept low while tuning on the SwissFEL output, to avoid long-term damage due to demagnetization of the permanent magnets inside the undulators.

II. BAYESIAN OPTIMIZATION WITH CONSTRAINTS

Bayesian optimization (BO) [2,3,21] is a flexible, data-driven approach for global optimization with noisy feedback. The main idea is to fit a statistical regression model on the target function, using the data collected during optimization, and possibly, any available prior data. The next evaluation point is then chosen to reduce the model uncertainty about the unknown optimal solution. The approach was shown highly effective in many real-world applications, including optimization of particle accelerators

[1,23,38]. Variants for safety-critical tuning were developed by [32,39,40], where the algorithm learns a model of additional constraints on-line. The constraint models are used to estimate a feasible “safe” set of parameters. The method then only evaluates such safe inputs and thereby ensures that no query point violates the safety constraints (with high probability).

A. Line Bayesian optimization

In the following, we introduce a variant of safe Bayesian optimization developed in [1]. The main idea is to alternate between two phases (see Fig. 1). First, data is collected within a small ball around the current incumbent solution. Then a line search is performed in the direction of the (predicted) largest increase. In both phases, the data points are chosen to maximize an *upper-confidence bound (UCB)* of the regression estimate, which is known to be an effective search surrogate [41]. This procedure is repeated until a satisfactory solution is found, or the search budget is exhausted. The intuition for these choices is twofold: First, by concentrating samples on a one-dimensional domain or a small enough ball we obtain reliable function estimates with few samples in a local region of interest. Second, determining the acquisition points is computationally much simpler compared to standard Bayesian optimization, which requires to solve a nonconvex optimization problem over the full input domain in each round.

To formally introduce the algorithm, we write the tuning objective (maximized) and l constraints as follows:

$$\begin{aligned} & g_1(x) \leq 0 \\ \max_{x \in \mathcal{X}} f(x) \text{ s.t. } & \vdots \\ & g_l(x) \leq 0. \end{aligned}$$

Here, $f(x)$ is the unknown target function with input domain $\mathcal{X} \subset \mathbb{R}^d$, e.g., $\mathcal{X} = \{x \in \mathbb{R}^d : 0 \leq x_i \leq 1\}$,

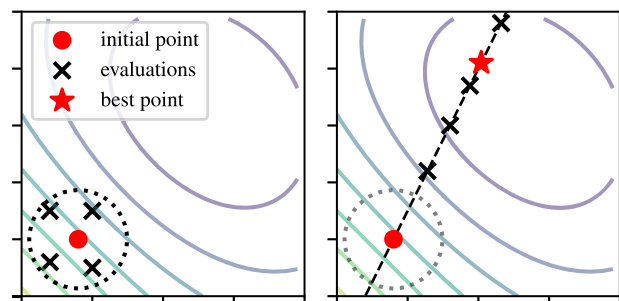


FIG. 1. The A-LineBO algorithm alternates between a local search (left; line 2 to 8 in Algorithm 1) and a line search (right; line 9 to 16 in Algorithm 1). On the left the search domain is restricted to a ball centered around the initial point. On the right, the search domain is chosen as a line in the direction of the estimated steepest ascent. The procedure is repeated at the best point found during the line search.

corresponding to the set of d tuning parameters on an allowed range. The functions $g_1(x), \dots, g_l(x)$ are constraints that need to be satisfied during operation, such as a loss limit. Note that the $g_1(x), \dots, g_l(x)$ are assumed to be unknown initially [like the target $f(x)$], and the only way of accessing the constraint functions is by evaluating a parameter on the machine. Provided an input parameter $x_t \in \mathcal{X}$ at step t , the machine interface [denoted $\text{MACHINE-EVAL}(x_t)$] returns a set of noisy measurements,

$$\begin{aligned} y_t &= f(x_t) + \epsilon_t \\ z_{t,1} &= g_1(x_t) + \xi_{t,1} \\ &\vdots \\ z_{t,l} &= g_l(x_t) + \xi_{t,l}, \end{aligned}$$

where ϵ_t and $\xi_{t,1}, \dots, \xi_{t,l}$ is measurement noise. In practice, we average multiple evaluations to reduce the noise variance. Also note that the constraint limit is zero without loss of generality, because we can always redefine a constraint $g_i(x) \leq c$ as $\tilde{g}_i(x) = g_i(x) - c \leq 0$. We say that an evaluation procedure is *safe* if it guarantees that with very high probability, all evaluation points satisfy $g_i(x_t) \leq 0$ for all $i = 1, \dots, l$ and time steps $t \geq 1$.

The next step is to compute a regression model given the available data $\mathcal{D}_t = \{(x_s, y_s, z_{s,1}, \dots, z_{s,l})\}_{s=1}^t$. In general, this is a user choice which we abstractly encapsulate in the $\text{REGRESSION}(\cdot)$ oracle function. Given the data \mathcal{D}_t , the regression oracle produces a model estimate \hat{m}_t . We assume that the model provides the following estimates for the target function and the constraints ($h \in \{f, g_1, \dots, g_l\}$): (i) $\text{MEAN}_h(\hat{m}_t, x)$ —the mean estimate of the target function evaluated at x , typically a consistent estimator of $h(x)$; (ii) $\text{UCB}_h(\hat{m}_t, x, \delta)$ —an upper confidence bound (UCB) on the true function, which takes a confidence parameter $\delta \in [0, 1]$ and satisfies:

$$\mathbb{P}[\forall x \in \mathcal{X}: h(x) \leq \text{UCB}_h(\hat{m}_t, x, \delta)] \geq 1 - \delta,$$

where the randomness is due to noisy evaluations; (iii) $\text{LCB}_h(\hat{m}_t, x, \delta)$, a lower confidence bound (LCB) on the true function, defined analogously to UCB_h ; and (iv) $\text{CW}_h(\hat{m}_t, x, \delta) \stackrel{\text{def}}{=} \text{UCB}_h(\hat{m}_t, x, \delta) - \text{LCB}_h(\hat{m}_t, x, \delta)$, the width of the confidence band at $x \in \mathcal{X}$.

The most common regression model used in Bayesian optimization is Gaussian process (GP) regression (or kernel least squares) [42], which we also use in our experiments. A brief, illustrative introduction to GP regression and the exact specification of our regression function is given in the Appendix A.

Finally, Bayesian optimization acquires points that are targeted to reduce the model uncertainty of the true maximizer $x^* = \arg \max_{x \in \mathcal{X}} f(x)$. To account for the safety

constraints, at any step t , the acquisition is restricted to the *safe set* with margin $\tau \geq 0$,

$$\mathcal{S}_t^\tau = \{x \in \mathcal{X}: \max_{i \in [l]} \text{UCB}_{g_i}(\hat{m}_t, x, \delta) \leq -\tau\}.$$

The safe set is defined as the subset of the domain \mathcal{X} where with probability at least $1 - \delta$, the model does not predict a constraint violation with margin τ . A larger safety margin makes the algorithm more conservative. Also note that the model predicts the average constraint level for a given input parameter. Therefore a positive safety margin is needed when the constraint measurements are noisy and the goal is to avoid constraint violations with high probability (i.e., controlling the tail of the constraint distribution).

The candidate solution \hat{x}_i^* in iteration i is defined as the maximizer of the mean estimate on \mathcal{S}_i^τ ,

$$\hat{x}_i^* = \text{CANDIDATE}(\mathcal{S}_i^\tau, \hat{m}_i) \stackrel{\text{def}}{=} \arg \max_{x \in \mathcal{S}_i^\tau} \text{MEAN}_f(\hat{m}_i, x).$$

Evaluating *just* the incumbent solution however does not induce enough exploration, and the algorithm would get stuck in local optima. Instead, the next sequential measurement is chosen to maximize the upper confidence bound score function, which is a well-behaved surrogate to reduce the uncertainty of the true maximizer:

$$x_t = \arg \max_{x \in \mathcal{S}_t^\tau} \text{UCB}_f(\hat{m}_t, x, \delta). \quad (1)$$

This choice leads to evaluation of points where the model predicts either a high reward or large uncertainty (or both). Upper confidence bound algorithms are widely studied in the literature [43] and convergence guarantees for these algorithms are known for the unconstrained case [41,44]. The constrained case requires additional care to ensure that model obtains sufficient data to expand the safe set [32]. The exact acquisition procedure that we use is specified in Algorithm 2. Since finding a maximizer of the acquisition score (1) is challenging in high-dimensional domains, we rely on the LineBO approach by Kirschner *et al.* [1]. Instead of searching the full domain, LineBO restricts the search space to an adaptively chosen sequence of one-dimensional subspaces. Let $\mathcal{A}(x, w) = \{x + aw: a \in \mathbb{R}\}$ be an affine subspace for offset $x \in \mathbb{R}^d$ and direction $w \in \mathbb{R}^d$. To determine direction and offset we first collect data points in a small ball $\mathcal{B}(\hat{x}_{i-1}^*, \eta)$ around the previous incumbent solution \hat{x}_{i-1}^* to compute a new candidate \hat{x}_i^* . The offset is then set to the new candidate \hat{x}_i^* and the search direction is chosen as $w = \hat{x}_i^* - \hat{x}_{i-1}^*$.

For step-size control during the line-search phase, the algorithm constrains the next evaluation point x at any time to $\mathcal{B}(\hat{x}_i^*, \eta)$. The complete approach is summarized in Algorithm 1. An illustration is given in Fig. 1 and the line search phase is shown in Fig. 2.

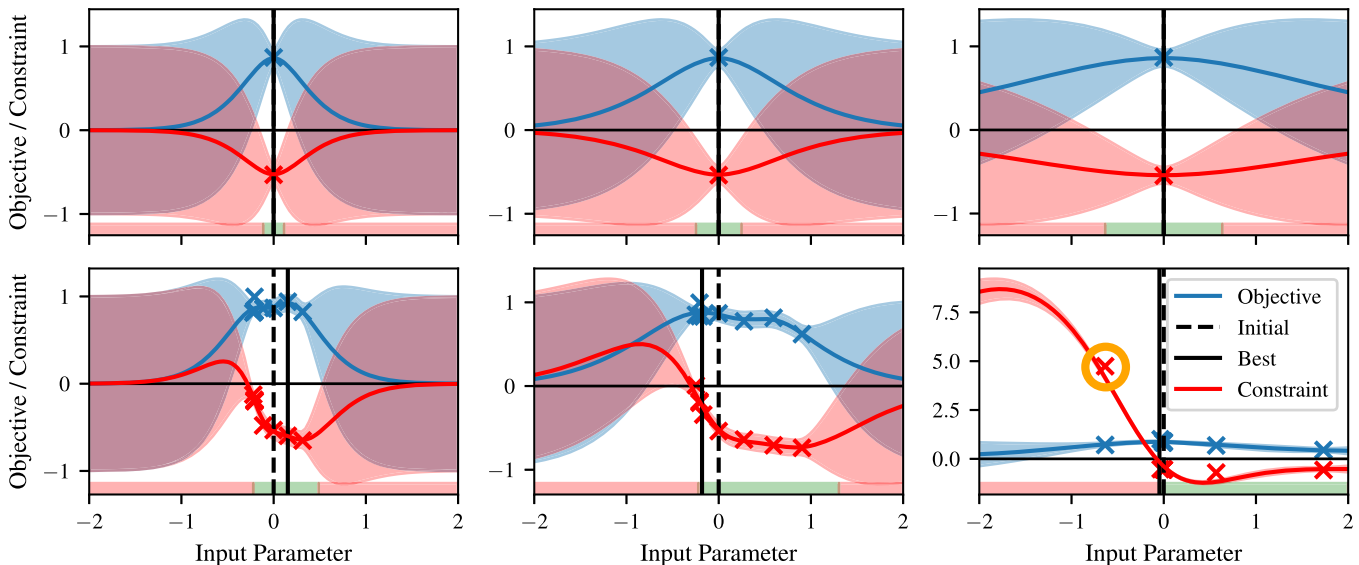


FIG. 2. The plots show one-dimensional slices of a synthetic objective (blue, maximized) and constraints (red). The left, middle and right columns correspond to different GP models with length scales 0.1, 0.2 and 0.5, respectively, on the same objective. We produce such plots live during optimization to evaluate model fit and confidence bands relative to the measurements (crosses). The vertical dashed line indicates the best parameter prior to optimizing the one-dimensional subspace, and the vertical solid line indicates the updated best parameter. Green and red intervals on the x axis correspond to predicted safe and unsafe parameters. Note that the safe region is expanded as more data is collected (cf. top to bottom plot). When choosing hyperparameters such as the kernel length scale, system experts can visually inspect the predicted safe set, and make adjustments accordingly. A smaller length scale leads to a more conservative method that expands the safe set in smaller steps. Note that in the right column, the length scale is chosen too large. At the first evaluation the safe set is predicted incorrectly (see upper right plot), which consequently leads to an unsafe query point (orange circle, bottom plot).

B. Variants

In our experiments, we evaluate the following variants of the proposed approach.

A-LineBO-loc.—This is the main (ascent & localized) variant shown in Algorithm 1. We use $\eta = 0.1$ relative to a domain that is normalized to $[0, 1]^d$.

A-LineBO.—For this variant, we do not enforce the step size constraint in line 12 of Algorithm 1, which corresponds to the variant proposed in [1,22].

C-LineBO-loc.—This variant uses coordinate aligned search directions, also known as *coordinate descent*. Specifically, we replace lines 2–8 in Algorithm 1 by setting $w_i = e_{i \bmod d}$, where e_1, \dots, e_d are the basis vectors.

C-LineBO.—Similar to A-LineBO, this variant does not enforce the step size constraint in line 12. Note, this variant (without the step-size constraint) was proposed and evaluated by Kirschner *et al.* [1,22].

C. Choice of hyperparameters

Algorithm 1 relies on several hyperparameters that are configured by the user. We first explain our choices in the following. Then we briefly discuss more general methods for adjusting the hyperparameter parameters.

For the number of evaluations per round, we set $j_{\text{ball}} = 2d$ and $j_{\text{line}} = 10$. From our experience, this allows to estimate a reasonable search direction and

make sufficient progress on each line (and the method can revisit the same search direction in the next iteration). When the signal is very noisy, a larger number of evaluations per iteration can be appropriate. On systems with a large number of hyperparameters ($d > 50$) but low *effective dimensionality*, it can be effective to choose search direction uniformly at random [1]. The step-size η and margin τ are provided by the user. The exact choice is usually based on system considerations. We use a maximum step size of 0.1 (i.e., 10% of the allowed input range) and $\tau = 0.1$ (i.e., 10% below the critical value).

Parameters of the GP model encode regularity assumptions on the objective and constraints. Choosing suitable model parameters is essential for the effectiveness of Bayesian optimization. In a first step, we normalize input ranges to obtain a normalized domain $\mathcal{X} = [0, 1]^d$. Further, we normalize the observed objective values to $[0, 1]$, and the range of feasible constraint values to $[-1, 0]$. The latter is important to ensure that the prior upper-confidence bound predictions of the constraints are positive. This corresponds to assuming that an input parameter is unsafe in the absence of data.

We then use a Matern52 kernel with length scale 0.2 and unit prior variance, which we found effective and robust choices across all our experiments. For the observation

Algorithm 1: A-LineBO

Input: domain \mathcal{X} , initial evaluation point \hat{x}_0 , initial data \mathcal{D}_0 , confidence $\delta \in [0, 1]$, step-size $\eta > 0$, safety margin $\tau \geq 0$, $j_{\text{ball}}, j_{\text{line}} \in \mathbb{N}$ inner loop evaluations

Output: Estimated maximizer

- 1 $t \leftarrow 1, \quad i \leftarrow 0, \quad \hat{x}_0^* \leftarrow x_0$ // initialization
- 2 **while** *not stopped by user* **do**
- // Determine best search direction w_i
- 3 **for** $j = 1, \dots, j_{\text{ball}}$ **do**
- 4 $\hat{m}_t \leftarrow \text{REGRESSION}(\mathcal{D}_{t-1})$
- 5 $x_t \leftarrow \text{SAFE-ACQUISITION}[\mathcal{B}(\hat{x}_{t-1}^*, \eta) \cap \mathcal{X}, \hat{m}_t, \tau]$
- 6 $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup [x_t, \text{MACHINE-EVAL}(x_t)]$
- 7 $t \leftarrow t + 1$ // evaluation counter
- // Update candidate and search direction
- 8 $\hat{x}_i^* \leftarrow \text{CANDIDATE}[\mathcal{B}(\hat{x}_{t-1}^*, \eta) \cap \mathcal{X}, \hat{m}_t, \tau]$
- 9 $w_i \leftarrow \hat{x}_i^* - \hat{x}_{t-1}^*$
- // Search line in direction w_i & offset \hat{x}_i^*
- 10 **for** $j = 1, \dots, j_{\text{line}}$ **do**
- 11 $\hat{m}_t \leftarrow \text{REGRESSION}(\mathcal{D}_{t-1})$
- 12 $\hat{x}_i^* \leftarrow \text{CANDIDATE}[\mathcal{A}(\hat{x}_i^*, w_i) \cap \mathcal{B}(\hat{x}_i^*, \eta) \cap \mathcal{X}, \hat{m}_t, \tau]$
- 13 $x_t \leftarrow \text{SAFE-ACQUISITION}[\mathcal{A}(\hat{x}_i^*, w_i) \cap \mathcal{B}(\hat{x}_i^*, \eta) \cap \mathcal{X}, \hat{m}_t, \tau]$
- 14 $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup [x_t, \text{MACHINE-EVAL}(x_t)]$
- // Optional user feedback:
- 15 $\text{LINE-PLOT}(\mathcal{D}_t, \hat{m}_t, \hat{x}_t, w_i)$
- 16 $t \leftarrow t + 1$ // evaluation counter
- 17 $i \leftarrow i + 1$ // next iteration
- 18 **return** \hat{x}_{i-1}^*

noise likelihood, we use a Gaussian distribution with an empirical variance estimate. We choose to keep all hyperparameters fixed during optimization, but adaptive choices are possible in principle.

The above choice of kernel and length scale is appropriate when a small change in the input parameters (e.g.,

< 10% relative to the input range) leads to a moderate change in the output, e.g., does not suddenly increase the constraint level to a critical value. This requires to choose suitable allowed ranges of tuning parameters, which we determined together with system experts.

To adjust these parameters on a new system, the practitioner can inspect the model predictions after the initial (safe) point has been evaluated. Most importantly, the predicted safe set for each parameter can be plotted and evaluated by a system expert. Based on the visual feedback, the length scale or input ranges of each parameter can be adjusted to meet the operators expectation on the safety of the method. The method acts more conservatively for larger safety-margin τ , smaller step-size η , and smaller length scale parameter used for the kernel estimation. The visual inspection procedure is illustrated in Fig. 2.

We also produce the line plots on-line during the line-search phase. We found this to be an extremely valuable tool to further calibrate hyperparameters and obtain feedback on the optimization progress. Specifically, too conservative parameter choices (e.g., a very small length scale) lead to slow expansion of the safe set and no/little extrapolation of the target values beyond the data points. On the other hand, if the regression estimate appears too smooth relative to the observed data points, or the algorithm evaluates points close or above the constraint limit, more conservative hyperparameters are advisable.

Algorithm 2: SAFE-ACQUISITION

Input: search domain $\tilde{\mathcal{X}}$, model \hat{m} , confidence δ , safety margin τ

Output: acquisition point x

// Define safe-set:

- 1 $\mathcal{S}^\tau \leftarrow \{x \in \tilde{\mathcal{X}}: \max_{i \in [l]} \text{UCB}_{h_i, \delta}(\hat{m}, x) \leq -\tau\}$
- // UCB and Safe UCB parameter:
- 2 $x^{\text{UCB}} \leftarrow \arg \max_{x \in \tilde{\mathcal{X}}} \text{UCB}_f(\hat{m}, x, \delta)$
- 3 $x^{\text{UCB-S}} \leftarrow \arg \max_{x \in \mathcal{S}^\tau} \text{UCB}_f(\hat{m}, x, \delta)$
- 4 **if** $x^{\text{UCB}} = x^{\text{UCB-S}}$ **then**
- 5 **return** $x^{\text{UCB-S}}$
- // Compute expanding point towards x^{UCB} :
- 6 $x^{\text{EXP}} \leftarrow \arg \min_{x \in \mathcal{S}} \|x - x^{\text{UCB}}\|$
- 7 **if** $\max_{i \in [l]} \text{CW}_{g_i}(\hat{m}, x^{\text{EXP}}, \delta) > \text{CW}_f(\hat{m}, x^{\text{UCB-S}}, \delta)$ **then**
- 8 **return** x^{EXP}
- 9 **else**
- 10 **return** $x^{\text{UCB-S}}$

In applications where prior data or a physics simulation is available, an alternative approach is to estimate hyperparameters in advance, e.g., using a maximum likelihood or marginalized likelihood estimation [42]. For a successful approach to extract kernel hyperparameters from a physics model, see the work by Hanuka *et al.* [24]. Some previous work has also explored adaptive model selection, e.g., [45], although this is still largely an active research area today.

Lastly, a word of caution is unavoidable. Safety and effectiveness of the methods is subject to the modeling assumptions, and therefore, choice of hyperparameters. In the absence of any assumptions on the system, a principled choice is limited. The challenge is amplified for safe optimization, because no method can guarantee safety if the system behaves in a way not anticipated by the algorithm.

D. Remarks on computation

GP regression requires to compute the inverse of the kernel matrix, which can be done in $\mathcal{O}(t^3)$ basic operations per round. With incremental updates, the complexity is reduced to $\mathcal{O}(t^2)$. In practice, this effectively limits the number of data points that can be handled without significantly increasing computation time to ~ 1000 – 2000 . See Appendix A for further comments.

The SAFE-ACQUISITION subroutine requires solving nonconvex optimization problems over continuous domains. When the domain is one dimensional as in the line search, a fine uniform (equidistant) discretization of the domain easily produces a solution below the system accuracy/noise level, such that the approximation error translates to a negligible difference on the objective value (we used 300 points per line). Maximizing the acquisition score within in the ball $\mathcal{B}(\hat{x}^*, \eta)$ is computationally expensive in general. Nevertheless, since the domain is relatively small compared to the full domain, we found a random search to be effective in our experiments. We compute the UCB score at 500 points sampled uniformly from $\mathcal{B}(\hat{x}^*, \eta)$ and choose the maximizer among these. An alternative is to maximize the UCB score using any standard optimization algorithm such as gradient descent with random restarts.

Lastly, we note that in time-critical applications with a large number of constraints, a significant speedup can be obtained by using a joined covariance matrix for all constraints. Further, the most expensive step in computing the estimates is the inversion of the covariance matrix, which can be precomputed while waiting for the machine to return the next evaluation.

E. Limitations

Enforcing a step-size sometimes limits the algorithm's ability to escape local optima. However, theoretical analysis suggests that such an approach at least finds a local optimum [46,47]. In our experiments we did not see any performance degradation from enforcing a step size (see Sec. III). This indicates that, perhaps surprisingly, high-

quality local solutions exist in the optimization surfaces that we encountered. On systems where local optima prevent the algorithm from progressing, one can use multiple starting points and drop the step-size constraint during the line-search phase. Another option is to replace the line search by a search over higher-dimensional subspaces.

Another challenge is model misspecification. If a parameter is incorrectly predicted as safe, the parameter can become infeasible under the model's predictions *after* the evaluation (cf. Fig. 2, right column). With a small step size, this can lead to a situation where the predicted safe set is empty. A simple resolution is to back-track and restart the method from an earlier safe point. In such cases, it is further advisable to choose a smaller length scale, to prevent evaluation of infeasible points in the first place.

III. EXPERIMENTAL EVALUATION

We first present empirical results on HIPA, and then an additional experiment on SwissFEL below.

A. HIPA

The different versions of the BO algorithm with constraints have been tested in several dedicated experiments at HIPA. The target of these experiments has been to reduce the overall beam losses around the machine by minimizing a target signal defined as a weighted sum of about 60 beam loss monitors (or loss related monitors) spread across the machine. This weighted sum reflects that the beam loss monitors are not distributed equidistantly along the machine and that losses at lower energies cause less activation.

For each beam-related signal that could trigger a beam interruption (*interlock*), a safety constraint function is defined as the difference between the signal and its beam interruption limit. If available, the *warning* limit of each signal is used and not the actual limit that would produce a beam interruption when crossed, which allows a safety margin in case the safety constraint is accidentally violated when operating near the limit. For optimization of HIPA, we used 224 of such signals, mostly current and temperature measurements in collimators and beam loss monitors. We normalized each of the safety signals $g_i^{\text{loss}}(x)$ to define constraint functions,

$$g_i(x) = \frac{g_i^{\text{loss}}(x) - g_{i,\text{limit}}}{g_{i,\text{limit}}},$$

where $g_{i,\text{limit}}$ is the limit of constraint i . Therefore the constraint reads $g_i(x) \leq 0$, and the safe outcome range is $[-1, 0]$ [assuming that $g_i^{\text{loss}}(x) \geq 0$]. For faster computation, it is possible to take the maximum over multiple constraints. However, this can lead to less smooth functions that are harder to learn by the GP model.

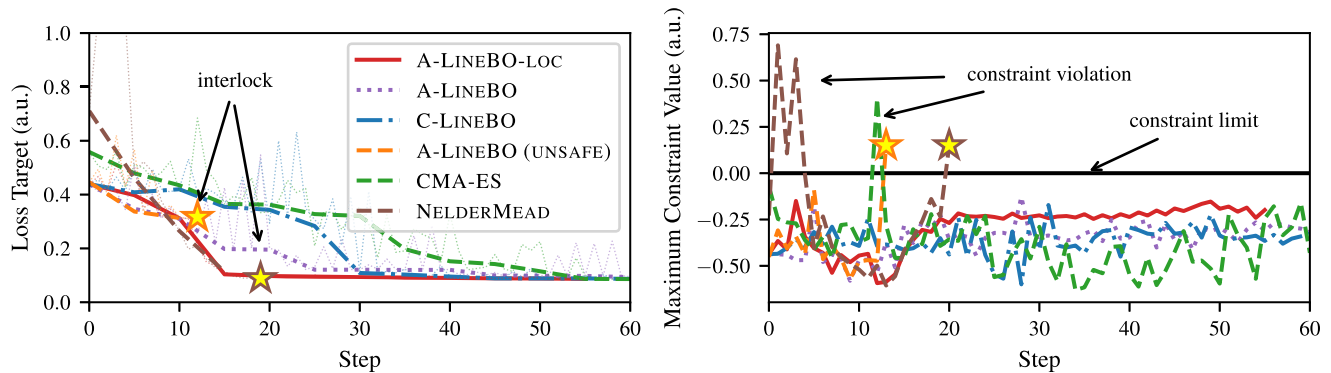


FIG. 3. Step-by-step evolution of the target function for the different optimization algorithms. We performed optimization at low beam intensity, using five quadrupole strengths on a manually detuned initial point. The left plot shows the target values. The faint lines represent the measurement of the target function at each step. The solid lines depict the measurement of the target function at the best set of parameters predicted by the optimization models (updated only every five steps). The right plot shows the maximum constraint value at each step, where a value above zero corresponds to a constraint violation. Note that NELDERMead and A-LineBO (unsafe) caused interlocks at step 12 and 19 respectively. Note that the exact values of the corresponding constraint violations are unknown, because the machine protection system stops beam operation before the measurement is registered. CMA-ES and NELDERMead caused constraint violations that did not trigger interlocks at steps 12 and 2-4 respectively.

At HIPA, a beam current regulation and several beam orbit feedbacks are in continuous operation to ensure stable operation without drifts. Every time a new evaluation point is set, the algorithms wait for these feedbacks to stabilize before acquiring data. Larger steps typically result in more time required until the machine is stable and the measurement is taken.

In our experimental validations, the variables used to optimize the target function are the field strengths of quadrupoles in the 870 keV transport beam line after the Cockcroft-Walton preaccelerator towards the first of the two cyclotrons, the injector II, with an allowed empirical range of ± 2 A. This range is equivalent to an integrated field gradient of about ± 0.01 T, and is a typical range operators use for tuning. In the center of injector II the beam is collimated from 10 mA to about 2 mA with several collimators [48]. By adjusting the quadrupole strengths upstream, the beam can be shaped and matching in the cyclotron can be improved [49], which will avoid beam tails and reduce beam losses throughout the accelerator downstream.

In order to protect the machine, the first experiments were performed at low intensity ($100 \mu\text{A}$), which also allows a larger margin in the constraint functions limits as the losses are in general lower at low intensity. To give some headroom to the automatic algorithms and to simulate a suboptimal machine, the selected quadrupole settings are manually moved away from the default settings. These detuned settings are recorded to have a consistent initial point. In later sessions the algorithms were tested without detuning at full production current (about $1930 \mu\text{A}$).

The left plot in Fig. 3 shows the target evolution of the different algorithms at low beam intensity. The faint lines show the evaluation of the target function at every new

parameter configuration (step). The solid line shows the measurement of the target function for the best set of parameters predicted by the algorithms after every five steps (evaluating the candidate solution more frequently slows down the overall optimization). The right plot shows the constraint closest to the limit at each evaluation point (that is, the maximum over the normalized constraint levels). Note that NELDERMead and A-LineBO (unsafe) caused beam interruptions due to constraint violations. These constraint violations are *not* visible on the constraint plot because the machine projection system stops beam operation before the measurement is registered. Further, CMA-ES and NELDERMead show constraint violations that did not lead to beam interruptions. This can happen in our setup when the warning level of a loss monitor is reached, but not the beam interruption level. All safe methods and CMA-ES successfully optimize the loss to a final level of about 0.1, where A-LineBO-loc converges the fastest.

Our second experiment at full production intensity is shown in Fig. 4, where we optimize 16 quadrupoles (all the quadrupoles in the 870 keV beam line). The target reduction is less steep because, in order to avoid putting the machine in a dangerous state, in this case the parameters were not manually detuned and the initial point was already fairly close to the optimum. Notice also the difference on the magnitude of the target function due to the higher intensity and therefore higher losses around the machine. The C-LineBO-loc and CMA-ES algorithms stopped early as they triggered beam interruptions in their last parameter settings.

This experiment was performed with an earlier configuration of our method without safety margin ($\tau = 0$). Moreover, we used a single constraint defined as the

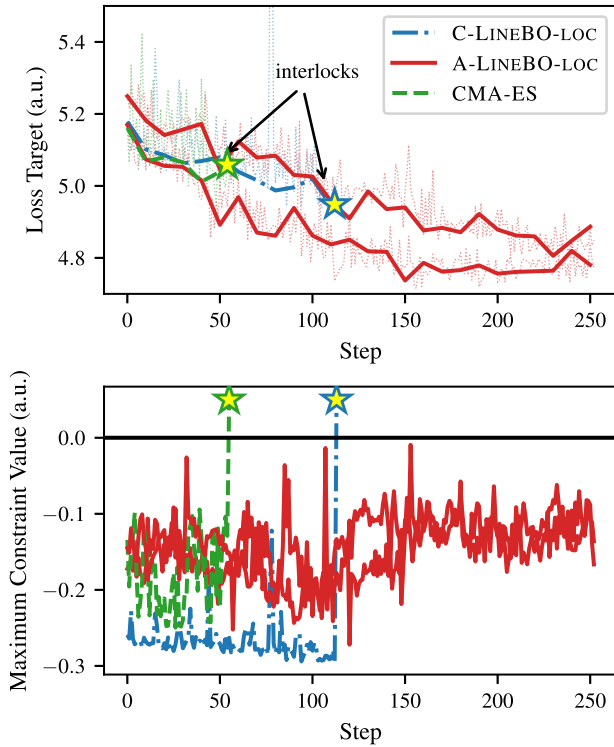


FIG. 4. (HIPA) Experiment at high beam intensity with 16 tuning parameters. Both CMA-ES and C-LineBO-loc caused interlocks, and the run was stopped after 50 and 110 steps respectively. Note that the exact constraint value of the constraint violation is not known as the machine protection system prevents the measurement. A-LineBO-loc successfully tunes the machine at high intensity (repeated twice). This experiment used an earlier configuration of the LineBO methods, where we used a single constraint defined as the maximum over the 224 (normalized) loss signals and no safety margin ($\tau = 0$). We think that this configuration led to misspecified confidence intervals, and therefore an interlock of C-LineBO-loc. In such an event we recommend to use more conservative hyperparameters.

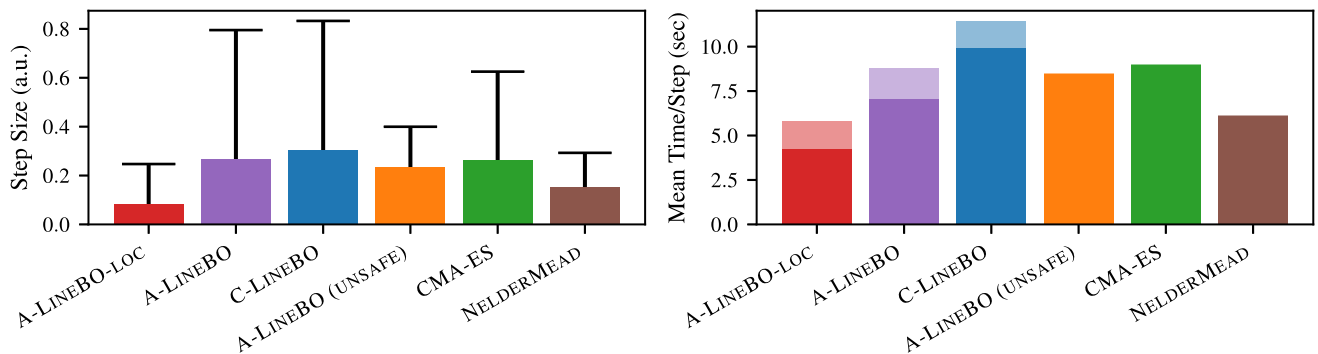


FIG. 5. (HIPA) Observed average step sizes (T-bars: maximum) for different algorithms measured in Euclidean norm distance between parameter vectors with normalized input range (left). The right plot shows time used per step, where the solid part of each bar corresponds to machine evaluation time, and the faint part to compute time (on a single core Intel[®] Xeon[®] Gold 6140 2.3 GHz with 4 GB RAM), i.e., A-LineBO-loc: 1.54 s, A-LineBO: 1.72 s, C-LineBO: 1.51 s, A-LineBO (unsafe): 0.01 s, CMA-ES: 0.0004 s, NELDERMEAD: 0.0004 s. The significant increase in computation time for the constraint method is due to the large number of constraints (224). This can be improved by using a joined covariance matrix for all constraints, see the remarks in II D. A-LineBO and C-LineBO are not step-size constrained, which leads to larger machine evaluation times. C-LineBO-loc has an explicit step-size limit which leads to shorter machine evaluation times. Note that A-LineBO (unsafe) and NELDERMEAD caused interlocks after few steps (see Fig. 3).

maximum over the 224 loss monitors. Consequently, an interlock was caused by C-LineBO-loc, either because of model misspecification (e.g., the maximum over the constraints was non-smooth) and consequently invalid confidence intervals, or noise of constraint monitor that was very close to its limit. In such cases we recommend more conservative hyperparameters (bigger margin τ , smaller length scales of the GP models), as was used in the experiment shown in Fig. 3. A-LineBO-loc completed the optimization successfully twice, without triggering any beam interruption.

Figure 5 shows an analysis of the step sizes and their impact on the speed of the optimization of the different algorithms. The plots show data from the low intensity experiment. The C-LineBO algorithm has no limitations on the step sizes and therefore is allowed to navigate the whole safe region in single steps. Compared with the A-LineBO-loc algorithm with limits on its step size it can be seen on the left plot how the unrestricted algorithms take much larger steps. The impact of these large steps can be seen on the right column plot, as due to the large steps taken by the unrestricted algorithm the optimization took about $1.5\times$ as long to complete because it takes the machine feedbacks and regulation systems much longer to converge for these large steps. We remark that the high computation time for the safe BO methods is due to the use of 224 constraints. We expect that significant speedups are possible when using a joint kernel matrix for all constraints. Further, the most expensive step of inverting the covariance matrix can be precomputed while waiting for the machine to return the next measurement.

A closer analysis (see Appendix B) shows that the step sizes taken by the CMA-ES diverge as the algorithm approaches the optimization solution, as the target function becomes close to invariant to the change of the parameters in this region and tries to spread the sampling points. This is

an undesirable behavior with respect to potential constraint violations or unstable machine states caused by large steps.

In conclusion, the experiments at HIPA demonstrate that the BO optimization algorithms successfully reduce the losses around the machine in a safe and efficient way, both at low intensity and full intensity operational setting. Particularly, the A-LineBO-loc algorithm outperforms or matches CMA-ES and the loss level achieved by human operators, while staying in general below the limits that would trigger beam interruptions. The constrained step sizes of this method also allow for faster and more stable tuning.

B. SwissFEL

Previous work [22] established that LineBO outperforms NELDERMead [4] for tuning the beam intensity on SwissFEL. Here, we provide an additional experiment on SwissFEL, including CMA-ES, A-LineBO, and the step-

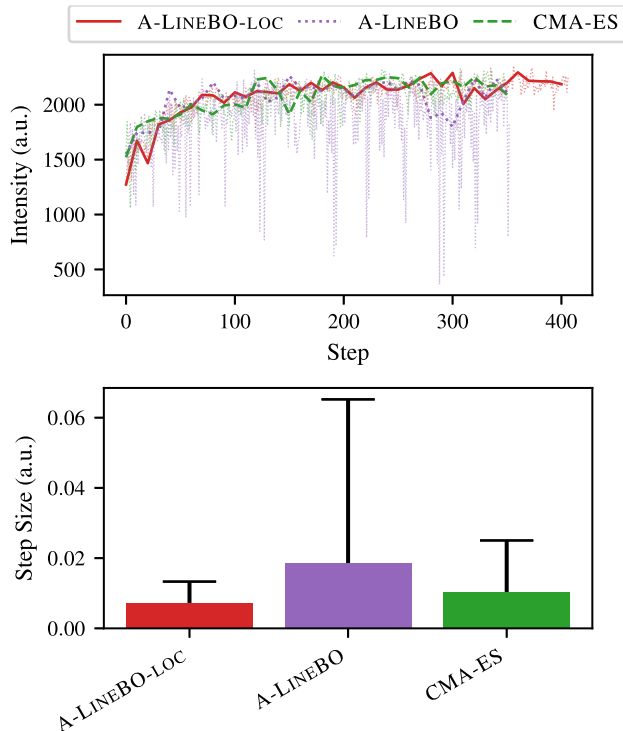


FIG. 6. (SwissFEL) Step-by-step evolution of the SwissFEL single pulse intensity (top) and average step size (T-bars: maximum) during optimization of 24 beam position monitor target values. The results demonstrate that the same methods can be used on different accelerators. We emphasize that, similar to the experiments on HIPA, the step-size constraints did not affect optimization performance while resulting in significantly smaller variation in the input space, and therefore a more stable machine. This can be seen in the top plot, where the faint lines depict the objective value observed at the evaluation points during optimization.

size constrained variant A-LineBO-loc. As mentioned in Sec. IC the tuning signal is a shot-by-shot signal of the gas detector that gives a signal proportional to the amount of photons in a single pulse. As optimization variables, we use ten (horizontal and vertical) beam position monitor target values for the trajectory feedback in the undulator section. We used as constraints 16 individual loss monitors that were combined into a single constraint. During optimization with the specified parameters, we did not observe any losses.

The top plot of Fig. 6 shows the target evolution of the different algorithms. The faint lines show the evaluation of the target function at every new parameter configuration (step). The solid line shows again the measurement of target function for the best set of parameters predicted by the algorithms every ten steps. While all three tested algorithms achieve a similar final target value, it can be seen that the target value often drops for A-LineBO. This is due to large steps in the parameter space, as shown in the bottom plot of Fig. 6, where the average (normalized) step size is plotted. It can also be seen that the A-LineBO-loc performs slightly better than CMA-ES in this respect. Having a low variability and a high average of the target value is especially important during *parasitic* tuning, i.e., tuning during user’s experiments. For this reason the step size constrained A-LineBO-loc is the preferred algorithm.

IV. CONCLUSIONS

While tuning particle accelerators remains a challenging task, we demonstrated the feasibility of safe Bayesian optimization for automated parameter tuning. Safe Bayesian optimization does not rely on any specific machine model, and accounts for an arbitrary number of safety constraints. By both optimizing and exploring the target and constraint functions, the algorithm continuously improves its estimate and uncertainty of these functions, and in this way uses all available data to evaluate promising but safe parameters. Multiple variants of safe Bayesian optimization have been described in detail, including a newly developed variant with step-size control. This is an important feature for tuning particle accelerators, since large parameter steps can lead to an unstable machine state as fast feedbacks might not be able to follow. These variants have been applied and compared with nonsafe search methods CMA-ES, NELDERMead and Bayesian optimization without constraints. We presented experimental data for two accelerators, HIPA and SwissFEL, showing efficacy and reproducibility in several optimization runs. Overall, we find that the different variants of Bayesian optimization and CMA-ES are both effective tools for beam optimization. However, safe Bayesian optimization is shown to cause constraint violations much less likely, and consequently, avoid beam interruptions. In addition, the step-size controlled variants allow for a faster and more stable tuning, which is less disruptive for the users.

ACKNOWLEDGMENTS

The authors would like to thank Rasmus Ischebeck for establishing the first contact, and his contributions to the earlier publications of the collaboration. Manuel Nonnenmacher and Andreas Adelman contributed to early development and testing of the LineBO algorithm as part of Manuel's Master thesis. Nicolas Lehmann and Marco Boll contributed to the GUI development. We acknowledge the support of the SwissFEL and HIPA operation teams during the tuning experiments. This research was supported by SNSF Grants No. 200020_159557 and No. 407540_167212 through the NRP 75 Big Data program. Johannes Kirschner acknowledges funding through the SNSF Early Postdoc.Mobility fellowship P2EZP2_199781. The project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program Grant Agreement No. 815943. This project was also supported by the Swiss Data Science Center (SDSC) within the Particle Accelerator and Machine Learning (PACMAN) project.

APPENDIX A: GAUSSIAN PROCESS REGRESSION

1. Kernel regression

With t evaluation-feedback pairs $\mathcal{D}_t = \{(x_1, y_1), \dots, (x_t, y_t)\}$ available from previous steps or data collected prior to optimization, we can compute an estimate \hat{f}_t to approximate the unknown true function, and the estimate is used to inform the next evaluation point. Most commonly, Bayesian optimization uses kernel least-squares regression to estimate the target and constraint functions:

$$\hat{f}_t = \arg \min_{f \in \mathcal{H}} \sum_{s=1}^t [f(x_s) - y_s]^2 + \|f\|_{\mathcal{H}_k}^2. \quad (\text{A1})$$

The minimization is over functions of a reproducing kernel Hilbert space \mathcal{H}_k (see, e.g., [42]), which is defined by a kernel function $k: \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, and $\|\cdot\|_{\mathcal{H}_k}$ denotes the associated Hilbert norm. The estimate \hat{f}_t can be conveniently computed in analytic closed form as follows:

$$\hat{f}_t(x) = k_t(x)^\top (K_t + \mathbf{1}_t)^{-1} y, \quad (\text{A2})$$

where $\mathbf{1}_t \in \mathbb{R}^{t \times t}$ is the identity matrix, K_t is the $t \times t$ matrix containing $(K_t)_{ij} = k(x_i, x_j)$ formed by pointwise evaluation of the kernel function, y is a vector of responses and lastly $k_t(x)_i = k(x, x_i)$ interpolates the measurement x to other data points. A direct implementation requires the inversion of the kernel matrix K_t with computational complexity $\mathcal{O}(t^3)$. Iterative updates using the Sherman-Morrison identity reduce the complexity to $\mathcal{O}(t^2)$ per round. If the computational burden is too large, methods relying on approximation tools for kernel spaces have been

developed with near-linear dependence for low dimensional problems [50,51].

Frequentist confidence bands $\hat{f}_t(x) \pm \beta_{t,\delta}^{1/2} \sigma_t(x)$ have been derived in the literature, which contain the true function with high probability in the realizable case [41,52]. The quantity σ_t corresponds to the estimation uncertainty and can likewise be derived in closed form,

$$\sigma_t(x) = \sqrt{k(x, x) - k_t(x)^\top K_t^{-1} k_t(x)}, \quad (\text{A3})$$

where K_t and $k_t(x)$ as above. The parameter $\beta_{t,\delta}^{1/2}$ is a confidence parameter that determines the coverage and can be chosen according to frequentist theoretical results or tuned as a hyperparameter (we used $\beta_{t,\delta} = 1$ in our experiments).

Using the notation from Sec. II, this leads to the following model estimates:

$$\text{MEAN}(\hat{f}_t, x) = \hat{f}_t(x)$$

$$\text{UCB}(\hat{f}_t, x, \delta) = \hat{f}_t(x) + \beta_{t,\delta}^{1/2} \sigma_t(x)$$

$$\text{LCB}(\hat{f}_t, x, \delta) = \hat{f}_t(x) - \beta_{t,\delta}^{1/2} \sigma_t(x)$$

$$\text{CW}(\hat{f}_t, x, \delta) = 2\beta_{t,\delta}^{1/2} \sigma_t(x).$$

The Bayesian viewpoint offers an alternative interpretation where the kernel estimator corresponds to the posterior mean of a Gaussian process regressor [53] which was exploited in deriving tighter predictive bands in [41].

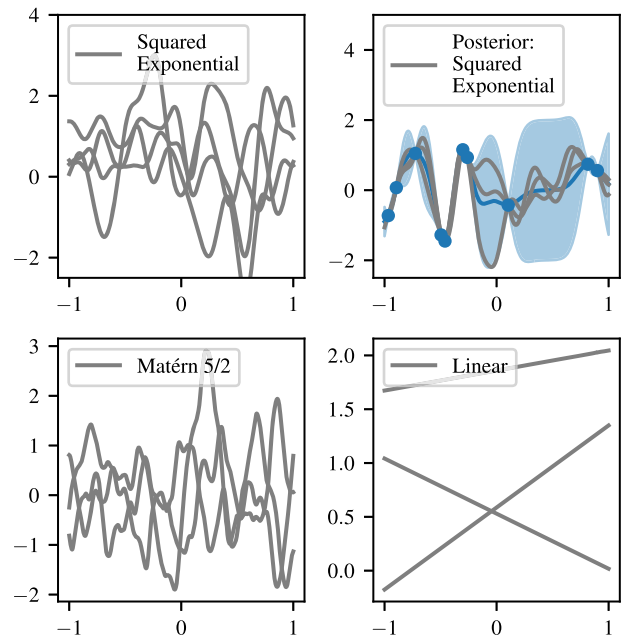


FIG. 7. The plots show functions sampled from a Gaussian process defined for different kernels. In addition, the top-right plot shows the mean estimate \hat{f} along with the 95% confidence bands in gray.

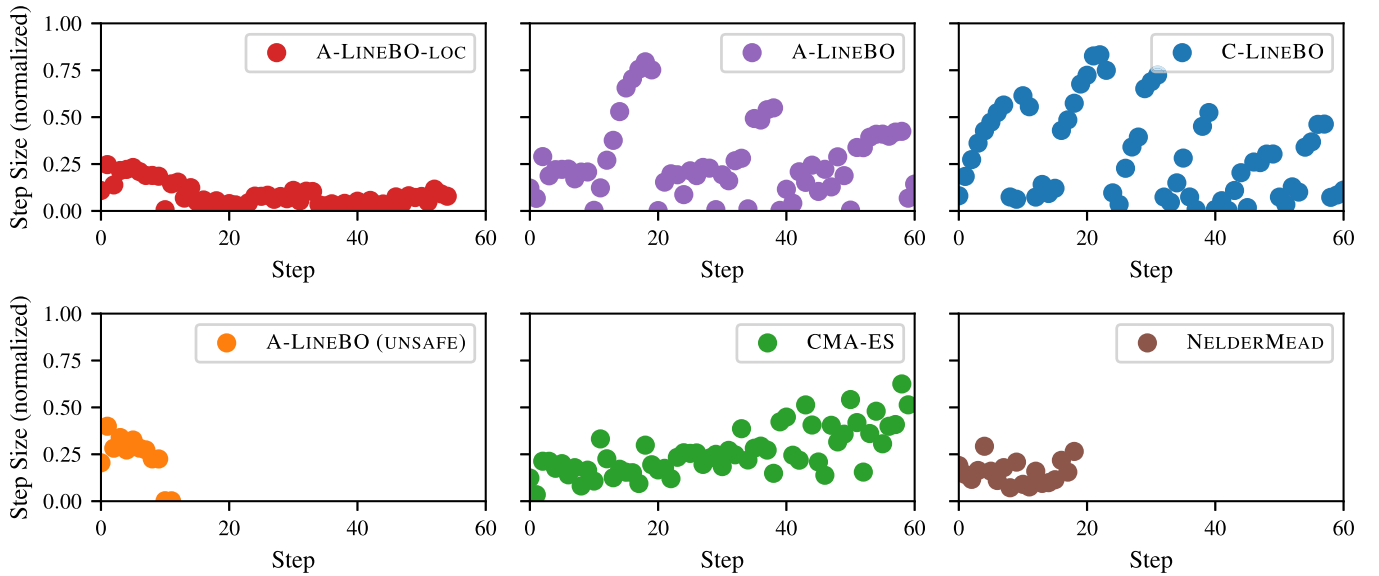


FIG. 8. Each plot shows the step size (measured in Euclidean norm) for the different methods. Note that without step-size constraints, LineBO takes large steps throughout (top middle/right). We also observed that CMA-ES was slowly increasing step sizes, likely due to invariant subspaces of the objective. Unsafe LineBO and NELDERMEAD stopped early because of interlocks, therefore interpretation of the data is limited.

2. Kernel choice

The choice of the kernel is crucial for obtaining a data efficient model. Depending on the kernel choice, we can represent a different class of functions. Failure to identify the right kernel for the application leads to misspecified confidence sets and hence potential failure of the method. Different choices for the kernel are demonstrated in Fig. 7 for squared exponential $k(x, y) = \exp(-\frac{(x-y)^2}{\gamma^2})$, Matérn kernel [42] and linear kernel $k(x, y) = x^\top y$. In the case of squared exponential kernel (as well as Matérn), γ is often referred to as length scale or bandwidth, and determines the smoothness of the estimated functions.

3. Bayesian optimization

Classical Bayesian optimization [2] usually revolves around a simple procedure depicted in Fig. 2, where a utility function—in this case the UCB—is maximized to determine the next evaluation point. Other acquisition functions are possible, see e.g., Shahriari *et al.* [21].

APPENDIX B: ADDITIONAL EXPERIMENTAL DATA

Figure 8 shows the step sizes for each algorithm, measured as the Euclidean distance between two consecutive input parameters on a normalized scale. Note that the step-size constrained variant takes significantly smaller steps throughout. We also found that CMA-ES was increasing its internal step size during this run. This is due to

invariant subspaces on the objective, which leads to an increased sampling (co)variance along these directions.

-
- [1] J. Kirschner, M. Mutný, N. Hiller, R. Ischebeck, and A. Krause, Adaptive and safe Bayesian optimization in high dimensions via one-dimensional subspaces, in *Proceedings of the International Conference for Machine Learning (ICML)*, edited by K. Chaudhuri and R. Salakhutdinov, Proceedings of Machine Learning Research Vol. 97 (2019), pp. 3429–3438, <https://proceedings.mlr.press/v97/kirschner19a.html>.
 - [2] J. Mockus, *Bayesian Approach to Global Optimization* (Springer, Dordrecht, 1989), <https://link.springer.com/book/10.1007/978-94-009-0909-0>.
 - [3] P. I. Frazier, A tutorial on Bayesian optimization, [arXiv: 1807.02811](https://arxiv.org/abs/1807.02811).
 - [4] J. A. Nelder and R. Mead, A simplex method for function minimization, *Comput. J.* **7**, 308 (1965).
 - [5] N. Hansen, S. D. Müller, and P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES), *Evolutionary computation* **11**, 1 (2003).
 - [6] J. Kennedy and R. Eberhart, Particle swarm optimization, in *Proceedings of the ICNN'95-International Conference on Neural Networks* (IEEE, New York, 1995), Vol. 4, pp. 1942–1948.
 - [7] I. Agapov, G. Geloni, S. Tomin, and I. Zagorodnov, OCELOT: A software framework for synchrotron light source and FEL studies, *Nucl. Instrum. Methods Phys. Res., Sect. A* **768**, 151 (2014).

- [8] S. Tomin, G. Geloni, I. Zagorodnov, A. Egger, W. Colocho, A. Valentinov, Y. Fomin, I. Agapov, T. Cope, D. Ratner *et al.*, Progress in automatic software-based optimization of accelerator performance, *Proceedings of IPAC'16, Busan, Korea* (2016), p. 3064, <https://inspirehep.net/literature/1470409>.
- [9] S. Tomin, M. Scholz, and I. Zagorodnov, On-line optimization of European XFEL with OCELOT, in *International Conference on Accelerator and Large Experimental Control Systems (ICALEPCS'17), Barcelona, Spain* (2017), <https://inspirehep.net/literature/1656298>.
- [10] X. Huang, Robust simplex algorithm for online optimization, *Phys. Rev. Accel. Beams* **21**, 104601 (2018).
- [11] M. Aiba, M. Böge, N. Milas, and A. Streun, Ultralow vertical emittance at SLS through systematic and random optimization, *Nucl. Instrum. Methods Phys. Res., Sect. A* **694**, 133 (2012).
- [12] A. Scheinker, X. Pang, and L. Rybarcyk, Model-independent particle accelerator tuning, *Phys. Rev. ST Accel. Beams* **16**, 102803 (2013).
- [13] A. Scheinker, X. Huang, and J. Wu, Minimization of betatron oscillations of electron beam injected into a time-varying lattice via extremum seeking, *IEEE Trans. Control Syst. Technol.* **26**, 336 (2018).
- [14] A. Scheinker, D. Bohler, S. Tomin, R. Kammering, I. Zagorodnov, H. Schlarb, M. Scholz, B. Beutner, and W. Decking, Model-independent tuning for maximizing free electron laser pulse energy, *Phys. Rev. Accel. Beams* **22**, 082802 (2019).
- [15] X. Huang, J. Corbett, J. Safranek, and J. Wu, An algorithm for online optimization of accelerators, *Nucl. Instrum. Methods Phys. Res., Sect. A* **726**, 77 (2013).
- [16] X. Huang and J. Safranek, Online optimization of storage ring nonlinear beam dynamics, *Phys. Rev. ST Accel. Beams* **18**, 084001 (2015).
- [17] K. Tian, J. Safranek, and Y. Yan, Machine based optimization using genetic algorithms in a storage ring, *Phys. Rev. ST Accel. Beams* **17**, 020703 (2014).
- [18] X. Huang and J. Safranek, Nonlinear dynamics optimization with particle swarm and genetic algorithms for SPEAR3 emittance upgrade, *Nucl. Instrum. Methods Phys. Res., Sect. A* **757**, 48 (2014).
- [19] W. F. Bergan, I. V. Bazarov, C. J. R. Duncan, D. B. Liarte, D. L. Rubin, and J. P. Sethna, Online storage ring optimization using dimension-reduction and genetic algorithms, *Phys. Rev. Accel. Beams* **22**, 054601 (2019).
- [20] J. Liu, A. Moreau, M. Preuss, J. Rapin, B. Roziere, F. Teytaud, and O. Teytaud, Versatile black-box optimization, in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (IEEE, 2020), pp. 620–628, <https://ieeexplore.ieee.org/document/7352306>.
- [21] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, Taking the human out of the loop: A review of Bayesian optimization, *Proc. IEEE* **104**, 148 (2015).
- [22] J. Kirschner, M. Nonnenmacher, M. Mutný, N. Hiller, A. Adelman, R. Ischebeck, and A. Krause, Bayesian optimization for fast and safe parameter tuning of SwissFEL, in *Proceedings of the International Free-Electron Laser Conference (FEL2019)* (2019).
- [23] J. Duris, D. Kennedy, A. Hanuka, J. Shtalenkova, A. Edelen, P. Baxevanis, A. Egger, T. Cope, M. McIntire, S. Ermon *et al.*, Bayesian Optimization of a Free-Electron Laser, *Phys. Rev. Lett.* **124**, 124801 (2020).
- [24] A. Hanuka, X. Huang, J. Shtalenkova, D. Kennedy, A. Edelen, Z. Zhang, V. R. Lalchand, D. Ratner, and J. Duris, Physics model-informed Gaussian process for online optimization of particle accelerators, *Phys. Rev. Accel. Beams* **24**, 072802 (2021).
- [25] R. Roussel, A. Hanuka, and A. Edelen, Multiobjective Bayesian optimization for online accelerator tuning, *Phys. Rev. Accel. Beams* **24**, 062801 (2021).
- [26] P. T. Boggs and J. W. Tolle, Sequential quadratic programming, *Acta Numer.* **4**, 1 (1995).
- [27] R. H. Byrd, M. E. Hribar, and J. Nocedal, An interior point algorithm for large-scale nonlinear programming, *SIAM J. Optim.* **9**, 877 (1999).
- [28] J. R. Gardner, M. J. Kusner, Z. E. Xu, K. Q. Weinberger, and J. P. Cunningham, Bayesian optimization with inequality constraints, in *Proceedings of the 31st International Conference on Machine Learning*, edited by E. P. Xing and T. Jebara, Proceedings of Machine Learning Research Vol. 32 (PMLR, Beijing, China, 2014), pp. 937–945, <https://proceedings.mlr.press/v32/gardner14.html>.
- [29] J. M. Hernández-Lobato, M. A. Gelbart, R. P. Adams, M. W. Hoffman, and Z. Ghahramani, A general framework for constrained Bayesian optimization using information-based search, *J. Mach. Learn. Res.* **17**, 1 (2016), <https://jmlr.org/papers/v17/15-616.html>.
- [30] D. Eriksson and M. Poloczek, Scalable constrained Bayesian optimization, in *International Conference on Artificial Intelligence and Statistics* (PMLR, 2021), pp. 730–738, <https://proceedings.mlr.press/v130/eriksson21a.html>.
- [31] A. Beck, A. Ben-Tal, N. Guttmann-Beck, and L. Tretushvili, The CoMirror algorithm for solving nonsmooth constrained convex problems, *Oper. Res. Lett.* **4**, 1 (2020).
- [32] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, Safe exploration for optimization with Gaussian processes, in *Proceedings of the 32nd International Conference on Machine Learning* (PMLR, Lille, France, 2015), Vol. 37, pp. 997–1005, <https://proceedings.mlr.press/v37/sui15.html>.
- [33] M. Seidel, S. Adam, A. Adelman, C. Baumgarten, Y. Bi, R. Doelling, H. Fitze, A. Fuchs, M. Humbel, J. Grillenberger, D. Kiselev, A. Mezger, D. Reggiani, M. Schneider, J. Yang, H. Zhang, and T. Zhang, Production of a 1.3 MW proton beam at PSI, *IPAC 2010 - 1st International Particle Accelerator Conference* (IEEE, Kyoto, Japan, 2010), pp. 1309–1313, <https://accelconf.web.cern.ch/IPAC10/papers/tuyra03.pdf>.
- [34] M. Daum, P. Duperrex, G. Dzieglewski, U. Frei, T. Korhonen, A. Mezger, U. Müller, and D. Reggiani, A macropulsed 1.2 MW proton beam for the PSI ultracold neutron source, in *Proceedings of the 23rd Particle Accelerator Conference, Vancouver, Canada, 2009* (IEEE, Piscataway, NJ, 2009), pp. 1748–1750.
- [35] C. J. Milne *et al.*, SwissFEL: The Swiss x-ray free electron laser, *Appl. Sci.* **7**, 720 (2017).

- [36] E. Prat *et al.*, A compact and cost-effective hard x-ray free-electron laser driven by a high-brightness and low-energy electron beam, *Nat. Photonics* **14**, 748 (2020).
- [37] P. Juranić, J. Rehanek, C. A. Arrell, C. Pradervand, R. Ischebeck, C. Erny, P. Heimgartner, I. Gorgisyan, V. Thominet, K. Tiedtke, A. Sorokin, R. Follath, M. Makita, G. Seniutinas, C. David, C. J. Milne, H. Lemke, M. Radovic, C. P. Hauri, and L. Patthey, SwissFEL Aramis beamline photon diagnostics, *J. Synchrotron Radiat.* **25**, 1238 (2018).
- [38] A. Hanuka, X. Huang, J. Shtalenkova, D. Kennedy, A. Edelen, Z. Zhang, V. R. Lalchand, D. Ratner, and J. Duris, Physics model-informed Gaussian process for online optimization of particle accelerators, *Phys. Rev. Accel. Beams* **24**, 072802 (2021).
- [39] R. R. Duivenvoorden, F. Berkenkamp, N. Carion, A. Krause, and A. P. Schoellig, Constrained Bayesian optimization with particle swarms for safe adaptive controller tuning, *IFAC-PapersOnLine* **50**, 11800 (2017).
- [40] F. Berkenkamp, A. Krause, and A. P. Schoellig, Bayesian Optimization with Safety Constraints: Safe and Automatic Parameter Tuning in Robotics (2021), <https://link.springer.com/article/10.1007/s10994-021-06019-1>.
- [41] N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger, Gaussian process optimization in the bandit setting: No regret and experimental design, *Proceedings of the 27th International Conference on International Conference on Machine Learning* (Omni Press, Madison, WI, USA, 2010), pp. 1015–1022.
- [42] C. E. Rasmussen, Gaussian processes in machine learning, in *Advanced Lectures on Machine Learning* (Springer, New York, 2004), pp. 63–71.
- [43] P. Auer, Using confidence bounds for exploitation-exploration trade-offs, *J. Mach. Learn. Res.* **3**, 397 (2003), <https://www.jmlr.org/papers/v3/auer02a.html>.
- [44] S. R. Chowdhury and A. Gopalan, On kernelized multi-armed bandits, in *International Conference on Machine Learning* (PMLR, Sydney, Australia, 2017), pp. 844–853, <https://proceedings.mlr.press/v70/chowdhury17a.html>.
- [45] F. Berkenkamp, A. P. Schoellig, and A. Krause, No-regret Bayesian optimization with unknown hyperparameters, *J. Mach. Learn. Res.* **20**, 1 (2019), <https://www.jmlr.org/papers/v20/18-213.html>.
- [46] R. Combes and A. Proutiere, Unimodal bandits: Regret lower bounds and optimal algorithms, in *International Conference on Machine Learning* (PMLR, Beijing, China, 2014), pp. 521–529, <https://proceedings.mlr.press/v32/combes14.html>.
- [47] H. Saber, P. Ménard, and O.-A. Maillard, Forced-exploration free strategies for unimodal bandits, [arXiv:2006.16569](https://arxiv.org/abs/2006.16569).
- [48] A. Kolano, A. Adelman, R. Barlow, and C. Baumgarten, Intensity limits of the PSI injector II cyclotron, *Nucl. Instrum. Methods Phys. Res., Sect. A* **885**, 54 (2018).
- [49] C. Markovits, The 870 keV high intensity proton beam transfer line for the injector II of SIN, in *Proceedings of the 1987 Particle Accelerator Conference* (IEEE, Washington, DC, 1987), Vol. 3, p. 1954.
- [50] M. Mutný and A. Krause, Efficient high dimensional Bayesian optimization with additivity and quadrature Fourier features, in *Neural and Information Processing Systems (NeurIPS)* (Curran Associates Inc., Red Hook, NY, USA, 2018), pp. 9019–9030.
- [51] D. Calandriello, L. Carratino, A. Lazaric, M. Valko, and L. Rosasco, Gaussian process optimization with adaptive sketching: Scalable and no regret, *Proceedings of the Thirty-Second Conference on Learning Theory*, edited by A. Beygelzimer and D. Hsu, Proceedings of Machine Learning Research Vol. 99 (2019), <https://proceedings.mlr.press/v99/calandriello19a.html>.
- [52] Y. Abbasi-Yadkori, Online Learning for Linearly Parametrized Control Problems, Ph.D. thesis, Department of Computing Science, University of Alberta, 2012.
- [53] M. Kanagawa, P. Hennig, D. Sejdinovic, and B. K. Sriperumbudur, Gaussian processes and kernel methods: A review on connections and equivalences, [arXiv:1807.02582](https://arxiv.org/abs/1807.02582).