

Customized Bayesian optimization for efficient beam tuning at the facility for rare isotope beams

Kilean Hwang^{✉,*}, Qiang Zhao, Tong Zhang[✉], Tomofumi Maruta[✉], Alexander Plastun[✉],
Kei Fukushima[✉], and Peter Ostroumov[✉]

*Facility for Rare Isotope Beams (FRIB), Michigan State University,
640 South Shaw Lane, East Lansing, Michigan 48824, USA*



(Received 29 August 2025; accepted 13 January 2026; published 17 February 2026)

Bayesian optimization (BO) has recently emerged as a powerful approach for on-line beam tuning, and it is rapidly gaining adoption across accelerator facilities due to its flexibility and efficiency in handling complex optimization tasks. At the Facility for Rare Isotope Beams, rapid and reliable tuning is essential to support the delivery of diverse ion species. To improve the practicality of BO in this setting, we implemented several enhancements, including scalarized composite objective construction for multicriteria optimization, asynchronous evaluation for better resource utilization, prior-mean-assisted optimization to accelerate convergence, and a local search strategy for rapid completion of the task. We present the details of these methods, discuss challenges-encountered, and share our experience applying them to specific beam-tuning tasks.

DOI: [10.1103/5hyd-bhpx](https://doi.org/10.1103/5hyd-bhpx)

I. INTRODUCTION

The Facility for Rare Isotope Beams (FRIB) houses a high-power heavy-ion linear accelerator capable of delivering a wide variety of primary ion species to produce requested isotopes for nuclear science experiments [1,2]. This unique capability requires frequent switching of primary beam ion species, each necessitating the commissioning of the accelerator setup [3,4]. Such operational demands impose strict time constraints, making rapid yet high-quality beam tuning essential for maximizing scientific productivity.

To address this challenge, we have developed and implemented physics-model-based methods and several machine learning (ML) methods for accelerator tuning. A key advantage of ML approaches is their ability to exploit diverse types of data—archived operational data, simulation results, and on-line measurements—in a computationally efficient manner. By contrast, traditional black-box optimization algorithms such as Nelder-Mead [5] typically rely only on information from the most recent iterations, which limits their data efficiency. Incorporating information from all the previous optimization steps, along with historical and simulated data, allows ML-based optimizers to make better-informed decisions and converge more quickly toward optimal machine settings.

Among ML approaches, Bayesian optimization (BO) has gained particular popularity at accelerator facilities due to its sample efficiency, interpretability, and relatively straightforward implementation [6–8]. In this work, we present a customized BO framework to improve FRIB beam tuning, where each objective function evaluation on the machine takes a few seconds. BO can efficiently leverage all on-line data collected during tuning, while also incorporating off-line information—such as archived operational records and physics-based simulation outputs. This integrated use of on-line and off-line knowledge enables faster convergence to high-quality beam configurations, reducing setup time and enhancing facility availability for experiments.

This paper is structured as follows: Sec. II provides an overview of why BO is particularly effective for beam tuning, including its sample efficiency and a comparative analysis with reinforcement learning. Section III details the specific customizations we have developed to adapt BO for the operational demands of FRIB, such as asynchronous and cost-aware strategies. Finally, Sec. IV presents successful applications of our customized BO framework to beam-tuning tasks in the FRIB accelerator.

II. WHY BAYESIAN OPTIMIZATION IS EFFECTIVE FOR BEAM TUNING

BO has rapidly gained traction at accelerator facilities due to its ability to efficiently handle expensive, noisy, and complex optimization tasks. Its strength lies in combining probabilistic surrogate modeling with principled exploration—exploitation strategies, making it particularly well suited for on-line beam tuning, where both evaluation time and data

*Contact author: hwang@frib.msu.edu

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

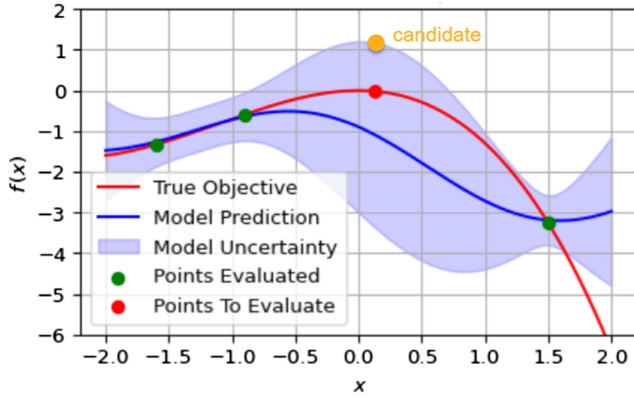


FIG. 1. Illustration of BO. The shaded area represents model uncertainty, which is larger in regions far from the observation points marked by green dots.

are limited. Below, we highlight several key reasons for its effectiveness in this context.

A. Sample efficiency

BO leverages probabilistic surrogate models, most commonly Gaussian processes (GP), to guide the search for optimal machine settings. These probabilistic models enable a principled balance between exploration of uncertain regions and exploitation of promising solutions (Fig. 1). This strategy allows BO to converge using as few machine evaluations of the objective as possible. Beam tuning tasks are inherently constrained by evaluation costs: each trial often requires magnet-power-supply ramping—taking several seconds—followed by averaging multiple beam measurements to suppress noise. Such time-consuming operations sharply limit the feasible number of evaluations. BO’s sample efficiency is therefore highly advantageous, ensuring that valuable machine time is used as effectively as possible. Moreover, surrogate model training and acquisition function optimization typically complete within seconds, which is about the same cost as a machine evaluation, allowing them to run in parallel with beam measurements without introducing additional overhead.

B. Robustness to noise

Beam diagnostics are inevitably affected by noise due to instrumentation limits and fluctuating beam conditions. BO can naturally account for this by incorporating a learnable noise parameter in its probabilistic model, typically represented as an additive Gaussian noise term σ^2 in the GP likelihood as shown below [9].

$$\begin{aligned} \log p(\mathbf{y}|\mathbf{X}) = & -\frac{1}{2}(\mathbf{y} - \mathbf{m})^T(\mathbf{K} + \sigma^2\mathbf{I})^{-1}(\mathbf{y} - \mathbf{m}) \\ & -\frac{1}{2}\log |\mathbf{K} + \sigma^2\mathbf{I}| - \frac{n}{2}\log 2\pi. \end{aligned} \quad (1)$$

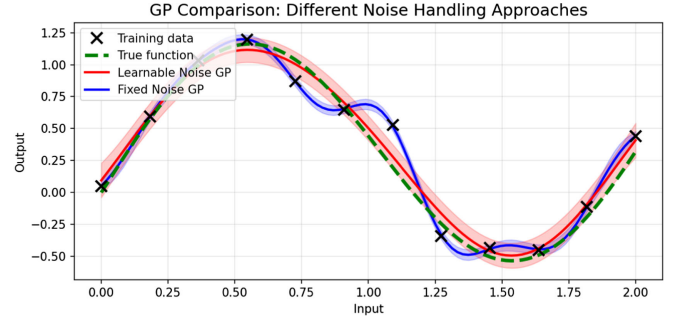


FIG. 2. Illustration of GP regression with and without noise modeling.

Here, $\mathbf{X}$ denotes the training input data, $\mathbf{y}$ the observed training outputs, $\mathbf{m}$ the mean function evaluated at $\mathbf{X}$, $\mathbf{I}$ the identity matrix, n the number of training data points, and $\mathbf{K} = K(\mathbf{X}, \mathbf{X})$ is the kernel matrix with K being the GP kernel function.

Figure 2 demonstrates this robustness by learning the noise variance σ^2 , the GP posterior avoids overfitting to noisy data.

C. Comparison with reinforcement learning

Reinforcement learning (RL) is a powerful framework for sequential decision making, typically formalized as Markov Decision Processes [10]. The goal is to learn a *policy* $\pi(a|s)$ that maps a state s to an action a . The typical RL objective is to maximize the expected cumulative discounted reward:

$$J(\pi) = \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R_t \right], \quad (2)$$

where R_t is the reward at time t and γ is the discount factor that determines how far into the future the agent reasons about long-horizon consequences.

In contrast, BO is specifically designed for optimization of expensive black-box functions $f(x)$. The objective is to identify:

$$x^* = \arg \max_x f(x). \quad (3)$$

BO is known for its high sample efficiency, largely due to its probabilistic surrogate modeling—frequently Gaussian processes (GPs). This makes BO especially suitable for accelerator applications where each evaluation of $f(x)$ (e.g., a knob ramp or a beam diagnostic measurement) consumes valuable beam time [7,8].

Because RL and BO are designed for fundamentally different problem classes (long-horizon sequential control versus static black-box optimization), a direct comparison is inherently limited. Nevertheless, if RL is cast to learn a policy to optimize a static objective function $f(x)$, this

learned policy may be compared to BO, although this is a very narrow aspect of RL. Most RL methods are significantly more data-hungry than BO [11]. Only a narrow family of model-based RL approaches achieves BO-like sample efficiency, and crucially, these methods explicitly incorporate probabilistic modeling such as GP models or Bayesian ensembles, making them closer to BO [12,13].

Without substantial pretraining—whether on-line or via large off-line datasets—typical RL agents struggle to compete with BO in on-line, cold-start optimization tasks. When abundant off-line data (such as high-fidelity physics simulations) are available, pretrained RL policies can achieve near-instant on-line tuning [14], provided the off-line data distribution matches the on-line environment. However, for a fair comparison, BO can similarly incorporate off-line information via pretraining prior model or fast differentiable simulations as direct prior models [15–17].

The critical distinction lies in robustness to *distribution mismatch* (Sim2Real gap). While it is nontrivial for pretrained RL policies to adapt to distribution mismatch [18,19], BO naturally and sample efficiently adapts its probabilistic surrogate to new data, in the presence of distributional shifts and modeling errors [16,17].

Finally, it is important to acknowledge that RL remains the superior framework for dynamic control tasks involving long-horizon cumulative objectives. For example, end-to-end beam tuning across multiple coupled accelerator segments may benefit from RL-style sequential decision making instead of segment-by-segment tuning independently. However, for static problems, BO generally remains more sample efficient.

III. CUSTOMIZATIONS TO BO FOR EFFICIENT BEAM TUNING

We customized BO to address the specific requirements of beam-tuning tasks, with a primary emphasis on reducing tuning time [20].

A. Single asynchronous BO

To minimize machine idle time during BO, we implemented an asynchronous evaluation strategy. In this approach, the machine evaluates the current candidate control setting while BO computations proceed in parallel to select the next candidate control setting as illustrated in Fig. 3. Specifically, since the machine can evaluate only one set point at a time, this is referred to as single asynchronous BO. Since the outcome of the current evaluation is not yet available, the selection of the next candidate must avoid the point currently being evaluated.

It is important to acknowledge that asynchronous candidate selection induces a temporary *information lag*: The surrogate model computes the next candidate without knowledge of the pending evaluation’s outcome. While this theoretically reduces the per-step information gain

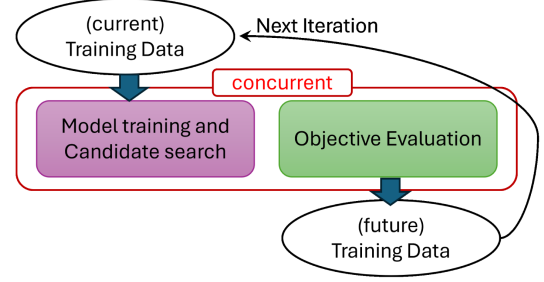


FIG. 3. Flowchart for asynchronous BO.

compared to a strictly sequential approach, empirical studies on penalization methods demonstrate that this degradation in sample efficiency is marginal in practice [21]. Conversely, the operational benefit is substantial; by ensuring the machine never idles waiting for computation, the asynchronous strategy significantly reduces the total wall-clock time required to achieve convergence.

Our approach follows a penalization strategy similar to Ref. [21], where the acquisition function is modified to discourage selection near the currently evaluating point. Specifically, we add the following penalization term:

$$f_{\text{penal}}(x) = -C_{\text{penal}} \prod_{i=1}^D \exp \left(-\frac{(x - x_{\text{penal}})_i^2}{L_{\text{penal},i}^2} \right), \quad (4)$$

where i indexes the control dimensions, D is the dimensionality of the control space. The penalization point x_{penal} is chosen to be the currently evaluating point, the penalization strength C_{penal} and length scales $L_{\text{penal},i}$ are automatically adapted based on the base acquisition function value at training data points or at sample points near the pending point. See Appendix A for details.

An alternative technique is to exploit *fantasization* [22], in which the unknown objective value at the currently evaluating point is temporarily “fantasized” using the model’s predictive distribution. The fantasized observation is then included in the training set to effectively guide the selection of the next candidate away from the fantasized point.

Yet another technique is to exploit the *joint predictive covariance* between the pending point and the candidate. By evaluating the acquisition function over the joint posterior distribution, the selection process intrinsically accounts for the statistical correlation between the currently evaluating point and the potential candidate. Although this is more information theoretic, we use an explicit penalization method for flexibility and explicit control of various aspects, not just penalizing the pending point, but also penalizing polarity crossing and ramping cost awareness.

Additionally, we designed the candidate selection process to terminate as soon as the machine completes its current evaluation, ensuring that new candidates can be evaluated without delay.

1. Computational complexity

Understanding the computational complexity is crucial for managing the time costs associated with BO. For standard BO using a GP surrogate model, the primary computational expenses stem from two key steps: fitting the GP model (including hyperparameter optimization) and optimizing the acquisition function (e.g., expected improvement or upper confidence bound).

GP fitting involves optimizing the log-marginal likelihood with respect to hyperparameters. Assuming gradient descent is used for optimization, the cost per iteration (including evaluation of the likelihood and its gradients) is $O(Dn^2 + n^3)$, where n is the number of training data points and D is the dimensionality of the input space. This arises from kernel matrix computation $O(Dn^2)$ and Cholesky decomposition $O(n^3)$ [23]. The total cost for fitting is multiplied by the number of iterations required by the optimizer.

Optimization of the acquisition function also incurs cost. Assuming gradient descent is used, the cost per iteration (including evaluation of the acquisition function and its gradients, with GP predictions) is $O(Dn + n^2)$. To address multimodality (i.e., multilocal optima), multistart strategies are commonly used, with the number of starts often scaling as $O(D)$ or higher. The total cost is multiplied by the number of optimizer iterations and the number of starts.

In practice, the combined time for GP fitting and acquisition optimization typically takes a few seconds on a moderately performant computer for moderate number of data (e.g., $n \lesssim 100$) and dimensionality (e.g., $D \leq 6$).

The fantasization approach is computationally more intensive, as it requires Monte Carlo sampling S fantasy outcomes from the GP posterior at the pending point that the machine is currently evaluating and refitting or conditioning GPs for each. The acquisition function is then averaged over these fantasies, adding further computational overhead of $O(S(n+1)^2)$ to the optimization process. In contrast, the penalization strategy modifies the acquisition function directly, incurring negligible additional complexity beyond standard optimization.

Therefore, when objective evaluation time on the real system is less than a few seconds, lightweight modeling and acquisition functions (such as penalization) are preferable to prevent BO from becoming the computational bottleneck.

B. Cost-aware BO

To mitigate time-cost associated with power supply ramping, we augment the acquisition function with favoring terms:

$$f_{\text{favor}}(x) = C_{\text{favor}} \prod_{i=1}^D \exp\left(-\frac{(x_i - x_{\text{favor},i})^2}{L_{\text{favor},i}^2}\right) + C_{\text{polar}} \prod_{i=1}^D \delta_K\left(\frac{x_{\text{favor},i}}{|x_{\text{favor},i}|}, \frac{x_i}{|x_i|}\right), \quad (5)$$

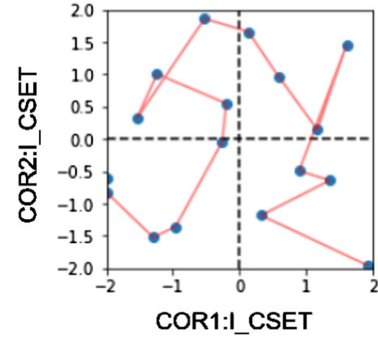


FIG. 4. Optimized evaluation path for corrector settings, designed to minimize ramping time costs during batch evaluations [6].

where x_{favor} denotes the point currently under evaluation. The first term favors the selection of candidates proximate to the current point, reducing the magnitude of control adjustments, while the second term, with the Kronecker delta δ_K , favors polarity conservation. This is critical, as polarity changes in the FRIB corrector magnet power supply incur a delay of over 15 s.

Note that the first term counteracts the penalization in Eq. (4) if $L_{\text{penal},i} = L_{\text{favor},i}$. To prevent the favoring term from negating the effect of the penalization, we ensure that the length scales satisfy $L_{\text{favor},i} \gg L_{\text{penal},i}$. This configuration preserves the penalization's effectiveness while promoting efficient ramping. Further details are in Appendix A.

Additionally, when evaluating batches of candidate points—such as during the initialization phase with random candidates—the evaluation sequence is optimized to minimize ramping costs. This optimized path, depicted in Fig. 4, reduces the cumulative time spent transitioning between control settings. Specifically, starting from the current set point, we find next point among candidate points that results in smallest ramping cost given ramping rate and polarity change time.

C. Objective function construction

The performance of any optimization algorithm strongly depends on the landscape of its objective function. An ill-posed or rugged landscape can impede convergence, regardless of the optimizer's sophistication. Consequently, the careful construction of a smooth and well-behaved objective is as critical as the choice of the optimization algorithm itself. We adopted a two-stage design for objective function construction: first, a scalarization framework to aggregate multiple objectives, and second, applying a regularization transform.

1. Scalarization

Practical beam tuning requires the simultaneous optimization of multiple, sometimes competing, objectives. Instead of applying multiobjective BO, we employ

scalarization to combine objectives into a single composite function. While multiobjective BO is a principled approach, it is (i) computationally more expensive both in terms of surrogate modeling and candidate searching, (ii) less suited for scenarios where local BO is required as the Pareto-front construction generally requires global exploration, and (iii) still requires a postdecision rule to select a single operating point from the Pareto set.

We formulate composite objective $O_{\text{composite}}$ as

$$O_{\text{composite}} = \sum_i w_i \varphi_i \left(\frac{v_{i,\text{meas}} - v_{i,\text{target}}}{\tau_i} \right), \quad (6)$$

where for the i th objective, w_i is its assigned weight, $v_{i,\text{meas}}$ is the measured value, $v_{i,\text{target}}$ is the desired target, and τ_i is a user-defined tolerance that normalizes the deviation into a dimensionless quantity. This normalization is crucial for meaningfully combining disparate measurements. The weight w_i becomes meaningful after the normalization. For example, in our beam-tuning task, we typically (though not always) require the beam centroid at BPM to be within $\tau_{x,y} = 1$ mm transversely, and $\tau_\phi = 1^\circ$ longitudinally.

The shaping function $\varphi(\cdot)$ translates the normalized deviation into an individual objective. For example, we mostly use (i) targeting objectives (e.g., beam centering): $\varphi(v) = 1 - v^2$ and (ii) maximization objectives (e.g., transmission maximization): $\varphi(v) = -\text{ELU}(-v)$

where exponential linear unit function is

$$\text{ELU}(v) = \begin{cases} v, & v > 0, \\ \exp(v) - 1, & v \leq 0. \end{cases} \quad (7)$$

This formulation encourages v_{meas} to exceed v_{target} . However, once the measured value surpasses the target by more than the tolerance (i.e., $v_{\text{meas}} > v_{\text{target}} + \tau$), the marginal benefit saturates asymptotically. An example of a composite objective configuration used for MEBT tuning is in Appendix B.

Finally, the scalarized output $O_{\text{composite}}$ from Eq. (6) is passed through the BiLog transformation for regularization and to emphasize the region near the optimum.

2. Regularization

The regularity of the objective function has a significant impact on optimization performance. We design the objective such that the maximum value is 1 and apply a BiLog transformation, defined as $y \rightarrow \text{sign}(y) \log(1 + |y|)$, to regularize the function and better resolve the region near the optimum.

Figure 5 illustrates the effect of the BiLog transformation [24]. Note that in the original objective function, large variations away from the optimum can obscure the relatively small variations near the peak. This increases the number of samples needed to resolve the optimum region accurately. The transformed objective compresses large

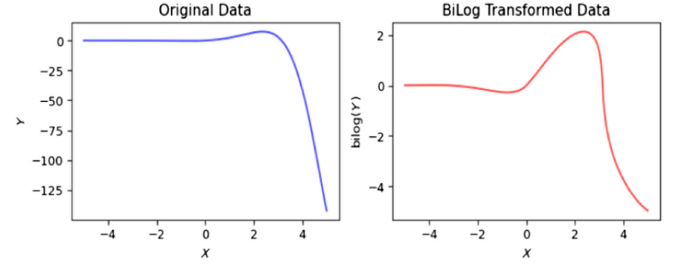


FIG. 5. Illustration of BiLog transformation.

values far from the optimum, allowing the surrogate model to better resolve the landscape near the optimum. The BiLog transformation mitigates this imbalance by compressing the scale of large values while preserving detail near the optimum.

D. Local optimization

While BO offers superior sample efficiency, performing an on-line global search in control spaces of moderate to high dimensionality ($D > 5$) under strict time constraints can be impractical. To address this, we employ local BO once a predefined global-evaluation budget has been consumed. Local optimization is realized by confining the search domain to a narrow hypercube centered on the current best-found point. We typically use about 10% of the global hard bound for the length of the hypercube. This hypercube is recentered on the updated optimum, focusing sampling in the most promising region (Fig. 6). It has the following benefits: (i) Enhanced model accuracy: In high-dimensional spaces, on-line training data are inherently sparse. By restricting the search to a smaller local region, we effectively increase data density, enabling more accurate surrogate modeling within the relevant subspace. (ii) Hysteresis mitigation: Smaller, incremental control-knob adjustments limit magnet-power-supply excursions, thereby reducing magnetic hysteresis errors. (iii) Implicit safety: The local search strategy serves as an implicit safety constraint by inherently preventing

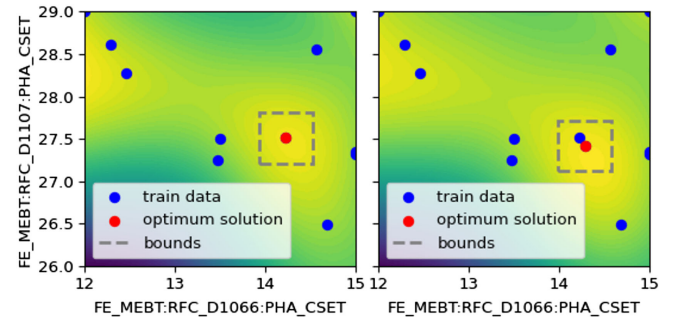


FIG. 6. Evolution of the local BO search domain over the model's predicted objective contours [6]. Left: transition from global to local search, centered on the current best point. Right: recentered local search region in the subsequent iteration. Brighter colors indicate higher predicted objective values.

large, hazardous excursions into parameter regions that risk significant beam loss. This approach avoids the computational overhead of explicit constraint modeling. (iv) Drift minimization: Accelerating convergence shortens the overall tuning duration, mitigating the effects of slow machine drifts and eliminating the need for explicit time-context modeling.

E. Prior-mean-assisted BO

When a corpus of off-line data—either from historical operational archives or simulations—is available, an off-line model can be trained to encapsulate existing knowledge of the system. Such an off-line model can be integrated into BO in various ways, but its incorporation as the prior mean of the GP surrogate model is a particularly effective strategy. This approach, known as prior-mean-assisted BO (pmBO) [16], is conceptually simple, computationally lightweight and requires no modification to the standard GP framework beyond the initial specification of the prior mean function.

The primary advantage of pmBO is its ability to accelerate convergence. If the off-line data can well represent the on-line physical machine, the optimization can converge almost instantaneously. More critically, pmBO is robust to distribution shift between the off-line model and the on-line environment [16,25]. The BO framework naturally adapts to this discrepancy by strategically collecting the most informative on-line data to update its posterior belief, thereby correcting the initial bias from the prior mean in a sample-efficient manner. The conceptual process of pmBO adapting to machine drift is illustrated in Fig. 7.

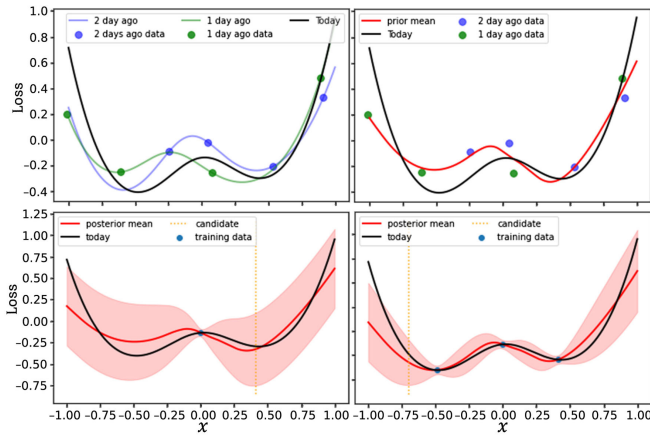


FIG. 7. Conceptual illustration of pmBO under machine drift [26]. Top-left: a hypothetical 1D loss function (to be minimized) across three days, illustrating daily machine variation. Colored lines show loss functions for different days; dots represent archived data used for off-line model training. Top-right: prior mean model (e.g., neural network) trained on archived data. Bottom-left: GP posterior after one on-line evaluation on the current machine. Bottom-right: GP posterior after two iterations of BO using the prior mean and updated observations. Orange vertical line corresponds to the candidate solution queried by BO.

We have experimentally validated the efficacy of pmBO for the front-end tuning of the Facility for Rare Isotope Beams (FRIB) accelerator. In this application, a neural network (NN) was trained on an off-line dataset of 539 historical data points corresponding to the tuning of an $^{36}\text{Ar}^{10+}$ beam [16]. The NN model was configured to predict beam properties based on 38 input features, including settings for solenoids, electric quadrupoles, and electric correctors. For the on-line optimization task, a subset of three critical devices—one horizontal corrector, one vertical corrector, and one solenoid—were selected as the control knobs. The objective was to minimize beam offset and maximize transmission. The pretrained NN served as the prior mean for the BO algorithm, which then guided the on-line tuning. The results demonstrated that the pmBO framework was able to find a near-instant solution, significantly outperforming what would be expected from a standard BO approach with a zero-mean prior.

IV. BEAM TUNING APPLICATIONS

BO can be applied to many beam-tuning tasks. Here we list some of the tasks that we often use.

A. Beam centering through apertures in the U-LEBT

The FRIB Low Energy Beam Transport (LEBT) section, as depicted in Fig. 8, is not equipped with beam position monitors (BPMs) because the beam is unbunched in this region. This absence of noninterceptive diagnostics makes beam centroid tuning challenging. To overcome this, we use two apertures in a straight section of the Upper LEBT (U-LEBT). The apertures were installed to cut the beam halo. They are separated by 60° of transverse betatron phase

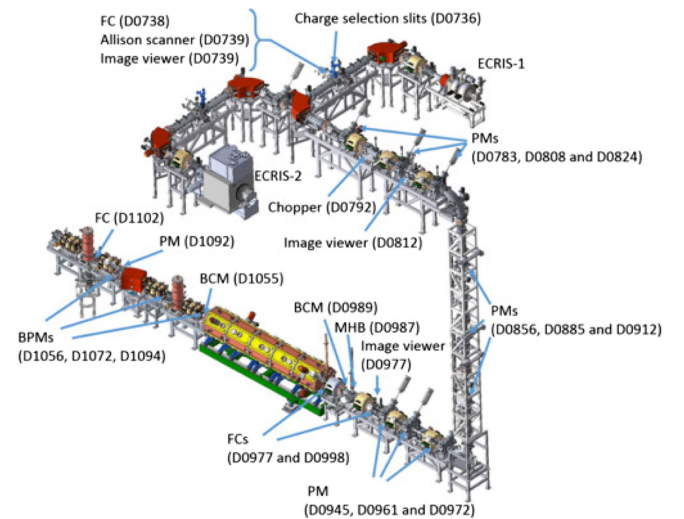


FIG. 8. Layout of the LEBT to MEBT with the location of beam instrumentation. Each device has a decimeter number showing the location along the FRIB beam line [27].

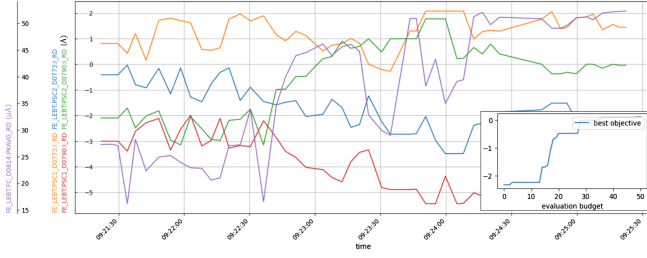


FIG. 9. Optimization progress through the two apertures in the U-LEBT. The horizontal and vertical corrector magnets are labeled PSC2 and PSC1, respectively. The beam current measured at FC is also shown. The inset plot displays the history of the cumulative best value of the normalized objective over the optimization iterations. The time axis is formatted in day–hour–minute.

advance—(rather than the ideal 90° , due to other constraints) for effective halo removal across the full phase space. By deactivating the corrector magnets between the apertures, a straight-line trajectory is enforced. The optimization objective is to maximize the beam current measured at a downstream Faraday cup (named by FC_D0814 which is not shown in the Fig. 8, but is just before the vertical dipole) by controlling two pairs of upstream corrector magnets (a four-dimensional parameter space). Proper beam centering through the apertures ensures symmetric halo removal, whereas a misaligned beam would only clip the halo on one side, leading to residual particles that can be accelerated and cause downstream losses.

Figures 9 and 10 illustrate a recent (October 4, 2025) application of $^{238}\text{U}^{35+}$ beam centering through two apertures by maximizing the beam current measured at a Faraday cup (FC). To visualize the model-predicted objective surface over the four control parameters, we construct two-dimensional projections by taking the maximum along the nonplotted directions:

$$\text{project}(x_i, x_j) = \max_{x_k \notin \{i,j\}} \mu(x),$$

where μ is the GP posterior mean and $x \in \mathbb{R}^4$ denotes the control vector. This projection method makes it possible to

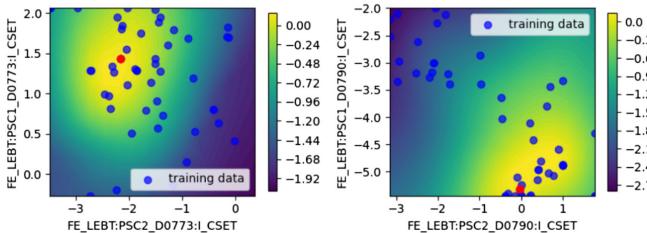


FIG. 10. Model-predicted mean objective values visualized on two pairwise projections of the four control parameters.

clearly highlight the locations of the predicted optima within the projected space. Note that the training data are localized, especially for the plot on the right in Fig. 10, due to the use of a moving window in the local optimization process.

In most operational scenarios, loading magnet settings scaled from an archived setpoint according to the new ion's charge-to-mass ratio yield a nonzero initial beam current at FC_D0814, even with the apertures inserted. Because the system starts near a viable solution, a local BO is highly effective. This approach typically converges to an optimal solution in approximately 3 min. This robust method is also routinely applied in various sections of the Reaccelerator (REA) facility, supported by a dedicated Graphical User Interface (GUI).

B. RFQ transmission and MEBT beam centering

Experience has shown that after performing LEBT beam centering, the RFQ transmission and beam position in the MEBT are typically not very far from their optimal values in most cases. The inability to fully restore the exact same state is primarily due to irreproducible beam quality from the ion source and to magnetic hysteresis effects. Therefore, we also apply local BO in most cases for MEBT tuning. The goal is a composite objective: (i) matching the transverse (x , y) and longitudinal (z) beam centroids at three MEBT BPMs to their archived target values and (ii) maximizing beam current as measured at two downstream locations: the beam current monitor at the end of the RFQ (BCM_D1055) and a FC further downstream (FC_D1102), as shown in Fig. 8. This goal is based on the assumption that restoring the RFQ injection trajectory, which is quantified by MEBT 6D BPM positions and beam transmission, the beam envelope is also likely to be restored as RFQ effectively serves to shape transverse beam envelope as well as longitudinal bunching and acceleration. This means that MEBT settings are fixed to the saved set, and only LEBT correctors are used to optimize.

The specific transmission targets are set to greater than 99% at BCM_D1055 to align the beam trajectory through the RFQ center and greater than 80% at FC_D1102 for RFQ transmission of accelerated particles. The optimization typically manipulates six to eight LEBT corrector magnets with a budget of approximately 70 iterations, which completes in about 17 min. To mitigate the risk of machine damage from uncontrolled beam loss in the RFQ, the beam duty factor is limited to less than 5%, especially when global BO is performed. However, since local BO is standard for this task, its inherently smaller step sizes effectively prevent significant beam loss.

Figure 11 shows an application result with eight LEBT correctors as control knobs. Note that 6D beam centroid is well restored to the target values within tolerance except for BPM_D1072:XPOS while beam current is maximized.

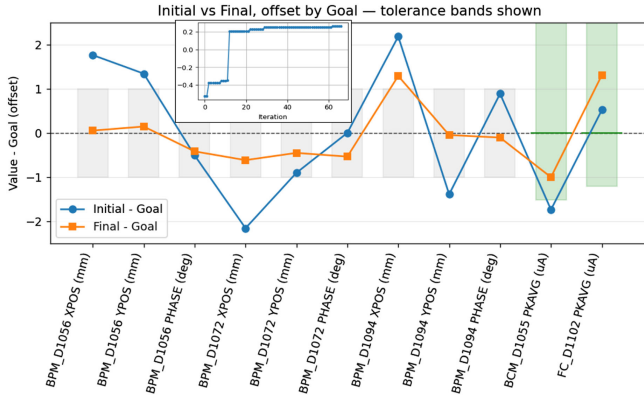


FIG. 11. Multiple objectives before and after optimization for MEBT beam centering. Vertical axis shows measured value subtracted by target (goal). The shaded area shows the chosen tolerance of each objective. The inset plot displays the history of the cumulative best value of the normalized objective over the optimization iterations.

C. MEBT buncher phase fine-tuning

Once MEBT tuning is complete, we send the beam to the SRF linac. Although MEBT tuning restores MEBT BPM phases as much as possible, the downstream BPM phases can oscillate from the archived target but not too far generally (within 1° at 80.5 MHz). The phases of the MEBT buncher cavities can be fine-tuned to minimize such oscillation using local BO. Figure 12 shows one such application result.

D. GUI application for ReA

The ReAccelerator (ReA) at FRIB is designed to reaccelerate stopped beams produced via in-flight fragmentation. To support beam-tuning workflows using BO, a dedicated graphical user interface (GUI) has been developed [28]. Due

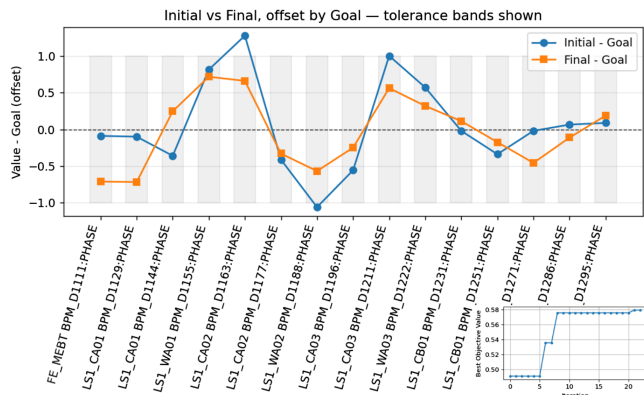


FIG. 12. Multiple objectives before and after optimization for linac phase restoration fine-tuning two MEBT bunching cavities. Vertical axis shows measured value subtracted by target (goal). The shaded area shows the chosen tolerance of each objective. The inset plot displays the history of the cumulative best value of the normalized objective over the optimization iterations.

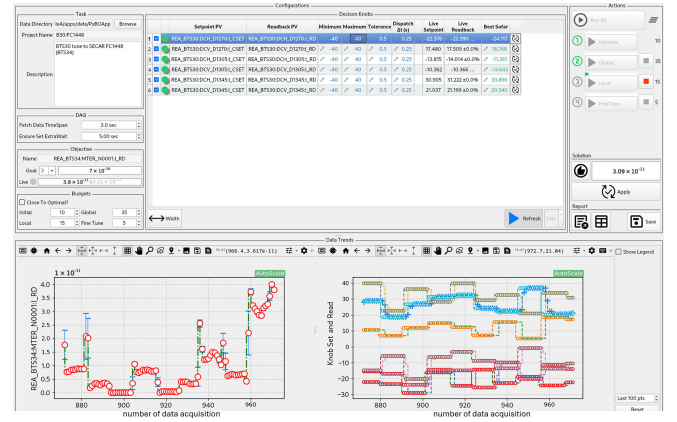


FIG. 13. Beam tuning result screenshot of GUI application of BO for ReA. MTER_N0001 represents the beam current reading at SECAR FC1448.

to the low beam current in ReA, the system is not equipped with BPMs. This makes the beam centering as well as beam transmission optimization challenging. Instead, ReA has a series of slit scanners distributed along the linac. Each slit scanner has thin rectangular openings that serve as beam-profile monitors, as well as a circular aperture. Traditionally, hand tuning with the beam image on screen was employed. Since the deployment of BO and GUI, the beam centering as well as transmission optimization are efficiently performed by inserting two adjacent slit scanners in a straight section with circular apertures positioned at the nominal beam axis and maximizing the transmitted beam current using corrector and quadrupole magnets. This procedure is taken at multiple sections of ReA. Figure 13 shows a screenshot of one such GUI application. In the example shown, six control knobs are used to maximize the beam current through two apertures measured at a downstream Faraday cup.

V. CONCLUSION

We have successfully developed and implemented a customized BO that enhances the efficiency of beam tuning at FRIB for tasks where each objective function evaluation per iteration on the machine takes a few seconds. The implementation of an asynchronous evaluation scheme minimizes machine idle time, while a cost-aware acquisition function intelligently reduces delays associated with hardware adjustments, such as magnet ramping and polarity crossing. Furthermore, our careful construction of the objective function—using a scalarization method to combine multiple beam objectives and a BiLog transformation to regularize the landscape—ensures smooth and efficient convergence. The transition from a global to a local search strategy further accelerates convergence in limited time and was one of the most practical approach with many benefits. Finally, the use of prior-mean-assisted BO provides a powerful means of leveraging historical data to generate an informed starting point, enabling near-instantaneous

convergence while remaining robust to distribution shifts caused by irreproducible beam quality from the ion source and by magnetic hysteresis. These customized methods have been successfully applied to beam-tuning tasks, including beam centering in the LEBT, optimizing RFQ transmission, and fine-tuning MEBT buncher phases. The principles and techniques presented here are broadly applicable for optimizing other complex, resource-constrained scientific instruments.

ACKNOWLEDGMENTS

This work is supported by the U.S. Department of Energy, Office of Science, Office of Nuclear Physics, under Award No. DE-SC0024707 and used resources of the FRIB Operations, which is a DOE Office of Science User Facility under Award No. DE-SC0023633.

DATA AVAILABILITY

Data for Figs. 1, 2, 4–7, and 14 are not publicly available. Data for Figs. 10–12 are not publicly available upon publication because it is not technically feasible and/or the cost of preparing, depositing, and hosting the data would be prohibitive within the terms of this research project. The data are available from the authors upon reasonable request.

APPENDIX A: PENALIZATION AND FAVORIZATION PARAMETERS

In our application, penalization is primarily used to avoid selecting the pending point, whereas favorization encourages candidate search to reduce power-supply ramping costs. Accordingly, we determine the associated parameters from the training data near the pending point and from the values of the bare acquisition function, $\alpha_{\text{bare}}(\cdot)$, before applying any modifications. The default procedure is as follows: (i) For each input dimension, get the n smallest distances (for each dimension independently) L_{sorted} from the pending point x_{penal} , with $n = \min(2d, n_{\text{train}})$. Here d is the input dimension and $n_{\text{train}} > 1$ is assumed for the number of training data. (ii) Set the penalization length scale to half of the mean of the selected distances,

computed independently for each dimension: $L_{\text{penal}} = 0.5\langle L_{\text{sorted}} \rangle$. (iii) Draw 8d random samples $x_s \sim \mathcal{N}(x_{\text{penal}}, L_{\text{penal}})$ and evaluate bare acquisition function $\alpha_{\text{bare}}(x_s)$ for all $x_s \in \chi_s$. (iv) Choose penalization coefficient by $C_{\text{penal}} = \max_{\chi_s}(\alpha_{\text{bare}}) - \min_{\chi_s}(\alpha_{\text{bare}})$.

This way, the penalization reflects both the local data length scale and the local variability of the bare acquisition function. Favorization parameters are chosen analogously, using $L_{\text{favor}} = 5\langle L_{\text{sorted}} \rangle$ and $C_{\text{favor}} = 0.5[\max_{\chi_s}(\alpha_{\text{bare}}) - \min_{\chi_s}(\alpha_{\text{bare}})]$. As for the polarity penalty, we use a global χ_s rather than local estimate of the acquisition function range. Specifically, we take 20% of the range of the bare acquisition function evaluated over all the training data points: $C_{\text{polar}} = 0.2[\max_{\chi_{\text{train}}}(\alpha_{\text{bare}}) - \min_{\chi_{\text{train}}}(\alpha_{\text{bare}})]$, where χ_{train} is the set of all input training data.

Figure 14 illustrates the resulting penalization and favorization effects. Note that the pending point is effectively penalized while the polarity is favorized. The favorization around the pending point is not visual because of a weak favorization coefficient.

These choices are based on simple heuristics and visualization of the acquisition function but have not been thoroughly studied or optimized.

APPENDIX B: COMPOSITE OBJECTIVE CONSTRUCTION EXAMPLE

To demonstrate the practical application of our scalarization framework, we present one of the configurations used for tuning the MEBT section as an example in Table I. This section requires simultaneous optimization of beam transmission and beam centroid. The objective function aggregates measurements from three beam position monitors (BPMs), one beam current monitor (BCM), and a Faraday cup (FC). The default shape functions in Sec. III C 1 are used. Since we are controlling LEBT correctors while fixing MEBT magnets, we put less weight on more downstream BPMs. The target values of the BCM and FC are 99% and 80%, respectively, of the measured beam current from the upstream FC, following Sec. IV B. BPM target values are from a saved previously operated “good tune.”

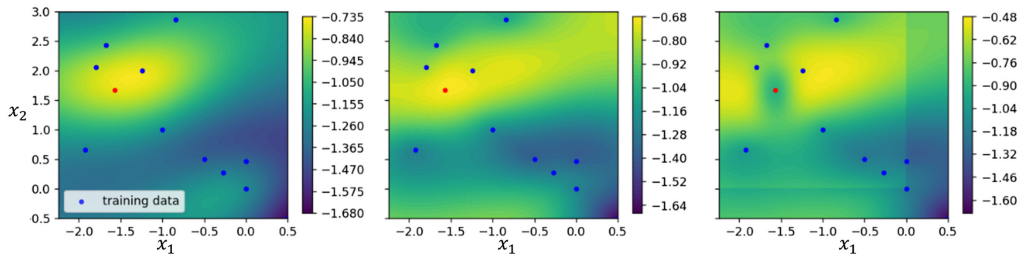


FIG. 14. Effect of penalization and favorization on a synthetic objective function over 2D control x_1 and x_2 . Left: GP mean prediction. Middle: bare acquisition function. Right: acquisition function after applying penalization and favorization. The color bar indicates the objective function value. The red marker denotes the pending point.

TABLE I. Composite objective configuration for MEBT tuning. The tolerance τ serves as the normalization factor, and weights w are normalized such that $\sum_i w_i = 1$.

Objective	v_{target}	τ	w
BPM_D1056: X	−0.43 mm	1.0 mm	0.087
BPM_D1056: Y	0.71 mm	1.0 mm	0.087
BPM_D1056: Phase	77.7°	1.0°	0.087
BPM_D1072: X	0.44 mm	1.0 mm	0.043
BPM_D1072: Y	0.40 mm	1.0 mm	0.043
BPM_D1072: Phase	−25.2°	1.0°	0.043
BPM_D1094: X	−0.24 mm	1.0 mm	0.022
BPM_D1094: Y	−0.19 mm	1.0 mm	0.022
BPM_D1094: Phase	−14.7°	1.0°	0.022
BCM_D1055: Current	>30.3 μA	1.5 μA	0.326
FC_D1102: Current	>24.2 μA	1.2 μA	0.217

- [1] J. Wei, H. Ao, S. Beher, N. Bultman, F. Casagrande, S. Cogan, C. Compton, J. Curtin, L. Dalesio, K. Davidson, M. Kelly, Y. Momozaki, G. Apollinari, K. Assamagan, M. Bai *et al.*, Advances of the FRIB project, *Int. J. Mod. Phys. E* **28**, 1930003 (2019).
- [2] P. N. Ostroumov *et al.*, Acceleration of uranium beam to record power of 10.4 kW and observation of new isotopes at Facility for Rare Isotope Beams, *Phys. Rev. Accel. Beams* **27**, 060101 (2024).
- [3] P. N. Ostroumov, F. Casagrande, K. Fukushima, K. Hwang, M. Ikegami, T. Kanemura, S.-H. Kim, S. Lidia, G. Machicoane, T. Maruta, D. Morris, A. S. Plastun, H. Ren, J. Wei, T. Xu, T. Zhang, Q. Zhao, and S. Zhao, FRIB commissioning, in *Proceedings of the 15th International Conference on Heavy Ion Accelerator Technology (HIAT'22)*, Darmstadt, Germany, 2022 (JACoW, Geneva, Switzerland, 2022), pp. 118–123.
- [4] P. N. Ostroumov, K. Fukushima, K. Hwang, T. Maruta, A. S. Plastun, J. Wei, T. Zhang, and Q. Zhao, Accelerator and beam physics challenges in support of FRIB experiments, *J. Phys. Conf. Ser.* **2687**, 052012 (2024).
- [5] P. Virtanen *et al.*, SciPy 1.0: Fundamental algorithms for scientific computing in Python, *Nat. Methods* **17**, 261 (2020).
- [6] K. Hwang, A. Plastun, K. Fukushima, P. N. Ostroumov, T. Maruta, T. Zhang, and Q. Zhao, Machine-learning-assisted beam tuning at FRIB, in *Proceedings of the 32nd Linear Accelerator Conference (LINAC'24)*, Chicago, IL, USA, 2024 (JACoW, Geneva, Switzerland, 2024).
- [7] R. Roussel *et al.*, Bayesian optimization algorithms for accelerator physics, *Phys. Rev. Accel. Beams* **27**, 084801 (2024).
- [8] J. Duris, D. Kennedy, A. Hanuka, J. Shtalenkova, A. Edelen, P. Baxevanis, A. Egger, T. Cope, M. McIntire, S. Ermon, and D. Ratner, Bayesian optimization of a free-electron laser, *Phys. Rev. Lett.* **124**, 124801 (2020).
- [9] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning* (MIT Press, Cambridge, MA, 2006).
- [10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. (MIT Press, Cambridge, MA, 2018).
- [11] J. Martinez-Marin, B. Mustapha, B. Blomberg, E. Letcher, G. Dunn, D. Stanton, and K. Bunnell, Reinforcement learning and Bayesian optimization for ion linac operations, in *Proceedings of the 15th International Conference on Heavy Ion Accelerator Technology (HIAT'22)*, Darmstadt, Germany, 2022 (JACoW, Geneva, Switzerland, 2022), pp. 153–156.
- [12] M. P. Deisenroth and C. E. Rasmussen, PILCO: A model-based and data-efficient approach to policy search, in *Proceedings of the International Conference on Machine Learning (ICML 2011)*, Bellevue, WA, USA (2011), pp. 465–472.
- [13] K. Chua, R. Calandra, R. McAllister, and S. Levine, Deep reinforcement learning in a handful of trials using probabilistic dynamics models, in *Proceedings of the 32nd International Conference Neural Information Processing System (NeurIPS'18)*, Montréal, Canada, 2018 (2018), pp. 4759–4770, <https://neurips.cc/virtual/2018/poster/11467>.
- [14] J. Kaiser *et al.*, Reinforcement learning-trained optimisers and Bayesian optimisation for online particle accelerator tuning, *Sci. Rep.* **14**, 15733 (2024).
- [15] J. Kaiser, C. Xu, A. Eichler, and A. S. García, Bridging the gap between machine learning and particle accelerator physics with high-speed, differentiable simulations, *Phys. Rev. Accel. Beams* **27**, 054601 (2024).
- [16] K. Hwang *et al.*, Prior-mean-assisted Bayesian optimization application on FRIB front-end tuning, [arXiv:2211.06400](https://arxiv.org/abs/2211.06400).
- [17] T. Boltz, J. L. Martinez, C. Xu, K. R. L. Baker, Z. Zhu, J. Morgan, R. Roussel, D. Ratner, B. Mustapha, and A. L. Edelen, Leveraging prior mean models for faster Bayesian optimization of particle accelerators, *Sci. Rep.* **15**, 12232 (2025).
- [18] I. Kostrikov, A. Nair, and S. Levine, Offline reinforcement learning with implicit Q-learning, in *Proceedings of the International Conference Learning Representations (ICLR'22)* (2022), <https://openreview.net/forum?id=68n2s9ZJWF8>.
- [19] R. Wang, D. P. Foster, and S. M. Kakade, Instabilities of offline RL with pre-trained neural representation, in *Proceedings of the 38th International Conference on Machine Learning (ICML'21)*, edited by M. Meila and T. Zhang, Proceedings of Machine Learning Research Vol. 139 (PMLR, 2021), pp. 10837–10847, <https://icml.cc/Conferences/2021>.
- [20] K. Hwang, J. Wan, T. Zhang, T. Maruta, Q. Zhao, A. Plastun, K. Fukushima, and P. N. Ostroumov, Machine-learning-assisted beam tuning at FRIB, in *Proceedings of the 5th North American Particle Accelerator Conference (NAPAC'25)*, Sacramento, CA, USA, 2025 (JACoW, Geneva, Switzerland, 2025).
- [21] J. Gonzalez, Z. Dai, P. Hennig, and N. Lawrence, Batch Bayesian optimization via local penalization, in

- Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS'16)*, Cadiz, Spain (2016), Vol. 51, pp. 648–657, <https://proceedings.mlr.press/v51/gonzalez16a.html>.
- [22] D. Ginsbourger, R. L. Riche, and L. Carraro, Kriging is well-suited to parallelize optimization, in *Computational Intelligence in Expensive Optimization Problems*, Adaptation Learning and Optimization Vol. 2 (Springer, Berlin, Heidelberg, 2010), pp. 131–162.
- [23] Y. Shen and C. Kingsford, Computationally efficient high-dimensional Bayesian optimization via variable selection, in *Proceedings of the 2nd International Conference on Automated Machine Learning (AutoML'23)*, Proceedings of Machine Learning Research Vol. 224 (PMLR, Potsdam, Germany, 2023), pp. 15/1–27, <https://2023.automl.cc/>.
- [24] D. Eriksson, M. Pearce, J. R. Gardner, R. D. Turner, and M. Poloczek, Scalable global optimization via local Bayesian optimization, in *Proceedings of the 33rd International Conference on Neural Information Processing System, (NeurIPS'19)*, Vancouver, Canada, 2019 (2019), https://papers.neurips.cc/paper_files/paper/2019/hash/6c990b7aca7bc7058f5e98ea909e924b-Abstract.html.
- [25] K. Hwang, A. S. Plastun, K. Fukushima, P. N. Ostroumov, T. Maruta, T. Zhang, Q. Zhao, and S. Cogan, Beam tuning at the FRIB front end using machine learning, in *Proceedings of the 13th International Particle Accelerator Conference (IPAC'22)*, Bangkok, Thailand, 2022 (JACoW Geneva, Switzerland, 2022), pp. 983–986.
- [26] P. Ostroumov, K. Hwang, A. Sheinker *et al.*, Online autonomous tuning of the FRIB accelerator using machine learning (2023); Project proposal narrative (unpublished).
- [27] P. N. Ostroumov, S. Cogan, K. Fukushima, S. Lidia, T. Maruta, A. S. Plastun, J. Wei, J. Wong, T. Yoshimoto, and Q. Zhao, Heavy ion beam acceleration in the first three cryomodules at the Facility for Rare Isotope Beams at Michigan State University, *Phys. Rev. Accel. Beams* **22**, 040101 (2019).
- [28] T. Zhang, K. Fukushima, T. Maruta, A. Plastun, K. Hwang, Q. Zhao, and P. Ostroumov, Physics application software for FRIB: From commissioning to operational excellence, in *Proceedings of the 20th International Conference on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'25)*, Chicago, IL, USA, 2025 (JACoW, Geneva, Switzerland, 2025), pp. 916–920.