

<https://doi.org/10.1038/s41524-024-01274-x>

Evolution-guided Bayesian optimization for constrained multi-objective optimization in self-driving labs

Check for updates

Andre K. Y. Low^{1,2,9}, Flore Mekki-Berrada^{3,9}, Abhishek Gupta⁴, Aleksandr Ostudin³, Jiaxun Xie³, Eleonore Vissol-Gaudin¹, Yee-Fun Lim^{2,5}, Qianxiao Li^{6,7}, Yew Soon Ong⁸, Saif A. Khan³✉ & Kedar Hippalgaonkar^{1,2}✉

The development of automated high-throughput experimental platforms has enabled fast sampling of high-dimensional decision spaces. To reach target properties efficiently, these platforms are increasingly paired with intelligent experimental design. However, current optimizers show limitations in maintaining sufficient exploration/exploitation balance for problems dealing with multiple conflicting objectives and complex constraints. Here, we devise an Evolution-Guided Bayesian Optimization (EGBO) algorithm that integrates selection pressure in parallel with a q-Noisy Expected Hypervolume Improvement (qNEHVI) optimizer; this not only solves for the Pareto Front (PF) efficiently but also achieves better coverage of the PF while limiting sampling in the infeasible space. The algorithm is developed together with a custom self-driving lab for seed-mediated silver nanoparticle synthesis, targeting 3 objectives (1) optical properties, (2) fast reaction, and (3) minimal seed usage alongside complex constraints. We demonstrate that, with appropriate constraint handling, EGBO performance improves upon state-of-the-art qNEHVI. Furthermore, across various synthetic multi-objective problems, EGBO shows significant hypervolume improvement, revealing the synergy between selection pressure and the qNEHVI optimizer. We also demonstrate EGBO's good coverage of the PF as well as comparatively better ability to propose feasible solutions. We thus propose EGBO as a general framework for efficiently solving constrained multi-objective problems in high-throughput experimentation platforms.

With the expansion of automation in laboratories and the development of new machine learning tools to guide optimization processes, recent years have seen a boom of materials acceleration platforms (MAPs) in material science^{1–3}. High-throughput experimental (HTE) platforms now enable rapid synthesis procedures⁴, with inline material characterization^{5,6} and parallelization of workflows to facilitate batch sampling. Many examples of closed-loop optimization on HTE platforms can be found in the literature.

For example, Wagner et al.⁷ developed the AMANDA platform to produce and characterize solution-processed thin-film devices for organic photovoltaic applications. Other MAPs have been implemented for the discovery of metal halide perovskite alloys⁸, polymer composites for photovoltaics^{9,10} and metal nanoparticles^{11–13}.

Correspondingly, many scientific and/or engineering challenges in materials science and chemistry require satisfying multiple objectives and

¹School of Materials Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore. ²Institute of Materials Research and Engineering (IMRE), Agency for Science, Technology and Research (A*STAR), 2 Fusionopolis Way, Innovis #08-03, Singapore 138634, Republic of Singapore. ³Department of Chemical and Biomolecular Engineering, National University of Singapore, Singapore 117585, Singapore. ⁴School of Mechanical Sciences, Indian Institute of Technology Goa, Goa 403401, India. ⁵Institute of Sustainability for Chemicals, Energy and Environment (ISCE2), Agency for Science, Technology and Research (A*STAR), 1 Pesek Road, Singapore 627833, Republic of Singapore. ⁶Department of Mathematics, National University of Singapore, Singapore 119077, Singapore. ⁷Institute for Functional Intelligent Materials, National University of Singapore, Singapore 117544, Singapore. ⁸School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore. ⁹These authors contributed equally: Andre K. Y. Low, Flore Mekki-Berrada.

✉ e-mail: saifkhan@nus.edu.sg; kedar@ntu.edu.sg

constraints, resulting in increased interest in framing solutions for constrained multi-objective optimization problems (cMOOPs)^{5,14}. A few examples of cMOOPs are reported in material science literature. For instance, Erps et al.¹⁵ demonstrated the optimization of 3D printing materials for maximal compression modulus, compression strength and toughness, while Cao et al.¹⁶ proposed liquid formulated products that minimize viscosity, turbidity and price and used classification to differentiate stable formulations.

While HTE platforms enable fast and more extensive sampling of these cMOOPs, evaluation budgets are often still limited to 10^2 – 10^3 samples due to operating and chemical costs, finite available quantities of reactants and time considerations. However, multi-objective optimization requires intensive sampling, especially as the dimensionality of the decision space and the number of conflicting objectives increases¹⁷. Moreover, the objective landscape can be complex with non-linear input-output relationships furthering the need for more samples. The complexity of cMOOPs notably arises when dealing with conflicting objectives, as one needs to find not only one global optimum but an ensemble of the most optimal across all objectives, also known as the Pareto Front (PF). Discovering the PF is often used to decide how to operate the HTE platform to produce the best material. Alternatively, knowledge extraction can be performed *a posteriori* to understand the relationship between each objective at the PF¹⁸. However, the accuracy of knowledge extraction strongly depends on the sampling resolution at the PF. Therefore, special attention should be given to the PF exploration during the optimization process.

Another level of difficulty is reached with constraints that make some of the trade-off solutions potentially impossible to evaluate¹⁹. To learn and handle the constraint functions, the optimizer will inevitably suggest infeasible candidates which might damage the HTE platform itself, a critical consideration for scientists. Therefore, an ideal optimization algorithm must not only (1) efficiently exploit towards the PF location, but also (2) uniformly explore the PF, while (3) avoiding infeasible regions near the PF, as illustrated in Fig. 1 (top left). Moreover, when the objectives are experimentally measured, the optimizer would also need to deal with fidelity/noise concerns in synthesis and characterization, which complicates the objective landscape.

Amongst recently proposed optimizers, q-Noise Expected Hypervolume Improvement based Bayesian Optimization (qNEHVI-BO)²⁰ is a state-of-the-art Bayesian optimization approach that caters to constraint handling, batch sampling and noisy evaluations. It was successfully implemented in a self-driving lab for multi-objective optimization of thin films by MacLeod et al.²¹. However, Low et al.²² demonstrated that the optimization procedure may lead to over-exploration, inevitably resulting in sampling wastage. In this paper, we address this concern, which is particularly acute for cMOOPs, by introducing selection pressure from an evolutionary algorithm (EA) as a parallel optimization mechanism, creating a hybrid framework we hereby name as Evolution-Guided Bayesian Optimization (EGBO). For the rest of this section, we first describe the self-driving platform and then the proposed optimizer EGBO.

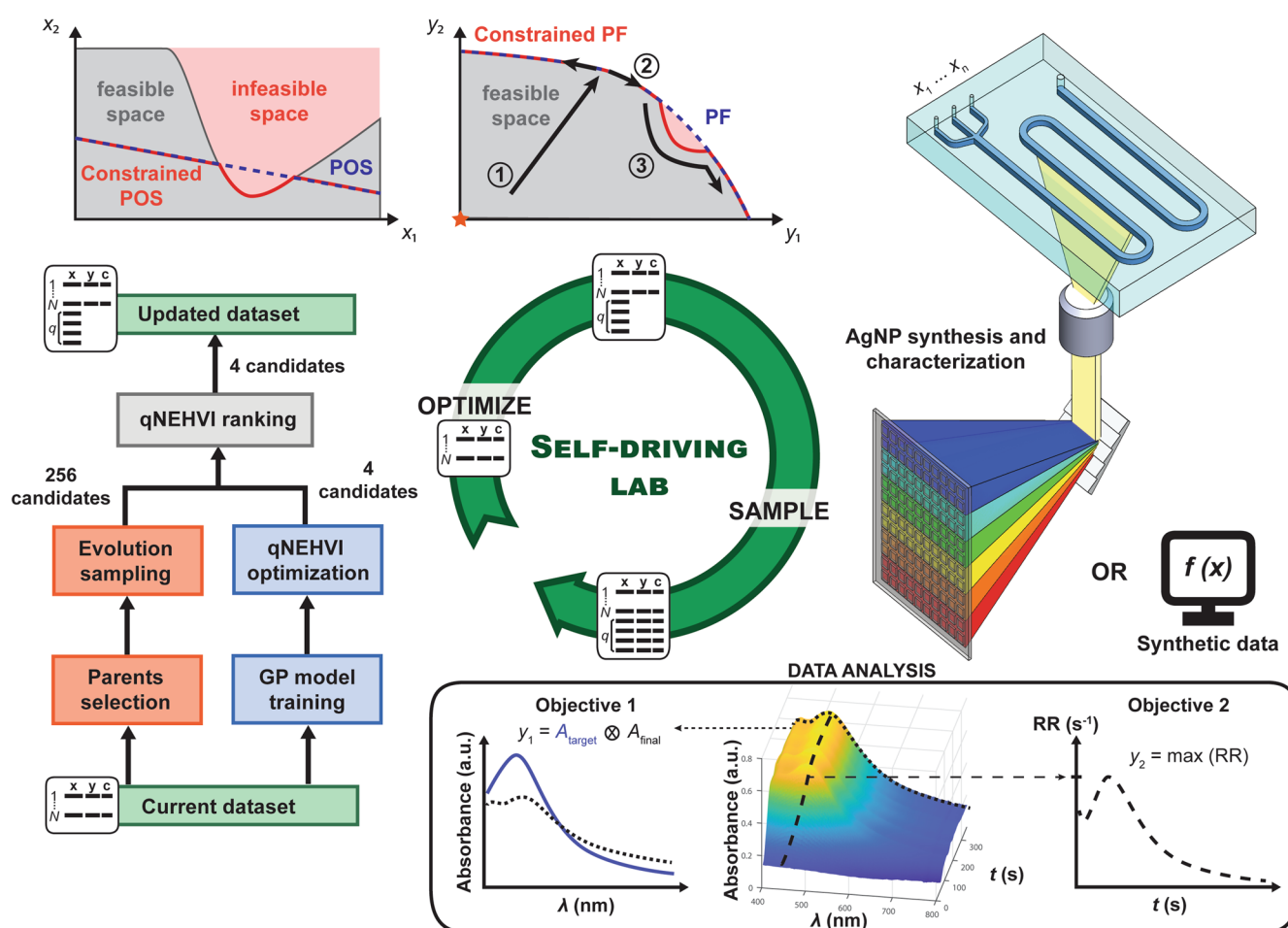


Fig. 1 | Closed-loop workflow of the EGBO-guided HTE platform. The illustration on the top left of the figure shows the unconstrained (blue dashed line) and constrained (red solid line) Pareto Optimal Set (POS) in a constrained decision space, and its respective projection as the Pareto Front (PF) in the objective space. EGBO algorithm (left) combines an evolutionary algorithm (orange) and a qNEHVI-BO

(blue) working in parallel to suggest 4 optimal candidates for the cMOOP. The optimizer's goal is to (1) efficiently reach the PF, (2) uniformly explore the PF and (3) avoid infeasible domains near the PF (top left). The candidates are then sampled on a hyperspectral HTE platform optimizing AgNP synthesis (right) and further analysis is done to derive the objective values (bottom) before a new EGBO iteration.

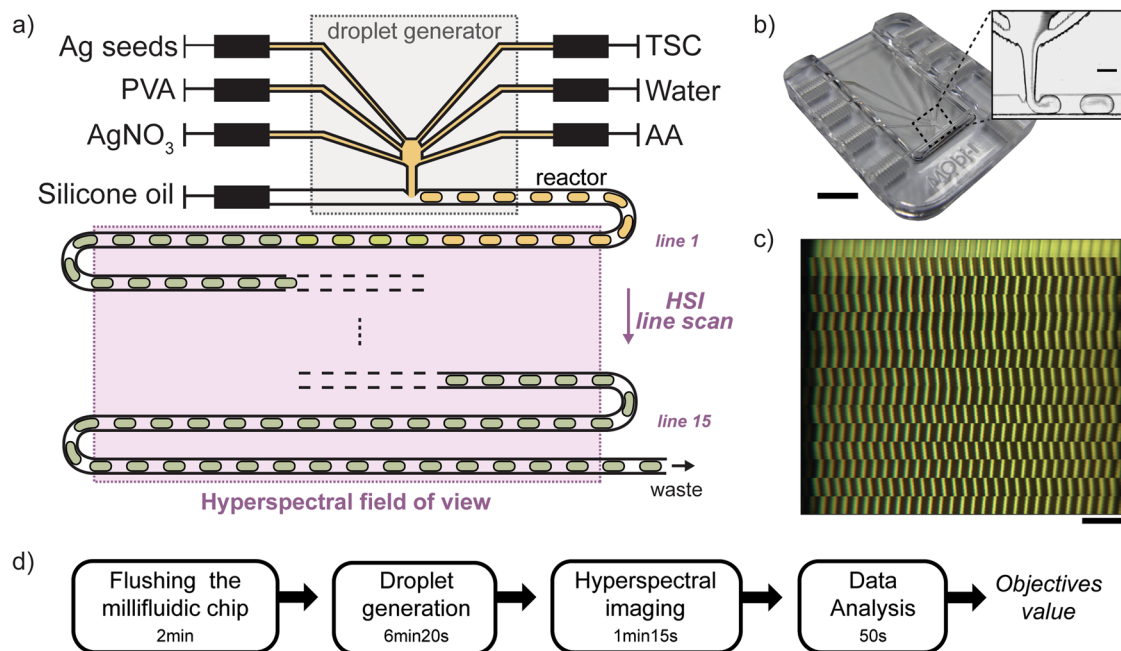


Fig. 2 | Overview of the AgNP HTE platform. **a** Detailed schematic of the microfluidic platform. Droplet generation is performed in a **(b)** customized chip. Reacting droplets are then released in a 1.5 m long reactor tube (PFA, 1 mm ID) where the hyperspectral image **(c)** is taken. **d** Workflow of the HTE platform.

Self-driving silver nanoparticle synthesis

Silver nanoparticles (AgNPs) have received particular attention due to their many applications, notably in chemical sensing²³, photovoltaics²⁴, electronics²⁵ and biomedicine²⁶. AgNPs can take various sizes (1–100 nm) and shapes (spheres, rods, prisms, cube, etc.), resulting in a wide range of spectral signatures due to localized surface plasmon resonance²⁷. Here, we propose to optimize a AgNP synthesis method based on seed-mediated growth. Our optimization goal is threefold: (1) target a desired spectral signature, (2) maximize the reaction rate and (3) minimize the usage of the costliest reactant, here silver seeds. For this purpose, we develop a closed loop machine learning driven HTE platform consisting of a microfluidic platform that produces droplets containing the reacting chemicals and a line-scan hyperspectral imaging system capturing the droplets UV/Vis spectral image in situ²⁸ as seen in Fig. 2.

The chemical composition of the droplets is controlled by the flowrate Q of each reactant: silver seeds (10 nm, 0.02 mg mL⁻¹ in aqueous buffer stabilised with sodium citrate), silver nitrate (AgNO₃, 15 mM), ascorbic acid (AA, 10 mM), trisodium citrate (TSC, 15 mM) and polyvinyl alcohol (PVA, 5 wt%). These flowrates define the 5 decision variables of the cMOOP. DI water flowrate is adjusted to maintain a constant total aqueous flowrate of 120 μL min⁻¹, to get stable droplets that are long enough to get accurate absorbance measurements²⁸.

The objectives and constraints of the cMOOP can be written as follows:

$$\max \begin{cases} y_1 = \min(3, -\delta(A_{\max}(t_{\text{end}})) \times \log_{10}(1 - \text{cosim}(A(\lambda, t_{\text{end}}))))/3 \\ y_2 = \max_t \left(\frac{A_{\max}(t)}{t} \right) \\ y_3 = 1 - Q_{\text{seed}}/120 \end{cases} \quad (1)$$

$$\text{st} \begin{cases} Q_{\text{TSC}}, Q_{\text{AgNO}_3}, Q_{\text{PVA}}, Q_{\text{seed}}, Q_{\text{AA}} \in [0.6 : 24] \mu\text{L min}^{-1} \\ c_1 = 0.3 - \frac{Q_{\text{AgNO}_3}}{Q_{\text{AA}}} \leq 0 \\ c_2 = 2 - \frac{Q_{\text{AgNO}_3}}{Q_{\text{AA}}} - \frac{Q_{\text{seed}}}{Q_{\text{AgNO}_3}} \leq 0 \end{cases} \quad (2)$$

The first objective y_1 quantifies the closeness to the targeted spectral signature, defined as the cosine similarity between the final absorbance spectrum of the sample $A(\lambda, t_{\text{end}})$ and of the target $A_{\text{target}}(\lambda, t_{\text{end}})$ to evaluate how similar their normalized spectra are. We use logarithmic scale to rebalance the distribution and favor sampling of cosine similarity values from 0.9 to 0.999. We further multiply it by a gating function δ defined as:

$$\delta(A) = \begin{cases} 1 & \text{if } A \in [0.3 : 1.2] \\ 0 & \text{if } A < 0.3 \text{ or } A > 1.2 \end{cases} \quad (3)$$

with this operation, the objective value y_1 is set to zero if the final absorbance amplitude $A_{\max}(t_{\text{end}})$ does not fall in the operation range of the hyperspectral sensor, i.e., when $A_{\max}(t_{\text{end}}) \in [0.3 : 1.2]$. This is to ensure precise measurements, as low amplitude spectrum would lead to a noisy measurement of the cosine similarity, while a saturating spectrum would make its estimation inaccurate. Finally, the objective value is capped to 3 as the sensor sensitivity does not allow sufficient precision when cosine similarity goes above 0.999.

The second objective y_2 is to maximize the reaction rate (RR). The time evolution of the absorbance peak is extracted from the absorbance map (Fig. 1 bottom) to estimate the maximal reaction rate, here defined as the ratio between $A_{\max}(t)$, the maximal absorbance amplitude over the wavelengths and t , the residence time of the reacting droplet. The best reaction rate of a condition is defined as the maximal reaction rate obtained over the reaction time. It indicates what would be the highest reaction rate achievable if the reactor was cut to its optimal length. Finally, the third objective y_3 is about minimizing the consumption of silver seeds, the costliest reactant for this synthesis method. This is done by minimizing the flowrate ratio of silver seeds against the total aqueous phase. We note that objectives are normalized to [0,1] within the optimization workflow, while values reported in the following figures are unprocessed.

The cMOOP is subject to box constraints for the 5 decision variables. Two additional non-linear engineering constraints are implemented to prevent secondary nucleation of AgNPs, which could cause irreversible clogging of the microfluidic droplet generator. Based on previous work²⁸ and preliminary reproducibility tests on the HTE platform, we identified two regions that could lead to clogging or fluctuations of the absorbance maps. The constraints c_1 and c_2 are defined as an overconfident estimate of the

boundary planes of these two regions. They are consistent with our expertise knowledge, as secondary nucleation could either come from an excess of the reducing agent AA with respect to the quantity of silver atoms, or from a deficiency of silver seeds for silver atoms to grow on.

Evolution-guided Bayesian optimization

Bayesian Optimization (BO) is a popular technique for optimization of expensive functions. Most commonly, a Gaussian Process (GP) prior is defined over the objective and constraint functions, expressed as $f(\mathbf{x}) \sim \text{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ where $\mu(\mathbf{x})$ and $k(\mathbf{x}, \mathbf{x}')$ represent the mean and covariance respectively²⁹. Taking observations $D = \{(x_i, y_i)\}_{i=1}^N$ where N is the number of data points, the GP is then updated to a posterior distribution of form $f(\mathbf{x}|D) \sim \text{GP}(\mu_{\text{post}}(\mathbf{x}), k_{\text{post}}(\mathbf{x}, \mathbf{x}'))$ with updated mean and covariance. Thereafter, an appropriate acquisition function $\alpha(\mathbf{x})$ is defined, and then optimized to provide $\mathbf{x}_{\text{new}} = \text{argmax}_{\mathbf{x}} \alpha(\mathbf{x})$ as the candidate(s) to sample for maximal gain.

BO can be extended to multi-objective by considering scalarization of objectives into a single metric such as hypervolume (HV), which is defined as the Euclidean measure of the space covered by non-dominated PF solutions and bounded by a reference point³⁰. Daulton et al.^{20,31} proposed an acquisition function Noisy Expected Hypervolume Improvement $\alpha_{\text{NEHVI}}(\mathbf{x}) = \int \alpha_{\text{EHVI}}(\mathbf{x}|D_n) p(f|D_n) df$ by considering the expected hypervolume improvement (HVI) with uncertainty of the posterior distribution conditioned on noisy observations D_n . The analytically tractable form is derived by Monte-Carlo (MC) integration to give $\hat{\alpha}_{\text{NEHVI}}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \text{HVI}(\tilde{f}(\mathbf{x})|P_i)$ where $\tilde{f}(\mathbf{x})$ are samples taken from the posterior and P_i is the PF of the observations. In BoTorch³², this is efficiently optimized by performing gradient-based optimization from starting samples to attain maximum $\hat{\alpha}_{\text{NEHVI}}(\mathbf{x})$, implemented as the `optimize_acqf` function. Additionally, constraint handling is also integrated via sigmoid weighting the acquisition value on constraint predictions from the GP. Thereafter, we refer to the overall optimization workflow as q-Noise Expected Hypervolume Improvement based Bayesian Optimization (qNEHVI-BO).

HV improvement as a strategy has been shown to be robust in multi-objective optimization compared to decomposition or scalarization based methods which may require more careful hyperparameter tuning³³. Furthermore, qNEHVI-BO has been demonstrated to outperform other well-performing algorithms including Thompson sampling efficient multi-objective optimization (TSEMO)³⁴ and diversity-guided efficient multi-objective optimization (DGEMO)³⁵. We further note that these algorithms do not include native constraint handling as well, making their implementation to cMOOPs less suitable.

The performance of qNEHVI-BO remains limited when considering cMOOPs. Indeed, studies from Low et al.²² on high-dimensional problems show that qNEHVI-BO can efficiently reach the PF but will fail to explore it uniformly. Work by Rasmussen et al.³⁶ also highlighted a key point in understanding BO performance: a poorly calibrated surrogate model may be inconsequential as optimization is driven by the search strategy or with enough random sampling. In contrast, evolutionary algorithms are better at exploring diverse points along the PF but require more evaluations to reach the PF. In both cases, sampling wastage will occur, as sub-optimal or infeasible candidates are suggested. Sampling wastage will even increase at larger sampling batches.

Our proposed EGBO aims to improve upon qNEHVI-BO by integrating an EA into the workflow for maximizing acquisition function $\hat{\alpha}_{\text{NEHVI}}(\mathbf{x})$. EAs are nature-inspired algorithms which rely on the idea of natural selection and evolution to optimize³⁷. The addition of an EA provides selection pressure to promote higher exploitation towards the PF, better exploration along the PF and better handle constraints to minimize sampling wastage, in line with the goals as illustrated in Fig. 1 top left.

The EGBO framework is illustrated in Fig. 1, left. For the work shown here, we used unified non-dominated sorting genetic algorithm III (U-NSGA-III)³⁸ as the EA component in the EGBO framework, implemented through `pymoo`³⁹. Once the dataset and surrogate models are updated with newly sampled objective and constraint values, two parallel processes happen. On one side, q new samples (where q is the defined batch size) are

proposed by `optimize_acqf` (similar to the original qNEHVI-BO) to give us the BO candidates.

In parallel, and independently, we perform one iteration of the evolution process in U-NSGA-III to form the EA candidates. Firstly, we filter for non-dominated solutions to undergo selection of parents, which we do with the `is_non_dominated` method from BoTorch. In U-NSGA-II the selection is done using TournamentSelection, which hierarchically selects for parents that belong to different niches/reference vector first, before considering non-dominance ranking and closeness to the reference vector. Given that we consider the EA as a secondary optimization mechanism to ensure minimum regret/wastage by biasing exploitation, we take strictly non-dominated parents to form candidates, since otherwise it is possible to randomly get a pair of dominated parents if they are reasonably diverse, getting poorer exploitation.

Thereafter, the reproduction step uses a crossover and mutation step to form the children set. The crossover operator enables for recombination of genes between parents for children, and the mutation operator introduces variations in those genes as a form of injecting stochasticity. In `pymoo`, this is coded by using a `.tell` method to update the current population, and then `.ask` which implements the actual sequence of selection, crossover and mutation. Afterwards, the two sets of new candidates are then combined to greedily maximize for $\hat{\alpha}_{\text{NEHVI}}(\mathbf{x})$, taking the q best candidates for evaluation.

In the following, we evaluate EGBO's performance on various cMOOPs. Firstly, in a real closed-loop experimental self-driving lab (see Fig. 1) and secondly, on synthetic problems of various complexities. We successfully demonstrate that EGBO outperforms qNEHVI-BO in terms of optimization efficiency and constraint handling, showing that selection pressure helps EGBO to bypass infeasible regions located near the PF, even in high dimensional decision spaces with very limited feasible space. Finally, we perform studies to explore how EGBO could be adapted to further improve constraint handling and comment on its wide applicability in the physical and chemical sciences.

Results and discussion

For each problem, we first seek a qualitative understanding of the optimization procedure. For synthetic problems, the optimization is run 10 times to consider results of stochasticity, and a probability density map is plotted to visualize the sampling distribution in the objective space, as proposed by Low et al.²². This allows us to report the mean and confidence intervals of the different metrics, and additionally allows us to show the distribution of sampling across all runs. For AgNP however, we report 2 parallel campaigns, one for each optimizer as is reported in other self-driving labs^{15,16}. We then follow the optimization trajectory in the objective space in real time and track the location of new sampling points at each iteration.

For a more quantitative evaluation of the optimization process, we used three performance metrics:

1. Hypervolume (HV) – directly shows the convergence ability and speed in reaching the PF. For synthetic problems, we consider the log function against the true PF: $\log \Delta \text{HV} = \log_{10}(\text{HV}_{\text{true}} - \text{HV}_{\text{current}})$.
2. Non-uniformity (NU) – quantifies the PF exploration by measuring the non-uniformity of the non-dominated solutions along the PF. We define the non-uniformity metric $\text{NU} = \frac{\sigma(d)}{\mu(d)}$, where d is the distance to the nearest neighbour, $\mu(d)$ is the averaged distance and $\sigma(d)$ the distance variance, considering only the non-dominated points.
3. Constraint handling performance – quantifies sampling wastage during the optimization through:
 - a. Infeasibility (%infeas) – the cumulative fraction of infeasible solutions suggested by the optimizer since the initialization until the considered iteration.
 - b. Constraint violation (CV) – the constraint violation is evaluated at each iteration by: $\text{CV}_i = \sum_{j \in \text{batch}} \max(0, c_i(x_j))$, where x_j are the batch of candidates suggested for the considered iteration.

An ideal optimizer would reach high HV values in a limited number of sampling iterations, with a low non-uniformity and a low percentage of

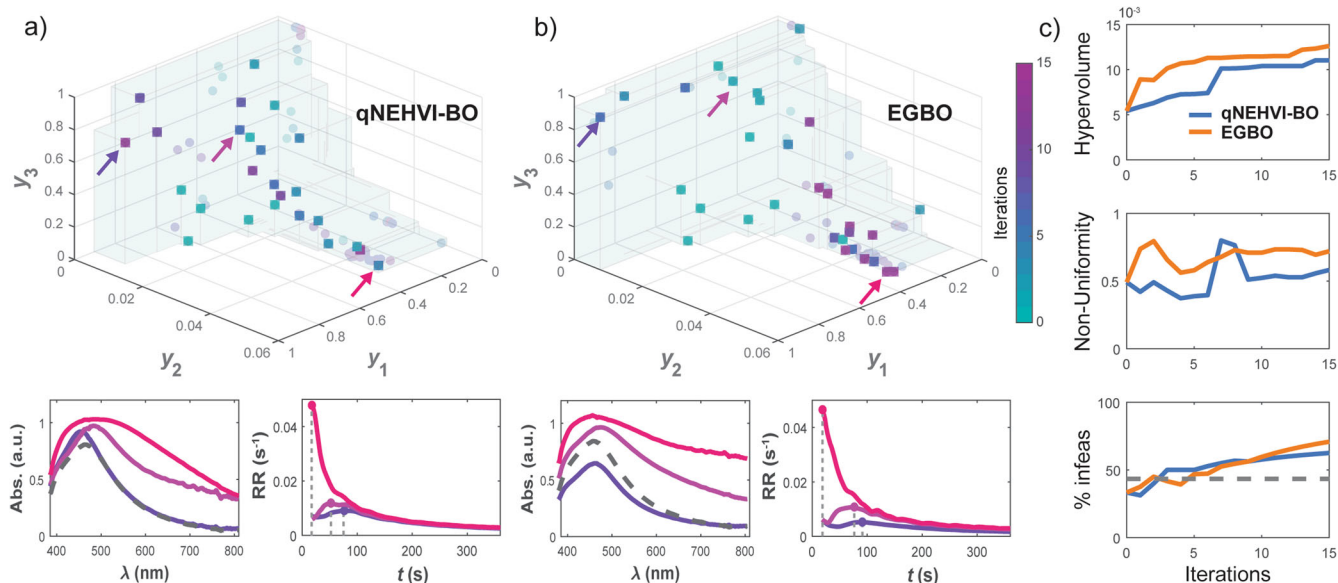


Fig. 3 | qNEHVI-BO and EGBO performance on the self-driving AgNP synthesis platform. Optimization trajectory of (a) qNEHVI-BO and (b) EGBO from initialization (blue) to the 15th iteration (purple). The final HV is highlighted in cyan. Non-dominated points are represented by squares. For each optimizer, three non-dominated points are chosen (purple/pink arrows) to highlight the diversity of final

absorbance spectra (coloured solid lines) compared to the target spectrum (grey dashed line) and time evolution of the reaction rate (coloured solid lines) obtained over the PF. c Evolution of the performance metrics for qNEHVI-BO (blue) and EGBO (orange solid) with the number of iterations. The percentage of infeasible points in the whole decision space is represented as a grey dashed line.

infeasible samples. Both HV and NU metrics are equally important, as high HV values indicate that the extrema of the PF have been sampled while low NU values show a uniform spread of the sampling points on the PF. The uniformity of non-dominated solutions is particularly important when *a posteriori* knowledge needs to be extracted on the PF¹⁸.

Self-driving lab for AgNP experimental campaign

The performance of two optimizers – (1) qNEHVI-BO and (2) EGBO employing a U-NSGA-III variant - are simultaneously tested on the AgNP cMOOP (detailed in Self-driving silver nanoparticle synthesis section). For each iteration, if the flowrates suggested by the optimizers are infeasible according to constraint equations c_1 and c_2 , they are repaired first before evaluation (details in Methods section).

Figure 3a, b (top) illustrate the optimization trajectories of qNEHVI-BO and EGBO in the objective space. We observe a clear trade-off between y_1 and y_2 as seen by the concave shape of the PF. While both optimizers converged to similar maximum objective values (Supplementary Fig. 1.1), EGBO maintained a superior HV score throughout, and was able to sample the highest y_1 value on the 1st iteration (highlighted with the dark purple arrow in Fig. 3b top). The corresponding shape of the final absorbance spectrum (dark purple line) is closer to the spectral shape of the target, while the best qNEHVI-BO sample along y_1 , obtained only on the 7th iteration, has a significant λ -shift, resulting in a lower cosine similarity. This explains the delay in HV improvement observed in Fig. 3c (top) between the two algorithms.

To elucidate further on the results, Fig. 3a, b (bottom) report the final absorbance spectra and time evolution of the reaction rate for 3 non-dominated points. The best y_2 candidates (pink arrows and curves) exhibit widely spread absorbance spectra, suggesting AgNP agglomeration in the droplets, whereas high y_1 candidates have a bell-shaped evolution of reaction rate (purple arrows and curves), potentially caused by an underlying slow growth process. Thus, we observe a large variety of behaviours among the non-dominated points.

EGBO exhibits a reduced number of solutions where y_1 equals zero, indicating that the gating function was better handled. However, when analysing the feasibility of the candidates before repair with

respect to constraint functions c_1 and c_2 , both algorithms show a high percentage of infeasible samples (see Fig. 2c, bottom), going above the percentage of infeasible points in the decision space (grey dashed line). Most notably, EGBO proposed more infeasible solutions compared to qNEHVI-BO despite a better HV metric and leads to concentrated sampling in the high y_2 region. Consequently, EGBO finds more non-dominated points and gets a wider PF by picking more extreme objective values, qNEHVI-BO showcases a more uniform sampling on the PF (see the squares distribution in Fig. 3a, b, top). This is supported by Fig. 3c (middle), where EGBO gets a higher HV of roughly 14% compared to qNEHVI-BO and a lower NU of 24% compared to EGBO. This can be explained by how the post-repair affects the optimization in a problem where the constraint functions intersects with the PF, which we discuss further below.

To summarize, we showcased the implementation of EGBO in a self-driving lab for the optimization of AgNP synthesis, a real-world cMOOP with 3 objectives and 2 complex constraints in a 5-dimensional decision space. Here, a single run of 15 iterations with batch size of 4 was performed for each optimizer (EGBO and qNEHVI-BO respectively). The obtained results suggest that EGBO reaches the PF faster than qNEHVI-BO, with an early sampling of extrema solutions, leading to a wider PF, although the spread of non-dominated solutions and the percentage of infeasible points were comparable for both optimizers. However, a single run can be sensitive to the choice of initial candidates and random noise. To demonstrate EGBO's improved performance, statistical analysis on multiple runs is required across problems of various complexities. For this purpose, we propose to consider synthetic problems below.

Synthetic studies

To complement the observations reported from the AgNP synthesis cMOOP in a more rigorous form, we further investigate in detail the performance of EGBO on various synthetic datasets, starting with MW7 from the MW test suite¹⁹. The complexity of the synthetic problem MW7 is similar to the AgNP synthesis cMOOP described earlier in terms of multiple objectives paired with complex constraints that intersect the PF. The

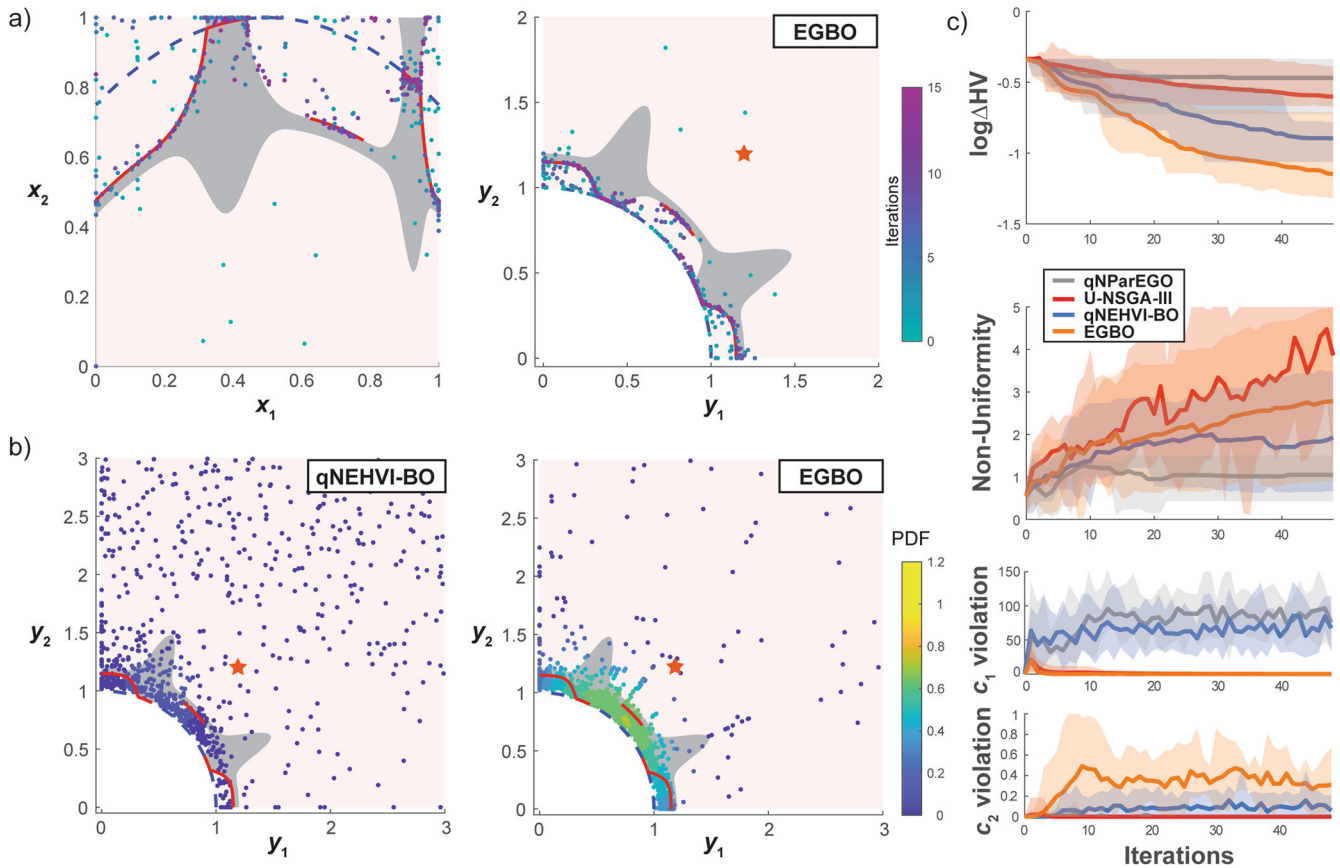


Fig. 4 | Performance of qNEHVI-BO and EGBO on MW7 problem.

a Optimization trajectory in the decision (left) and objective (right) spaces, obtained for $n = 2$ variables after 3 runs of 30 iterations with EGBO. **b** Probability density map in the objective space, obtained for $n = 8$ variables after 10 runs of 48 iterations with qNEHVI-BO (left) and EGBO (right). The feasible region is highlighted in grey and the infeasible one in red. Low probably density values (blue) indicate sparse sampling in the objective space, while high values (yellow) highlight regions that are

intensively sampled. The true unconstrained PF is shown in blue, and the true constrained PF in red. **c** Evolution of the performance metrics ($\log \Delta HV$, NU and constraint violations) with the number of iterations for qNEHVI-BO (blue) and EGBO (orange). The HV reference location is indicated by an orange star. We report the mean value of the metric (solid line) as well as its 95% confidence interval across 10 runs (shaded region).

equations for MW7 are stated below:

$$\min \begin{cases} f_1(\mathbf{x}) = g_3 x_1 \\ f_2(\mathbf{x}) = g_3 \sqrt{1 - (f_1/g_3)^2} \end{cases} \quad (4)$$

$$st \begin{cases} c_1(\mathbf{x}) = (1.2 + 0.4 \sin(4l))^{16} - f_1^2 - f_2^2 \geq 0 \\ c_2(\mathbf{x}) = (1.15 + 0.2 \sin(4l))^8 - f_1^2 - f_2^2 \leq 0 \end{cases} \quad (5)$$

$$g_3 = 1 + \sum_{i=2}^n 2(x_i - (x_{i-1} - 0.5))^2 - 1)^2$$

$$l = \arctan(f_2/f_1)$$

In contrast to the AgNP experimental campaign, we allowed infeasible solutions to be evaluated rather than rejected, and no repair was performed, since they do not cause the optimization to stop. Thus, the optimizer learns the objective and constraint values in the infeasible space. This allows us to showcase the superior ability of EGBO to learn and handle constraints, as reported by the constraint violations. We evaluate the optimization performance of qNEHVI-BO and EGBO, and include pure U-NSGA-III and qNParEGO²⁰ for comparison. qNParEGO is an acquisition function which relies on scalarized vectors to adapt single objective expected improvement and is implemented in BoTorch alongside qNEHVI-BO utilizing the same acquisition function optimization and constraint handling mechanisms.

Each computational experiment is run several times to extract statistical trends.

Figure 4a illustrates the simple case of MW7 at only $n = 2$ variables with EGBO after 3 runs consisting of 30 iterations. The grey feasible regions in the decision and objective spaces are both a single continuous region made of two large islands connected by a narrow bridge with a disjointed constrained Pareto Optimal Set (POS) for input space and PF in output space (shown as red solid lines). The first constraint c_1 defines the upper boundary of the feasible grey region, while the second constraint c_2 is the one that restrains the PF. Therefore, sampling points in the top right corner of the objective space corresponds to c_1 violations, while sampling points between the unconstrained PF (blue dashed line) and constrained PF (red solid line) are linked to c_2 violations (Supplementary Fig. 2.1). We see in Fig. 4a that EGBO seems to properly explore the whole constrained POS and PF, despite a few infeasible samples.

We then performed MW7 optimization at $n = 8$ variables, following a similar or higher dimensionality compared to other examples in the literature^{16,21,40,41}. Figure 4b shows the probability maps obtained for qNEHVI-BO and EGBO after 10 runs. EGBO is observed to be superior to qNEHVI-BO in exploiting the constrained PF with a higher probability density near the PF and 28% lower $\log \Delta HV$ reported in Fig. 4c, meaning that more solutions are closer to the true PF. However, this was done to the detriment of the sampling uniformity on the PF, with a 44% higher NU metric compared to qNEHVI-BO. The probability density shows that the central part of the PF is more sampled than the sides which can be explained

by the bias introduced by the HV reference point²², as the PF extrema are associated with lower HV improvement.

While qNEHVI-BO proposed many infeasible candidates far away from the PF (Fig. 4b, top right corner), EGBO does not waste as many samples in that region beyond the HV reference point (orange star) where HV improvement is null. We noted that EGBO reaches its highest probability density near the region where the PF is discontinuous, since many data points are required to learn a disjointed POS. The probability maps suggest that qNEHVI-BO did not handle constraints well, whereas EGBO managed to minimize sampling far from the feasible space. This is supported by the evolution of constraints violation over the iterations (Fig. 4c, bottom). EGBO c_1 violation converges towards zero within the first few evaluations.

Indeed, as selection pressure preserves and transfers parent feasibility to the children via crossover, EGBO candidates are more likely to evolve within the feasible region towards the PF. On the contrary, EGBO c_2 violation values consistently remain above zero. This can be explained by the perturbation introduced by mutation when proposing new candidates from parents that already are near the constrained PF. Since the c_2 violation values are two orders of magnitude smaller than the c_1 violation values (Supplementary Fig. 2.1), EA children near the PF easily cross the c_2 boundary. In contrast, pure qNEHVI relies solely on the constraint values predicted by the GP surrogate model. This presents a similar scenario to our observations on the AgNP experimental campaign, where it was noted that EGBO had a worse fraction of infeasible solutions despite a better HV metric, since qNEHVI-BO did not evaluate as many solutions near the POS. This phenomenon of sampling bias is also known as Simpson's Paradox⁴².

Figure 4c also reports the optimization performance of qNParEGO and pure U-NSGA-III. We see that both qNEHVI-BO and EGBO clearly outperform qNParEGO and U-NSGA-III in terms of HV improvement and sampling wastage. The sampling non-uniformity of qNParEGO is the lowest and, unsurprisingly, that of U-NSGA-III the highest since it prioritizes diversity of the parents first before non-dominance. Interestingly, qNEHVI-BO has a NU metric close to qNParEGO and EGBO lies between qNEHVI-BO and U-NSGA-III. Overall, EGBO balances a good spread of the solutions across the PF while maintaining superior HV improvement.

Finally, we selected a few other synthetic problems from the MW test suite¹⁹ and ZDT test suites⁴³ which provide a range of scalable 2 and 3-objective constrained problems. They are representative of the variety of complex problems that can be found in the real world. We report our results for these cMOOPs in Supplementary Discussion subsection 2.2, where we demonstrate that EGBO outperforms qNEHVI-BO in terms of HV across all the selected problems.

We therefore elucidate that having selection pressure in the optimization process, is not only superior in reaching the PF, but also more effective for handling objective constraints. In computing $\hat{\alpha}_{NEHVI}$, constraint handling is already integrated by predicting constraint values from the surrogate model and weighting the acquisition value accordingly as explained by Daulton et al.²⁰. However, the TournamentSelection in U-NSGA-III favours solutions far from each other, encouraging exploration of underexplored regions for enhanced PF coverage, while also inducing a strong sampling bias exploiting feasible regions of the decision space, as feasible solutions are always selected over infeasible ones.

Notably, this acts as a secondary means of constraint handling, which means that the candidates are more likely to be feasible even before being evaluated by the surrogate model.

In practice, this could be useful in real-world experimentation scenarios, where infeasible solutions cannot be evaluated at all. Common implementations of constraint handling often involve training a separate model to classify feasibility, which could present a challenge especially during earlier iterations with limited data, and also in the case where the constraint boundary is very complex.

Handling input and output constraints

In experimental campaigns, constraints can represent physical or engineering limitations that cannot be violated and thus cannot be evaluated. This is very different from computational problems in which infeasible solutions can be evaluated⁴⁴. We define two types of constraints: output constraints which can only be evaluated alongside the objective functions and input constraints which can be computed cheaply before any objective evaluation. This is different from previous literature which categorizes feasibility into defined constraints (being possible to compute) and undefined constraints (when computing the constraint crashes the simulation)^{45,46}. Input constraints are defined in the decision space. They are usually defined by domain expertise to pre-emptively minimize sample wastage⁴⁷, whereas output constraints present a more challenging optimization scenario since the goal of efficiently achieving multiple objectives needs to be balanced with enough sampling of the feasibility boundaries to learn the constraints. This was previously demonstrated by Khatamsaz et al. for alloy design^{40,48} as well as by Cao et al. for liquid formulations¹⁶, where in all cases an appropriate machine learning model needs to be trained a posteriori to classify whether samples are feasible or not. Based on the good performance of EGBO on both the self-driving AgNP platform and the synthetic problems, we now explore how to adapt the algorithmic framework to reduce sampling wastage.

While the MW7 synthetic cMOOP is subject to output constraints, the two constraints of the AgNP cMOOP are input constraints, as they only depend on the flowrate input parameters. However, we chose to handle these as if they were output constraints, as the PF was not expected to be strongly restricted by them. Candidates from the entire decision space, including infeasible points, were considered during the optimization process, relying on GP constraint learning and/or the EA selection operator to avoid suggesting infeasible candidates. Infeasible candidates suggested by the optimizer were then repaired afterwards (post-repair).

This method is hence sensitive to the definition of the repair operator equations, which projects the infeasible solutions into the feasible space. The equations defining the constraints for the AgNP synthesis project most of the infeasible points along the Q_{seed} direction onto the constraint boundary (Supplementary Fig. 1.3). Since part of the constraint boundary intersects the POS, both algorithms benefit from the post-repair process to suggest more points close to the PF. However, this comes with a bias, as part of the unconstrained POS seems to be located at high Q_{AA} . As most of the high Q_{AA} infeasible points project towards a small region of the decision space, this leads to the oversampling detected at high y_2 (see Fig. 3a, b top).

To avoid biased sampling linked to the arbitrary choice of the repair operator, we propose to use the repair operator on the discrete candidate set before optimization. This repair method is referred to as the pre-repair process. For qNEHVI-BO, instead of $\hat{\alpha}_{NEHVI}$ optimization using `optimize_acqf`, a Sobol candidate set is first repaired, and then ranked by discretely evaluating $\hat{\alpha}_{NEHVI}$ values, with the best q candidates are suggested for evaluation.

Figure 5a illustrates the difference between post- and pre-repair methods on a simple 2D case scenario with a single optimization direction (red arrow). We see that the post-repair process favours infeasible samples that are optimizing the objective, even though these points become sub-optimal after the repairing operation. Therefore, even when the repair function redistributes the infeasible points uniformly on the constraint boundary, like in this simple example, post-repair might lead to a less optimal selection of the candidates.

We performed further simulations to compare pre-repair to post-repair for both qNEHVI-BO and EGBO (in this case the combined candidate set is repaired before ranking) on the AgNP synthesis problem. We trained a GP model on all data obtained in the experimental campaign (Supplementary Fig. 1.4) to provide a baseline ground truth. A total of 10 runs was performed over 30 iterations with a batch size of 4 for both repair types. Additionally, a purely random sampling (also Sobol in this case) was also performed as a baseline comparison to understand the distribution of feasible samples.

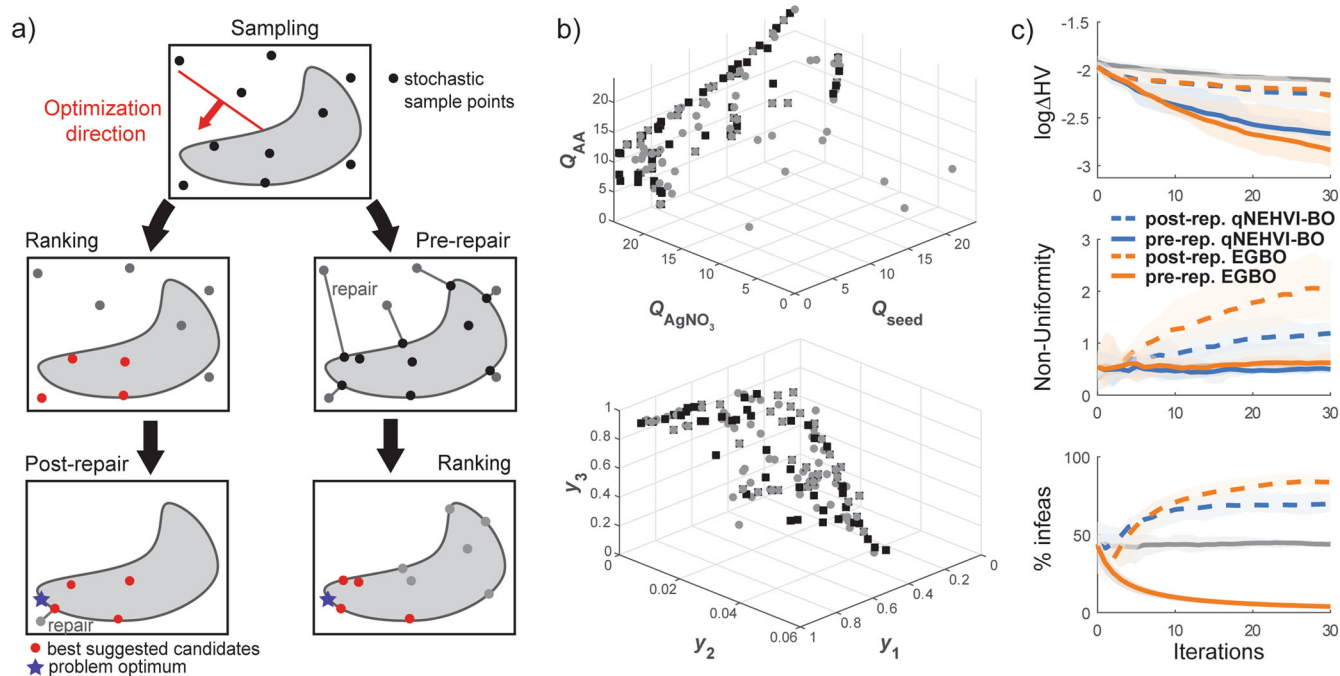


Fig. 5 | Handling of input constraints using pre- and post-repair. **a** Example of a 2D optimization towards a single optimum (blue star). In post-repair mode, stochastic sample points (black disks) are all considered during the optimization process, best candidates (red disks) are then repaired (gray line) to the nearest feasible point. In pre-repair mode, stochastic samples are first repaired (gray line) by projection to the feasible space, and the best repaired candidates (red disks) are

selected. **b** Location of dominated (grey disks) and non-dominated (black squares) points in the decision and objective space after a single run of EGBO with pre-repair. **c** Performance metrics for pure qNEHVI-BO (blue) and EGBO (orange), with dotted lines for post-repair and solid lines for pre-repair demonstrating the superiority of EGBO. We also include Sobol sampling (grey) as a baseline comparison.

Sample position for a single run is shown in the decision and objective spaces (Fig. 5b). We specifically plotted the decision space with Q_{AgNO_3} , Q_{seed} and Q_{AA} , as these are the flowrates affected by the repair operator. The sampling distribution of the non-dominated points (black square) in the objective space indicates that EGBO with pre-repair achieves a good coverage of the PF, without any oversampling at high y_2 . Part of these non-dominated points project into a line in the decision space, corresponding to the constraint boundary c_2 (Supplementary Fig. 1.3). But the other non-dominated points are still sampled within the feasible space. Interestingly, we also note that most of the non-dominated points are located on the box constraints, suggesting that better optimization results could have been achieved by increasing the maximal flowrate values.

Figure 5c reports the performance metrics of qNEHVI-BO and EGBO with post-repair (dash line) and pre-repair (solid line), compared to a pure Sobol sampling (grey line). We see a decrease of $\log\Delta\text{HV}$ with the pre-repair process for both qNEHVI-BO and EGBO, meaning that the algorithms get closer to the true PF. Furthermore, the NU metric for the pre-repair process remains as low as pure Sobol sampling, suggesting a relatively uniform sampling of the PF, as already suggested by Fig. 5b.

Finally, Fig. 5c (bottom) shows that optimization with post-repair suggested more infeasible points than Sobol sampling. The performance of such an approach strongly depends on the choice of the repair function, especially in the case of a constrained PF. If the repair operator projects infeasible points non-uniformly on the constraint boundary, or if it projects multiple infeasible points on the same place of the constraint boundary, as here in the case of AgNP synthesis, then the optimization will be biased. This bias can have dramatic effect if the POS is located at the constraint boundary, since most of the samples will be used to only explore a narrow region of the PF. Thus, post-repair requires a careful choice of the repair function as well as a priori knowledge on how the constraints restrain the PF. In contrast, pre-repair algorithms do not suggest any infeasible points. The only infeasible points will come from the initialization step. Therefore, pre-repair algorithms are ideal for handling input constraints on MAPs.

Further discussion

Overall, our pre-repair EGBO approach effectively combines the ability to deal with cMOOPs. The concept of integrating a gradient-based approach and EAs has been previously explored by Hickman et al.⁴¹. This constraint handling method is close to the approach reported in Handling input and output constraints section, as the objectives are only evaluated in the feasible region. However, their approach uses the constraint as a stop criterion for the gradient-based approach. As for the EA, infeasible children are projected onto the feasible boundary using binary search. This post-repair approach is more reliable than the one using a repair operator, as the binary search simply looks for the best feasible point located between the feasible parent and the infeasible offspring without bias. Although these two methods can converge towards the constrained PF, EGBO with pre-repair demonstrates a more robust approach that can cater to output constraints, as discussed above. Furthermore, Hickman's approach is limited to input constraints, where suggested infeasible samples are rejected, while EGBO is also designed to handle output constraints.

To showcase the generality of our results, we performed ablation studies where the EA sampling in EGBO is replaced with Multi-Objective Evolutionary Algorithm based on Decomposition with Improved Epsilon (MOEA/D-Iepsilon)⁴⁹, and the BoTorch optimization method `optimize_acqf` with a discrete Sobol sampling. We reported the results of these variants in Supplementary Discussion subsection 2.2, where we observed comparable performance using either U-NSGA-III or MOEA/D-Iepsilon in EGBO framework across multiple synthetic problems. We therefore confirm that the good performance obtained with EGBO is due to an intuitive balance of exploration from `optimize_acqf`/Sobol sampling and exploitation from EA sampling, where sample wastage and constraints are handled well. In essence, we can consider EGBO to be a no-regret implementation of BO, since there is an elitist algorithm (the EA component) which guarantees convergence while also maintaining sampling which are near-optimal (via considering only non-dominated parents).

Additionally, we note that our implementation of the AgNP self-driving lab used synchronous batch sampling where all samples in the batch must be evaluated at each iteration of optimization. Batch sampling is imperative for efficient optimization especially for a complex problem landscape. In our case, the experimental platform is technically $q = 1$, but we evaluate all 8 samples (4 from each optimizer) sequentially as a batch since the experimental run time for 8 samples is reasonable with a consistent time window (described in Fig. 2), where larger batch sizes would result in very long run times (leading to water wastage and need to refill the syringes) as well as to align the timing for the Jupyter notebook running the optimization code. A potential avenue for future work would be to extend EGBO to handle asynchronous batching^{50,51} to better handle this.

In conclusion, we have presented an efficient and general implementation of Evolution-Guided Bayesian optimization (EGBO) for multiple objectives with constraints – a problem that is common in materials and chemical sciences and engineering. We built a fully automated self-driving lab for silver nanoparticle synthesis and solved the challenge of high-quality synthesis with high yield and low seed particle usage using the EGBO approach. On both experimental and synthetic problems, we demonstrated the improved performance of EGBO based on 3 quantifiable metrics: (1) hypervolume, (2) non-uniformity and (3) sampling wastage. We also visualized the optimization results in terms of sampling distribution via probability density. Our optimization results on the self-driving lab highlighted the importance to properly integrate the repair operator into the overall optimization workflow when handling input constraints. We proposed a strategy to pre-repair these constraints.

Further simulations on the silver nanoparticle optimization problem predicted a significant increase in EGBO performance compared to the post-repair approach. Our approach allows handling of multiple objectives with explicit constraints, whether they are input or output, to map the Pareto Front effectively with limited sampling wastage – especially useful for industry-relevant problems where experiment time and chemical reactants are limited.

Methods

Experimental platform

The microfluidic platform is a customized droplet generator connected to a 1.5 m long serpentine reaction tube (Fig. 2a). The droplet generator (Fig. 2b) is 3D-printed by stereolithography with Somos® Waterclear Ultra material and coated with 10H ceramic coating to ensure stable droplet generation. Droplets are generated at the T-junction using silicone oil (10 cSt, 100 $\mu\text{L min}^{-1}$) as continuous phase and six coflowing aqueous solutions as dispersed phase. Small indentations are designed around the T-junction to prevent possible AgNP deposition in the T-junction inlet. After exiting the droplet generator, droplets continue flowing in the reaction tube (PFA, 1 mm ID), allowing the mixing of the reactants.

Flushing of all aqueous inlets is performed at the start of every sampling condition to prevent cross-contamination between conditions. All flowrates are then set to the values proposed by the algorithm and kept constant to generate droplets with the requested chemical composition. Due to AgNP size and shape evolution during the reaction, droplets change colour as they flow in the reactor. Once the reactor is full of droplets, a hyperspectral image of the reaction tube is taken (Fig. 2c). Data analysis is then performed to extract the absorbance spectra of each droplet and combine them into an absorbance map containing the temporal evolution of the absorbance spectra with the residence time (Fig. 1 bottom). The objectives and constraints are then derived from this absorbance map as well as from the flowrate values. Each sampled condition takes 10min30s to go through the whole experimental workflow (Fig. 2d).

Optimization algorithm

Below, we describe the pseudo-code for implementing the optimization loop in EGBO, with inputs: training data \mathbf{x} , problem $f(\mathbf{x})$, reference point ref_pt , surrogate model GP, batch size q

1: **while** $i < \text{maximum iterations}$:

2: **train** GP on $f(\mathbf{x})$

3: $\text{acq_func} = \hat{\alpha}_{\text{NEHVI}}(\text{GP}, \text{ref_pt})$

4: $\chi_{\text{BO}} = \text{optimize_acqf}(\text{acq_func}, q)$

5: non_dom solutions = **is_not_dominated**(\mathbf{x})

4: EA. **tell**(non_dom solutions)

6: $\chi_{\text{EA}} = \text{EA. ask}(\text{pop_size}, \text{reference_vector})$

6: $\chi_{\text{pool}} = \chi_{\text{BO}} \cup \chi_{\text{EA}}$

7: $\chi_{\text{new}} = \text{sortmax}(\chi_{\text{pool}}, \hat{\alpha}_{\text{NEHVI}}(\chi_{\text{pool}}), q)$

10: $X_{i+1} = f(\chi_{\text{new}})$

11: $i += 1$

For both EGBO and qNEHVI-BO, a list of SingleTaskGP (using ModelListGP) is used for training all objectives and constraints, taking a constant mean $\mu(\mathbf{x}) = C$ and a Matern covariance kernel $k(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu d})^\nu K_\nu(\sqrt{2\nu d})$ with $\nu = 2.5$. We applied the same optimization hyperparameters for defining $\hat{\alpha}_{\text{NEHVI}}(\mathbf{x})$ with num_samples = 128 for MC approximation, and optimization via optimize_acqf is performed with initial heuristic of raw_samples = 256.

The U-NSGA-III component is defined with pop_size = 256 as number of children proposed, and we set q reference vectors for TournamentSelection using s -energy distribution proposed in pymoo for good spacing. We take the default crossover and mutation operators with default hyperparameters, using SimulatedBinaryCrossover with eta = 30 and prob = 1.0, and PolynomialMutation with eta = 20. The specific library versions are as follows: GPyTorch 1.10, BoTorch 0.8.5, and pymoo 0.6.0.

Experimental campaign

For the optimizers' initialization, Sobol sampling is chosen as a space-filling design to provide good coverage of the decision space. Both algorithms start with the same $2 * (5 + 1) = 12$ initial candidates, implemented using draw_sobol_samples in BoTorch. At each new iteration, each optimizer suggests a batch of $q = 4$ new candidates. The reference point for computing qNEHVI acquisition function $\hat{\alpha}_{\text{NEHVI}}$ in both algorithms is set at [0,0,0]. At each new iteration, both optimizers suggest a batch of 4 new candidates.

The 8 (4 from each optimizer) post-repaired flowrate conditions proposed by both algorithms are sent to the HTE platform for evaluation. Experimental accuracy is ensured by performing a control condition at the start of the iteration to ensure the experimental reproducibility over previous iterations. To close the loop, the Jupyter notebook which runs the optimization will wait for the objectives and constraints tables to be updated before proceeding with the next iteration. The entire process is automated by setting appropriate delay times in LabVIEW (syringe pump control and hyperspectral imaging) and Python (data retrieval and optimization). The experimental campaign was stopped after 15 iterations due to time and resource considerations, giving a total of 72 samples per optimizer and 132 samples across the entire experimental campaign.

Repair operator

The feasibility of each candidate is first evaluated on the constraint equations, and the suggested flowrates repaired if infeasible, following these post-repair equations:

$$\text{if } c_1 > 0 : Q_{\text{AA}}^{\text{repaired}} = \min \left(\frac{Q_{\text{AgNO}_3}^{\text{suggested}}}{0.3}, Q_{\text{AA}}^{\text{suggested}} \right) \quad (6)$$

$$\text{if } c_2 > 0 : Q_{\text{seed}}^{\text{repaired}} = \max \left(\left(2 - \frac{Q_{\text{AgNO}_3}^{\text{suggested}}}{Q_{\text{AA}}^{\text{repaired}}} \right) \cdot Q_{\text{AgNO}_3}^{\text{suggested}}, Q_{\text{seed}}^{\text{suggested}} \right) \quad (7)$$

Synthetic studies

For the MW7 problem, we also initialized with a Sobol sampling of $2 * (n + 1)$ points, where n is the number of dimensions, with a batch size of $q = 4$. The reference point for HV computation is set to [1.2, 1.2].

Data availability

The source code for our work, as well data generated and analyzed during the current study can be found at [Kedar-Materials-by-Design-Lab/Constrained-Multi-Objective-Optimization-for-Materials-Discovery \(github.com\)](https://github.com/Kedar-Materials-by-Design-Lab/Constrained-Multi-Objective-Optimization-for-Materials-Discovery). A supplementary document is also available.

Code availability

The source code for our work, as well data generated and analyzed during the current study can be found at [Kedar-Materials-by-Design-Lab/Constrained-Multi-Objective-Optimization-for-Materials-Discovery \(github.com\)](https://github.com/Kedar-Materials-by-Design-Lab/Constrained-Multi-Objective-Optimization-for-Materials-Discovery). A supplementary document is also available.

Received: 8 November 2023; Accepted: 11 April 2024;

Published online: 13 May 2024

References

- Flores-Leonar, M. M. et al. Materials Acceleration Platforms: On the way to autonomous experimentation. *Curr. Opin. Green. Sustain. Chem.* **25**, 100370 (2020).
- Leong, C. J. et al. An object-oriented framework to enable workflow evolution across materials acceleration platforms. *Matter* **5**, 3124–3134 (2022).
- Seifrid, M., Hattrick-Simpers, J., Aspuru-Guzik, A., Kalil, T. & Cranford, S. Reaching critical MASS: Crowdsourcing designs for the next generation of materials acceleration platforms, *Matter*. <https://doi.org/10.1016/j.matt.2022.05.035> (2022).
- Burger, B. et al. A mobile robotic chemist. *Nature* **583**, 237–241 (2020).
- Phillips, T. W., Lignos, I. G., Maceiczky, R. M., DeMello, A. J. & DeMello, J. C. Nanocrystal synthesis in microfluidic reactors: where next? *Lab Chip* **14**, 3172–3180 (2014).
- Epps, R. W., Felton, K. C., Coley, C. W. & Abolhasani, M. Automated microfluidic platform for systematic studies of colloidal perovskite nanocrystals: Towards continuous nano-manufacturing. *Lab Chip* **17**, 4040–4047 (2017).
- Wagner, J. et al. The evolution of Materials Acceleration Platforms: toward the laboratory of the future with AMANDA. *J. Mater. Sci.* **56**, 16422–16446 (2021).
- Wang, T. et al. Sustainable materials acceleration platform reveals stable and efficient wide-bandgap metal halide perovskite alloys, *Matter* <https://doi.org/10.1016/j.matt.2023.06.040> (2023).
- Langner, S. et al. Beyond ternary OPV: high-throughput experimentation and self-driving laboratories optimize multicomponent systems. *Adv. Mater.* **32**, 1907801 (2020).
- Bash, D. et al. Multi-fidelity high-throughput optimization of electrical conductivity in P3HT-CNT composites. *Adv. Funct. Mater.* **31**, 2102606 (2021).
- Bezinge, L., Maceiczky, R. M., Lignos, I. & Kovalenko, M. V. A. J. deMello, Pick a color MARIA: adaptive sampling enables the rapid identification of complex perovskite nanocrystal compositions with defined emission characteristics. *ACS Appl. Mater. Interfaces* **10**, 18869–18878 (2018).
- Mekki-Berrada, F. et al. Two-step machine learning enables optimized nanoparticle synthesis. *Npj Comput. Mater.* **7**, 1–10 (2021).
- Jiang, Y. et al. An artificial intelligence enabled chemical synthesis robot for exploration and optimization of nanomaterials. *Sci. Adv.* **8**, eabo2626 (2022).
- Epps, R. W., Volk, A. A., Reyes, K. G. & Abolhasani, M. Accelerated AI development for autonomous materials synthesis in flow. *Chem. Sci.* **12**, 6025–6036 (2021).
- Erps, T. et al. Accelerated discovery of 3D printing materials using data-driven multiobjective optimization, *Sci. Adv.* **7**, eabf7435–eabf7435 (2021).
- Cao, L. et al. Optimization of formulations using robotic experiments driven by machine learning DoE. *Cell Rep. Phys. Sci.* **2**, 100295 (2021).
- Ye, Y. F., Wang, Q., Lu, J., Liu, C. T. & Yang, Y. High-entropy alloy: challenges and prospects. *Mater. Today* **19**, 349–362 (2016).
- Smedberg, H. & Bandaru, S. Interactive knowledge discovery and knowledge visualization for decision support in multi-objective optimization. *Eur. J. Oper. Res.* **306**, 1311–1329 (2023).
- Ma, Z. & Wang, Y. Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons. *IEEE Trans. Evol. Comput.* **23**, 972–986 (2019).
- Daulton, S., Balandat, M. & Bakshy, E. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Adv. Neural Inf. Process Syst.* **34**, 2187–2200 (2021).
- MacLeod, B. P. et al. A self-driving laboratory advances the Pareto front for material properties. *Nat. Commun.* **13**, 1–10 (2022).
- Low, A. K. Y., Vissol-Gaudin, E., Lim, Y.-F. & Hippalgaonkar, K. Mapping pareto fronts for efficient multi-objective materials discovery. *J. Mater. Inf.* **3**, 11 (2023).
- Prospolito, P., Burratti, L. & Venditti, I. Silver nanoparticles as colorimetric sensors for water pollutants. *Chemosensors* **8**, 26 (2020).
- Saadmim, F. et al. Efficiency enhancement of betanin dye-sensitized solar cells using plasmon-enhanced silver nanoparticles, in: *Adv Energy Res, Vol. 1: Selected Papers from ICAER2017*, Springer, 2020: pp. 9–18.
- Fernandes, I. J. et al. Silver nanoparticle conductive inks: Synthesis, characterization, and fabrication of inkjet-printed flexible electrodes. *Sci. Rep.* **10**, 8878 (2020).
- Naganthran, A. et al. Synthesis, characterization and biomedical application of silver nanoparticles. *Materials* **15**, 427 (2022).
- Wiley, B. J. et al. Maneuvering the surface plasmon resonance of silver nanostructures through shape-controlled synthesis. *J. Phys. Chem. B* **110**, 15666–15675 (2006).
- Mekki-Berrada, F., Xie, J. & Khan, S. A. High-throughput and High-speed Absorbance Measurements in Microfluidic Droplets using Hyperspectral Imaging. *Chem. Methods* **2**, e202100086 (2022).
- Rasmussen, C.E. Gaussian processes in machine learning, in: *Summer School on Machine Learning*, Springer, 2003: pp. 63–71.
- Auger, A., Bader, J., Brockhoff, D. & Zitzler, E. Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications. *Theor. Comput. Sci.* **425**, 75–103 (2012).
- Daulton, S., Balandat, M. & Bakshy, E. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Adv. Neural Inf. Process Syst.* **33**, 9851–9864 (2020).
- Balandat, M. et al. BoTorch: a framework for efficient Monte-Carlo Bayesian optimization. *Adv. Neural Inf. Process Syst.* **33**, 21524–21538 (2020).
- Hanaoka, K. Comparison of conceptually different multi-objective Bayesian optimization methods for material design problems. *Mater. Today Commun.* **31**, 103440 (2022).
- Bradford, E., Schweidtmann, A. M. & Lapkin, A. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *J. Glob. Optim.* **71**, 407–438 (2018).
- Konakovic Lukovic, M., Tian, Y. & Matusik, W. Diversity-guided multi-objective bayesian optimization with batch evaluations. *Adv. Neural Inf. Process Syst.* **33**, 17708–17720 (2020).
- Rasmussen, M. H., Duan, C., Kulik, H. J. & Jensen, J. H. Uncertain of uncertainties? A comparison of uncertainty quantification metrics for chemical data sets. <https://doi.org/10.1186/s13321-023-00790-0> (2023).
- Mitchell, M. An introduction to genetic algorithms, MIT press, 1998.
- Seada, H. & Deb, K. U-NSGA-III: a unified evolutionary optimization procedure for single, multiple, and many objectives: proof-of-principle results, in: *International Conference on Evolutionary Multi-Criterion Optimization*, Springer, 2015: pp. 34–49.
- Blank, J. & Deb, K. Pymoo: Multi-objective optimization in python. *IEEE Access* **8**, 89497–89509 (2020).

40. Khatamsaz, D. et al. Multi-objective materials bayesian optimization with active learning of design constraints: Design of ductile refractory multi-principal-element alloys. *Acta Mater.* **236**, 118133 (2022).
 41. Hickman, R. J., Aldeghi, M., Häse, F. & Aspuru-Guzik, A. Bayesian optimization with known experimental and design constraints for chemistry applications. *Digit Discov.* **1**, 732–744 (2022).
 42. Simpson, E. H. The interpretation of interaction in contingency tables. *J. R. Stat. Soc. Ser. B Methodol.* **13**, 238–241 (1951).
 43. Zitzler, E., Deb, K. & Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.* **8**, 173–195 (2000).
 44. Malkomes, G., Cheng, B., Lee, E. H. & Mccourt, M. Beyond the pareto efficient frontier: Constraint active search for multiobjective experimental design, in: International Conference on Machine Learning, PMLR, 2021: pp. 7423–7434.
 45. Tenne Y., Izui, K. & Nishiwaki, S. Handling undefined vectors in expensive optimization problems, in: European Conference on the Applications of Evolutionary Computation, Springer, 2010: pp. 582–591.
 46. Le Digabel, S. & Wild, S. M. A taxonomy of constraints in black-box simulation-based optimization. *Optim. Eng.* **25**, 1125–1143 (2024).
 47. Liu, Z. et al. Machine learning with knowledge constraints for process optimization of open-air perovskite solar cell manufacturing. *Joule* **6**, 834–849 (2022).
 48. Khatamsaz, D. et al. Bayesian optimization with active learning of design constraints using an entropy-based approach. *Npj Comput. Mater.* **9**, 49 (2023).
 49. Fan, Z. et al. An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions. *Soft Comput.* **23**, 12491–12510 (2019).
 50. Hickman, R. et al. Atlas: a brain for self-driving laboratories. <https://doi.org/10.26434/chemrxiv-2023-8nrxx> (2023).
 51. Tran, A. et al. aphBO-2GP-3B: a budgeted asynchronous parallel multi-acquisition functions for constrained Bayesian optimization on high-performing computing architecture. *Struct. Multidiscip. Optim.* **65**, 132 (2022).
- Institute for Functional Intelligent Materials (I-FIM, project No. EDUNC-33-18-279-V12).

Author contributions

These authors contributed equally: A.K.Y. Low, F. Mekki-Berrada. Conceived the research: A.K.Y. Low, F. Mekki-Berrada, K. Hippalgaonkar, S.A. Khan. Developed the optimization algorithm: A.K.Y. Low, A. Gupta, E. Vissol-Gaudin, Q. Li, Y.F. Lim, Y.S. Ong. Performed the synthetic experiments: A.K.Y. Low, F. Mekki-Berrada. Developed the AgNP self-driving lab: F. Mekki-Berrada, A. Ostudin, J. Xie, A.K.Y. Low. Performed the AgNP experiments: F. Mekki-Berrada, A. Ostudin. Interpreted the results: A.K.Y. Low, F. Mekki-Berrada, A. Gupta, E. Vissol-Gaudin. Wrote the manuscript with input from all co-authors: A.K.Y. Low, F. Mekki-Berrada, K. Hippalgaonkar, S.A. Khan.

Competing interests

KH owns equity in a startup focused on applying Machine Learning for Materials.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41524-024-01274-x>.

Correspondence and requests for materials should be addressed to Saif A. Khan or Kedar Hippalgaonkar.

Reprints and permissions information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024

Acknowledgements

The authors acknowledge funding from AME Programmatic Funds by the Agency for Science, Technology and Research under Grant No. A1898b0043 and No. A20G9b0135. KH also acknowledges funding from the National Research Foundation (NRF), Singapore under the NRF Fellowship (NRF- NRFF13-2021-0011). SAK and FMB also acknowledge funding from the 25th NRF CRP programme (NRF-CRP25-2020RS-0002). QL also acknowledges support from the NRF fellowship (project No. NRF- NRFF13-2021-0005) and the Ministry of Education, Singapore, under its Research Centre of Excellence award to the