RECOB: RELIABLE BENCHMARKING OF EXPERIMENTAL OPTIMIZATION IN CHEMISTRY AND MATERIALS SCIENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimization methods for experimental science are often evaluated on synthetic functions that are reproducible but omit important characteristics of real experiments. We introduce RECOB (REliable Chem Optimization Benchmark), a black-box optimization benchmark constructed exclusively from data generated through physical experiments in chemistry and materials science. The suite contains 14 single-objective and two multi-objective tasks spanning chemical reactions, material formulations, electrochemical systems, continuous-flow processes, and automated laboratories. Each task provides a machine-readable specification of its decision variables, feasible domain, physical constraints, objective direction, and experimental provenance. Continuously queryable learned oracles are screened using repeated holdout validation and prespecified admission criteria, while measured-table replay enables evaluation using the original experimental responses. Under a common paired evaluation protocol, we compare ten single-objective and eight multi-objective optimization methods. HEBO achieves the best aggregate single-objective rank, while qNEHVI leads the multi-objective comparison. Model-based methods generally outperform non-adaptive baselines, although their computational overhead varies substantially. We further assess benchmark reliability using independent oracle retraining and measured-table replay. Aggregate optimizer rankings remain highly consistent across retrained oracles, while replay preserves the broad performance hierarchy using only physically measured responses. Together, these results show that RECOB can reproducibly distinguish optimizer performance as an experimentally grounded and reliability-tested benchmark for black-box optimization in chemistry and materials science.

1 INTRODUCTION

Many discovery and process-development problems in experimental science involve optimizing experimental conditions, material compositions, or hyperparameter tuning (Yu et al., 2026; Luo et al., 2025; Wu et al., 2019). These tasks can often be formulated as expensive black-box optimization problems: the objective has no analytical expression, gradients are inaccessible, and each evaluation requires a physical experiment (Jones et al., 1998). Because experiments are typically time-consuming and may consume costly materials, the central goal is to identify high-performing conditions as rapidly as possible within a limited evaluation budget (Shields et al., 2021). Bayesian optimization (BO), evolutionary algorithms (EA), and other derivative-free methods have therefore been widely adopted for closed-loop experimental design in chemistry, materials science, and automated laboratories (Shahriari et al., 2015; Bäck & Schwefel, 1993; Burger et al., 2020; Dave et al., 2022).

Developing and comparing these methods requires testbeds that are reproducible, inexpensive to evaluate, and representative of real applications. Analytical test functions and standardized suites such as COCO/BBOB support exact evaluation and large-scale replication, but often fail to capture mixed variable types, physical constraints, and compositional domains (Hansen et al., 2021; 2009). Recent benchmarks have narrowed this gap using experimental reaction tables (Shields et al., 2021), pool-based materials datasets (Liang et al., 2021), mechanistic simulators (Felton et al., 2021), and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

learned emulators (Häse et al., 2021b). However, three important challenges remain. First, publicly available tasks cover a limited range of experimental systems, and their constraints, conditional spaces, and provenance are rarely represented in a unified, machine-readable format (Hickman et al., 2022). Second, prior work on learned benchmarks has shown that predictive quality alone is insufficient to establish benchmark fidelity: downstream optimizer rankings and search trajectories must also be preserved (Zela et al., 2020; Pfisterer et al., 2022). Third, finite candidate tables and continuously queryable emulators define different optimization problems, yet their agreement has not been systematically studied.

To address these challenges, we introduce **RECOB** (**RE**liable **C**hem **O**ptimization **B**enchmark), a black-box optimization benchmark constructed from real chemistry and materials experiments. We curate 19 experimental datasets originating from approximately 17 peer-reviewed publications. All response values are grounded in physical experiments rather than analytical test functions or computational-property calculations. Our inclusion criteria require that the input variables be experimentally controllable, that the objectives be directly measured or have a clearly defined experimental interpretation, and that data collection in the original study be explicitly associated with experimental optimization or design-space exploration. For each dataset, we reconstruct the variable types, experimental bounds, joint constraints, conditional spaces, objective directions, campaign information, and data provenance.

We train AutoGluon ensembles as continuously queryable oracles and screen them using five repeated holdout splits and prespecified admission criteria (Erickson et al., 2020). Based on predictive accuracy, worst-split performance, and between-split variability, tasks are assigned to core or conditional tiers. For single-objective evaluation, we select 14 tasks from nine independent publications and compare ten optimizers under a paired protocol in which all algorithms share the same initial design for each task–trial pair. We assess final and anytime performance, cross-task ranking stability, and candidate-generation cost. For datasets with multiple scientifically meaningful objectives, we also construct a multi-objective extension covering hypervolume-based, scalarization-based, information-theoretic, and evolutionary methods.

We evaluate benchmark robustness in two ways. First, we repeat the benchmark using independently trained oracles with different training seeds to quantify the stability of optimizer rankings. Second, we introduce a measured-candidate track that maps each proposal to the nearest unused experimental condition and returns that condition together with its recorded response. We then compare optimizer rankings between the two protocols.

Our main contributions are as follows:

1. We construct RECOB, an experimentally grounded benchmark comprising 14 single-objective and two multi-objective tasks across real-world problems in chemistry and materials science. Each task includes machine-readable variable definitions, feasible domains, physical constraints, objective directions, and data provenance.
2. We establish reproducible continuous-oracle and measured-candidate evaluation protocols, and compare ten single-objective and eight multi-objective optimizers under shared initial designs and equal evaluation budgets. The evaluation covers final and anytime performance, cross-task rankings, and optimizer computational overhead.
3. We assess benchmark reliability through independently trained oracles and measured-table replay. Optimizer rankings are highly stable across oracle training seeds, while replay preserves broad performance groups but not exact rankings on every task.

2 Related Work

Current optimization benchmarks range from analytical test suites to domain-specific virtual environments. COCO/BBOB (Hansen et al., 2021; 2009) provides exactly and repeatedly evaluable functions for large-scale algorithm comparison, but does not reproduce many features of experimental search spaces, such as mixed variables, compositional constraints, and historically sampled domains. Summit (Felton et al., 2021) improves domain relevance through chemistry-motivated mechanistic and data-driven virtual environments, while remaining primarily simulator- or emulator-based. Benchmarks constructed from historical experiments generally follow either a finite-candidate or a learned-emulator protocol: Shields et al. (Shields et al., 2021) evaluated

reaction-optimization methods on high-throughput experimental tables, while Liang et al. (Liang et al., 2021) used pool-based campaigns across several materials datasets. These protocols preserve recorded experimental responses but restrict evaluation to previously measured candidates. Olympus (Häse et al., 2021b; Hickman et al., 2023), by contrast, trained continuously queryable emulators from experimental datasets, enabling optimization beyond the original candidate tables but introducing dependence on learned response models.

Validation of learned benchmarks has been studied more explicitly in other domains. YAHPO Gym (Pfisterer et al., 2022) showed that predictive metrics should be complemented by downstream checks of optimization behavior and optimizer rankings. Following this principle, RECOB screens oracle predictive quality before task admission and evaluates whether aggregate optimizer conclusions remain stable across independently trained oracles and measured-table replay. Table 1 summarizes the resulting differences in data provenance, evaluation protocol, validation, and task support.

Table 1: Comparison of representative black-box optimization benchmarks.

Benchmark	**Data**		**Oracle and validation**			**Task support**	
	Physical experiments	**Measured track**	**Continuous oracle**	**Predictive validation**	**Rank/trajectory audit**	**Constraints / mixed**	**Objectives**
COCO/BBOB	No	No	Analytic	N/A	N/A	Limited	SO + MO
Summit	Partial	Not std.	Simulator / emulator	Task-specific	Not systematic	Yes	SO + MO
Shields et al.	Yes	Yes	No	N/A	N/A	Primarily discrete	Primarily SO
Liang et al.	Yes	Yes	No	N/A	N/A	Limited	SO
Olympus	Yes	Not std.	Learned	Yes	Not systematic	Yes	SO + limited MO
YAHPO Gym	No	Tabular comparison	Learned	Yes	Yes	Yes	SO + MO
RECOB	**Yes**	**Yes**	**Learned**	**Yes**	**Yes**	**Yes**	**SO + MO**

Notes. "Measured track" denotes an explicitly defined evaluation protocol based on previously measured candidates. "Not std." means that relevant data may be available, but the corresponding track is not standardized as part of the benchmark. SO and MO denote single- and multi-objective optimization, respectively.

Table 2: Frozen quantitative criteria for learned-oracle admission.

Criterion	Core (all required)	Conditional (all required)	Not admitted (any sufficient)
Labeled observations	≥ 50	≥ 40	< 40
Splits with $R^2 > 0$	$\geq 4/5$	$\geq 3/5$	$< 3/5$
Mean R^2	≥ 0.50	≥ 0.20	< 0.20
Mean Pearson r	≥ 0.70	≥ 0.50	< 0.50
Mean range-NRMSE	≤ 0.15	≤ 0.25	> 0.25

Related benchmark efforts also include offline model-based optimization. Design-Bench (Trabucco et al., 2022) evaluates methods that optimize designs using only a fixed historical dataset, without receiving additional objective feedback during optimization, while SOO-Bench (Qian et al., 2025) further examines stability across offline data distributions. In contrast, RECOB focuses on sequential optimization, where optimizers can repeatedly query an objective and use newly obtained responses to inform subsequent decisions. Therefore, these offline optimization benchmarks and corresponding optimization algorithms are not discussed in this work.

3 Benchmark construction

3.1 Data Processing

RECOB was designed to represent optimization in experimentally measured systems. We therefore collected data only from source studies reporting physical experiments, in which controllable experimental variables and measured responses could be formulated as an optimization problem.

Experimental tables were obtained from the article text, Supporting Information, author-maintained repositories, or previously released benchmark datasets. The complete list of source studies and provenance records is provided in Appendix A Table 6.

For each dataset, variables explicitly treated as optimization decisions in the source study were used as input features, while experimentally measured quantities identified as optimization objectives were used as response variables. Variable names and units were standardized without altering the measured values. No missing-value imputation or general outlier filtering was applied to the model-ready data.

Each dataset was then represented by a machine-readable optimization specification defining the external decision variables, model inputs, variable types and units, allowed categories or numerical bounds, joint and nonlinear constraints, and objective direction. When a source contained chemically distinct reactions or experimental campaigns, these were represented as separate optimization tasks.

The initial collection comprised 19 experimental datasets from 17 source studies, containing 5,291 experimental records and 43 candidate target–space combinations. These candidates were subsequently evaluated through the oracle-construction and screening procedure described below.

3.2 Oracle Construction

Continuously queryable oracles were constructed using AutoGluon Tabular (Erickson et al., 2020). A common automated machine-learning pipeline was used across all datasets to accommodate differences in sample size, dimensionality, scale, and variable type, while reducing task-specific discretion in model and hyperparameter selection. Before fitting the frozen oracle on all available data, we evaluated out-of-sample predictive quality using five fixed 80/20 holdout splits. Candidate tasks with insufficient or unstable validation performance were excluded from the continuous-oracle benchmark, thereby limiting the suite to tasks for which the experimentally measured input–response relationship could be reproduced with acceptable accuracy. For candidate tasks containing repeated exact input configurations, all repetitions were assigned to the same partition, and the resulting group-split metrics were used for screening to prevent leakage between training and validation data.

For each split, we computed the coefficient of determination (R^2), Pearson correlation coefficient (r), and range-normalized root mean squared error,

$$\mathrm{NRMSE}_{\mathrm{range}} = \frac{\sqrt{n^{-1}\sum_i (y_i - \hat{y}_i)^2}}{y_{\max} - y_{\min}},$$

where $y_{\max}$ and $y_{\min}$ denote the maximum and minimum measured responses for the corresponding task. Oracle admission was jointly determined by the number of labeled observations, the mean R^2, mean Pearson r, mean range-NRMSE, and the number of validation splits with positive R^2.

The quantitative criteria used for oracle admission are summarized in Table 2. A candidate was assigned to the Core set only if it satisfied all Core criteria. Among the remaining candidates, those satisfying all Conditional criteria were assigned to the Conditional set, whereas failure of any Conditional criterion resulted in exclusion from the aggregate continuous-oracle evaluation. Quantitative screening was applied only after the task definition and feasible optimization domain had been established during data processing.

Application of these criteria produced 10 Core and four Conditional tasks, of which the results are reported in Table 3. Across the 14 admitted tasks, mean R^2 ranged from 0.261 to 0.938, mean Pearson correlation from 0.618 to 0.970, and mean range-NRMSE from 0.057 to 0.231. The tier assignments are provided as task-level quality metadata rather than treated as an experimental factor in the optimizer comparison.

Following screening, the selected AutoGluon ensemble for each task was refitted using all valid labeled observations and frozen as the deployed oracle. Experimental noise was not explicitly simulated because its sources and distributions are task-specific and cannot be faithfully represented by a common Gaussian noise model. Every candidate is checked against the declared schema and feasibility constraints before prediction; invalid categories, off-grid values, non-finite inputs, and bound or joint-constraint violations are rejected rather than silently corrected. Multi-objective oracles are constructed by evaluating compatible single-objective models on the same validated candi-

Table 3: Predictive validation results and admission outcomes for the 14 retained single-objective tasks.

Experimental source	Opt target	Reference	Labeled observations	Positive-R^2 splits	Mean R^2	Mean Pearson r	Mean range-NRMSE
AgNP synthesis	Reaction rate	Low et al. (2024)	144	5/5	0.938	0.970	0.077
OPV formulation	Degradation	Langner et al. (2020)	2,322	5/5	0.839	0.918	0.057
Mobile robotic chemist	Hydrogen evolution	Burger et al. (2020)	595	5/5	0.821	0.913	0.103
BRINE	Conductivity, campaign 2	Ramezani et al. (2026)	121	5/5	0.763	0.884	0.114
Amide coupling	Yield, reaction 1	Wagner et al. (2024b)	117	5/5	0.754	0.886	0.094
3D-printing materials	Compression modulus	Erps et al. (2021)	143	5/5	0.718	0.868	0.083
Continuous-flow chemistry	N-benzylation impurity	Schweidtmann et al. (2018)	73	5/5	0.852	0.934	0.106
Continuous-flow chemistry	SNAr E-factor	Schweidtmann et al. (2018)	66	5/5	0.838	0.918	0.108
P3HT–CNT composites	Electrical conductivity†	Bash et al. (2021)	233	5/5	0.694	0.840	0.113
3D-printing materials	Toughness	Erps et al. (2021)	143	5/5	0.517	0.740	0.112
Li-ion electrolyte	Conductivity†	Dave et al. (2022)	125	4/5	0.376	0.618	0.157
Amide coupling	Yield, reaction 4	Wagner et al. (2024b)	40	5/5	0.529	0.832	0.231
Amide coupling	Yield, reaction 2	Wagner et al. (2024b)	72	5/5	0.454	0.711	0.177
BRINE	Conductivity, campaign 1	Ramezani et al. (2026)	114	5/5	0.261	0.621	0.147

Notes. Rows above and below the intermediate horizontal rule correspond to Core and Conditional tasks, respectively. Tasks marked with † were evaluated using exact-input group splits; all other tasks used five repeated random 80/20 holdout splits. Classification follows the criteria in Table 2.

date and concatenating their predictions, without fitting additional models. Training configurations, validation results, artifact hashes, and per-task screening decisions are provided in the Supporting Information.

4 Evaluation Protocol

4.1 Benchmark Interaction and Fairness Controls

RECOB follows a sequential ask–evaluate–tell protocol. At iteration t, an optimizer proposes a candidate $\mathbf{x}_t$ for the task domain $\mathcal{X}$. After validation, the corresponding model features are passed to the frozen learned oracle, which returns a scalar response $\hat{y}_t = \hat{f}(\mathbf{x}_t)$ for a single-objective task, or a response vector $\hat{\mathbf{y}}_t = [\hat{f}_1(\mathbf{x}_t), \ldots, \hat{f}_m(\mathbf{x}_t)]$ for a multi-objective task. Each response with batch size 1 is supplied to the optimizer for its next proposal. Multi-objective tasks reuse the corresponding frozen single-output oracles and do not require an additional multi-output model.

The benchmark interface supports continuous, discrete, categorical, and mixed-variable search spaces. Directly submitted candidates are checked against the bounds, legal categories, step grids, and joint or nonlinear constraints defined by the task. In particular, a step-constrained variable is valid only when $x_j = l_j + k\Delta_j$ for some $k \in \mathbb{Z}$, within the stored numerical tolerance. In the formal algorithm comparisons, optimizer outputs were first converted into legal external candidates using the same task-aware domain mappings, including grid mapping for step-constrained variables, and were then passed through the common validator. Derived model inputs were computed only after validation.

We used a paired design comprising 20 trials per task. Each task–trial pair was associated with a frozen initial design containing ten feasible, non-duplicate conditions. Every algorithm received the same 10 conditions in the same order and observed the corresponding oracle responses before algorithm-specific sampling began. Post-initialization random streams were generated using a deterministic seed policy, with independent optimizer seeds derived for each task, trial, and algorithm.

Each run contained ten initialization evaluations followed by 100 sequential optimization evaluations, giving a total budget of $N_{\text{eval}} = 10 + 100 = 110$. Internal model fitting and acquisition

optimization did not receive additional oracle calls. In the single-objective experiments, duplicate decoded proposals were resampled before evaluation, with the associated suggestion time retained; in the multi-objective experiments, repeated candidates remained in the recorded evaluation history and budget. No per-suggestion performance timeout was imposed in the completed formal experiments, and checkpointing allowed interrupted runs to resume without altering their histories. For a given task, all algorithms used the same oracle artifact, task definition, validation logic, objective direction, and initial-design assets.

Optimization performance and computational overhead were reported separately: the single-objective study records candidate-suggestion time, whereas the multi-objective runs retain end-to-end run time. All formal runs were submitted through Slurm to a dual-socket compute node equipped with two AMD EPYC 7702 64-core processors and 256 GB of DDR4 memory. Each task–algorithm–trial run was scheduled as an independent job with two CPU cores and 8 GB of memory. Reported suggestion times include only the recorded candidate-generation time of completed runs and exclude Slurm queueing time.

4.2 Evaluated Optimizers and Performance Metrics

The evaluated algorithms were selected to represent a broad range of search mechanisms. We considered only methods that can operate without task-specific scientific prior knowledge; each optimizer received only the variable types, feasible domain, constraints, and observations provided by the benchmark. The ten single-objective methods comprise two non-adaptive baselines (Random Search and scrambled Sobol Search (Owen, 1998)); three Gaussian-process BO methods (BoTorch LogEI, BoTorch UCB (Balandat et al., 2020), and BoFire LogEI (Dürholt et al., 2025)); a SMAC-inspired random-forest expected-improvement method (Hutter et al., 2011); a density-estimation method (Optuna TPE (Akiba et al., 2019)); an evolution strategy (CMA-ES (Hansen et al., 2003)); and two Bayesian optimization frameworks designed for complex search spaces (HEBO (Cowen-Rivers et al., 2022) and Gryffin (Häse et al., 2021a)). The RF/SMAC-like EI method is our own random-forest ensemble implementation and does not use the official SMAC software.

The eight multi-objective methods comprise Random Search and Sobol Search as non-adaptive baselines; NSGA-II (Deb et al., 2002) as an evolutionary method; qNEHVI (Daulton et al., 2021) as a hypervolume-based Bayesian optimization method; qNParEGO (Daulton et al., 2021) as a scalarization-based method; TSEMO (Bradford et al., 2018) as a Thompson-sampling method; and MESMO (Belakaria et al., 2019) and JES (Tu et al., 2022) as information-theoretic methods. The qNEHVI and qNParEGO experiments use their numerically stable logarithmic implementations. All multi-objective methods propose one new candidate per iteration. Implementation details and hyperparameters of all single- and multi-objective methods are provided in Appendix B.

For single-objective tasks, all responses were converted to a maximization convention. Since the true global optimum is unknown, we use the best historically observed value as the reference for computing normalized regret. For task d, let z_d^{best} and z_d^{worst} be the extrema of the frozen union of the 2,800 original ten-algorithm evaluation histories. For the best-so-far maximization response b_t, define $\widetilde{r}_t = \frac{z_d^{\mathrm{best}} - b_t}{z_d^{\mathrm{best}} - z_d^{\mathrm{worst}}}$. The endpoints are post hoc reporting constants and remain fixed when evaluating additional algorithms. Values are not clipped: a negative gap identifies an improvement beyond the frozen reference, and a value above one is also permitted. Final performance is $\widetilde{r}_{110}$; anytime performance is the trapezoidal mean of $\widetilde{r}_t$ from evaluations 10 through 110. Lower values are better. We report both mean and sample standard deviation and median with first and third quartiles across 280 task–trial runs per algorithm. Each task contributes 20 trials. For rankings, algorithms are compared by their median final gap within each task; ties receive average ranks and fractional wins. Overall rank averages these 14 task ranks.

For multi-objective tasks, all reported hypervolumes use one frozen scale and reference per task. For objective j, let l_j and u_j be the minimum and maximum over the 200 responses in the 20 frozen common initial designs (ten conditions per trial), read once from the Random Search histories to avoid duplicating the shared starts. The reporting transformation is $\widetilde{y}_j = (y_j - l_j)/(u_j - l_j)$ and the maximization reference is $(-0.1, \ldots, -0.1)$. These constants are shared across all algorithms, trials, and budgets within a task and are provided in Appendix B. Final performance is HV at evaluation 110, and AUC-HV is its trapezoidal mean from evaluations 10 through 110. Both are summarized

as mean and sample standard deviation over 20 trials per task. Algorithms are ranked by final HV within each task–trial block; the overall mean rank averages all 40 blocks.

5 Experimental Results

5.1 Benchmark Results

We evaluated ten single-objective optimizers on the 14 RECOB tasks. Table 4 summarizes final and anytime performance. HEBO achieved the best final aggregate performance, with the lowest mean task rank and the lowest mean final normalized simple regret. It won eight tasks outright and shared the best median result on two others. BoFire LogEI and BoTorch LogEI occupied the next two positions, followed by a more closely grouped set of RF/SMAC-like EI, BoTorch UCB, and CMA-ES. Gryffin, Sobol Search, and Random Search consistently remained near the bottom of the final ranking. The overall differences were significant according to the Friedman test ($\chi^2 = 81.30$, $p = 8.92 \times 10^{-14}$). Pairwise Wilcoxon signed-rank tests indicated that HEBO significantly outperformed all other methods except BoFire LogEI ($p = 0.084$) and BoTorch LogEI ($p = 0.084$).

Final performance and optimization speed did not produce exactly the same ordering. BoTorch LogEI obtained the lowest normalized regret AUC, indicating the strongest overall early-budget performance, while BoFire LogEI and HEBO followed closely. HEBO continued to improve over the longer budget and ultimately achieved the strongest final rank. CMA-ES and RF/SMAC-like EI also improved progressively, whereas Random and Sobol Search fell behind because their proposals did not adapt to the observed responses.

As a sensitivity analysis, we recomputed the single-objective rankings separately for the two fidelity tiers. The two rankings were broadly consistent (Spearman $\rho = 0.794$): HEBO ranked first in both subsets, whereas Gryffin, Sobol Search, and Random Search remained in the lower-performing group. Conditional tasks showed lower mean normalized final regret and regret AUC than the Core tasks, although neither difference was statistically significant at the task level. Thus, the admission tier did not introduce a qualitative change in the main performance hierarchy.

Performance remained task dependent. GP-based methods, CMA-ES, and TPE each won at least one problem, while HEBO was strongest across a broader subset of the suite. Final discrimination was weaker for OPV degradation and printing compression modulus because several algorithms reached the same oracle plateaus, although their regret AUC values and optimization trajectories still distinguished how rapidly those plateaus were reached.

Computational overhead. The mean suggestion time varied by more than five orders of magnitude. HEBO required approximately 3.50 seconds per optimization step, compared with 16.91 and 14.65 seconds for BoFire LogEI and BoTorch LogEI, respectively. RF/SMAC-like EI retained competitive performance with only 0.17 seconds per step, providing a favorable performance–cost compromise, while CMA-ES and Optuna TPE incurred negligible suggestion overhead. Gryffin had both the largest mean suggestion time and a low final rank. These measurements describe successful candidate-generation steps and were not incorporated into the performance ranking.

Eight multi-objective methods were evaluated on the three-objective amide task and the two-objective printing task. As shown in Table 5, qNEHVI achieved the best overall final-HV rank and the highest final HV on the amide task, whereas MESMO obtained the highest final HV on printing. qNEHVI also achieved the highest mean AUC-HV on both tasks, indicating that it constructed useful nondominated sets more rapidly across the full evaluation budget.

On the amide task, qNEHVI ranked first in 18 of 20 paired trials and significantly outperformed every alternative. Its performance reflected a favorable balance across the three reaction objectives rather than improvement in only one response. On printing, MESMO had the largest mean final HV. Its difference from qNEHVI was not statistically decisive (adjusted $p = 0.273$), whereas its advantage over qNParEGO was significant (adjusted $p = 0.0215$).

Computational requirements differed substantially among the multi-objective methods. qNParEGO had the lowest total wall time among the three leading model-based approaches, while retaining the third-best overall rank. NSGA-II required only 0.33 cumulative hours and improved substantially

Table 4: Aggregate single-objective performance across 14 tasks. Regret values are reported as mean ± standard deviation over 280 task–trial runs. Mean suggestion time excludes the ten shared initialization evaluations.

Algorithm	Mean rank	Task wins	Final normalized simple regret ↓	Normalized regret AUC ↓	Mean suggestion time (s step^{-1}) ↓
HEBO	1.71	8.5	0.062 ± 0.117	0.126 ± 0.148	3.50
BoFire LogEI	3.96	1.5	0.075 ± 0.114	0.125 ± 0.145	16.91
BoTorch LogEI	4.29	1.5	0.071 ± 0.112	0.111 ± 0.133	14.65
RF/SMAC-like EI	4.43	0	0.080 ± 0.080	0.146 ± 0.130	0.170
BoTorch UCB	4.46	0.5	0.115 ± 0.169	0.149 ± 0.183	12.00
CMA-ES	4.57	1.0	0.084 ± 0.089	0.160 ± 0.118	5.88×10^{-4}
Optuna TPE	5.43	1.0	0.105 ± 0.108	0.172 ± 0.142	9.54×10^{-3}
Gryffin	8.36	0	0.178 ± 0.146	0.231 ± 0.156	34.29
Sobol Search	8.50	0	0.189 ± 0.144	0.230 ± 0.151	8.47×10^{-5}
Random Search	9.29	0	0.200 ± 0.154	0.240 ± 0.161	5.71×10^{-5}

Table 5: Final and anytime multi-objective performance over 20 paired trials per task. HV and AUC-HV are reported as mean ± standard deviation. Total wall time is summed over the 40 successful runs of each algorithm and excludes scheduler queueing and unsuccessful preliminary attempts.

Algorithm	Amide final HV ↑	Amide AUC-HV ↑	Printing final HV ↑	Printing AUC-HV ↑	Mean rank	Wall time (h) ↓
qNEHVI	1.651 ± 0.020	1.553 ± 0.027	1.423 ± 0.062	1.280 ± 0.060	1.650	22.29
MESMO	1.588 ± 0.022	1.522 ± 0.026	1.437 ± 0.084	1.276 ± 0.098	2.200	17.39
qNParEGO	1.552 ± 0.056	1.424 ± 0.068	1.397 ± 0.062	1.243 ± 0.067	3.175	11.79
JES	1.566 ± 0.024	1.487 ± 0.019	1.341 ± 0.081	1.156 ± 0.071	3.275	18.04
TSEMO	1.504 ± 0.065	1.373 ± 0.070	1.215 ± 0.046	1.033 ± 0.054	4.825	27.06
NSGA-II	1.324 ± 0.109	1.097 ± 0.124	1.080 ± 0.108	0.915 ± 0.070	6.125	0.33
Sobol Search	1.173 ± 0.121	1.037 ± 0.104	0.955 ± 0.056	0.864 ± 0.045	7.250	0.41
Random Search	1.085 ± 0.127	0.970 ± 0.126	0.946 ± 0.091	0.862 ± 0.066	7.500	0.38

over Random and Sobol Search, although it did not match the final or anytime performance of the Bayesian methods. TSEMO incurred the largest total wall time but ranked fifth.

Overall, the multi-objective results distinguish adaptive from non-adaptive strategies as well as different Bayesian and evolutionary search mechanisms. Nevertheless, the cross-task ranking remains descriptive because the current multi-objective suite contains only two continuous, all-maximization tasks, and HV values are comparable only within the same task.

5.2 Benchmark Fidelity and Robustness

We evaluated RECOB reliability through oracle retraining and measured-table replay. The former tests whether benchmark conclusions are sensitive to randomness in oracle construction, whereas the latter provides a comparison using the original measured experimental responses. Additional quantitative results for these analyses, including prediction agreement across independent oracle fits, replay-pool construction, and task-level matching-distance diagnostics, are provided in Appendix D.

For the retraining analysis, independent AutoGluon oracles were trained with seeds 2026, 2027, and 2028 for six representative tasks: AgNP reaction rate, amide reaction-1 yield, BRINE campaign-2 conductivity, mobile hydrogen evolution, OPV degradation, and printing compression modulus. Pairwise correlations between the resulting ten-algorithm rankings were consistently high, with Spearman correlations of 0.964–1.000 and Kendall correlations of 0.911–1.000. The leading, intermediate, and lower-performing groups were preserved across all three retrainings: HEBO, BoFire LogEI, and BoTorch LogEI consistently occupied the leading group; BoTorch UCB, RF/SMAC-like EI, CMA-ES, and Optuna TPE remained in the intermediate group; and Gryffin, Sobol Search, and Random Search consistently occupied the final three positions. These results show that the optimizer comparisons are insensitive to the random seed used to train the AutoGluon oracles.

The numerical predictions were similarly stable when independently trained oracles were evaluated on shared response-independent candidate sets. Pairwise prediction correlations were 0.986–0.999, rank correlations were at least 0.976, and range-normalized mean absolute errors were 0.0046–0.0179. Thus, the independently trained oracles also produced closely agreeing predictions.

For measured-table replay, numerical variables were scaled by their declared ranges, while categorical mismatches were assigned a distance of one and matches a distance of zero. Each proposal was matched to the nearest unused experimental row. The matched candidate and its recorded response were returned to the optimizer, after which the row was removed from the candidate pool. At the matched budget of 60 evaluations, the learned-oracle and replay rankings had a Spearman correlation of 0.758 and a Kendall correlation of 0.556. BoFire LogEI and BoTorch LogEI remained among the top three methods under both protocols, while BoTorch UCB obtained the best replay rank. HEBO moved from the leading learned-oracle group to sixth place under replay. At the lower end, Gryffin, Sobol Search, and Random Search continued to occupy the final three positions. Therefore, replay preserved the broad distinction between the stronger model-based methods and the lower-performing baselines, despite some changes within the leading and intermediate groups.

At 110 evaluations, the measured-table replay produced a lower aggregate rank correlation of $\rho = 0.455$ because of the finite-candidate evaluation mechanism. Each optimizer proposal is mapped to the nearest unused experimental row, which is subsequently removed from the candidate pool. As the number of replay iterations increases, different algorithms increasingly encounter overlapping high-performing records, while candidate depletion and final-value ties compress their performance differences. Consequently, measured-table replay becomes less discriminative at larger evaluation budgets, and small differences in the order in which rows are selected can produce larger changes in the final ranks.

HEBO showed the largest change between the two protocols. Its strong continuous-oracle performance suggests that it benefits from concentrating search within narrow promising regions of a continuously queryable response surface. Under replay, however, each proposal is replaced by the nearest unused experimental row. Once nearby rows have been evaluated, subsequent proposals may be matched to increasingly different conditions, weakening the advantage of concentrated continuous search.

The two analyses revealed different levels of sensitivity: oracle retraining had little effect on optimizer rankings, whereas measured-table replay changed the ordering within the leading group. Independent AutoGluon retraining produced highly consistent oracle predictions and algorithm rankings, indicating that the aggregate conclusions were not driven by a particular oracle fit. Measured-table replay, in contrast, retained the broad separation between stronger model-based methods and lower-performing baselines using only experimentally recorded responses, although the ordering within the leading group changed. RECOB therefore supports reproducible aggregate comparisons, while exact optimizer rankings should be interpreted in the context of the evaluation protocol.

6 Limitations and Conclusion

In this work, we introduced RECOB, an experimentally grounded benchmark for black-box optimization in chemistry and materials science. Across 14 single-objective and two multi-objective tasks, RECOB revealed clear but task-dependent differences among optimization strategies. HEBO achieved the best aggregate single-objective rank among the evaluated methods, while qNEHVI achieved the best aggregate rank on the two evaluated multi-objective tasks. Optimizer overhead varied by more than orders of magnitude, showing a trade-off between optimization performance and calculation wall-time.

Optimizer rankings were highly consistent across independently retrained learned oracles, and measured-table replay preserved the broad performance hierarchy using physically measured responses. These results provide strong evidence that RECOB can reliably distinguish broad performance differences among black-box optimization algorithms. RECOB remains limited by the coverage and sampling density of its source experiments, does not explicitly model experimental noise or replicate variability, and cannot replace prospective laboratory validation. Within these limits, RECOB enables controlled comparison of optimization algorithms on experimentally derived problems before their evaluation in prospective laboratory campaigns.

AI USE STATEMENT

In this work, we used generative AI tools for assisting with translation, cleaning and reformatting datasets, supporting qualitative and thematic data analysis or interpreting results. We have not used generative AI tools for generating synthetic data sets or conceptual frameworks, designing or providing feedback on research methodology or experiments, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for creating and editing software code, and editing the paper to improve readability. We have reviewed all AI-assisted work. We checked that the datasets reformatted by the LLM were consistent with the original datasets and that the statistical analysis tools provided by the LLM were correct. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work exclusively uses publicly available data. All third-party data were accessed and used in accordance with the licenses specified by their original sources. Detailed information on the data sources and the licenses under which the data were made available and used in this study is provided in Appendix Table 6.

REPRODUCIBILITY STATEMENT

The supplementary repository contains the processed datasets, machine-readable task and constraint specifications, frozen learned-oracle manifests, common initial designs, raw evaluation histories, and analysis scripts used in this study. Dataset provenance and licensing are listed in Appendix A. Oracle fitting, domain mappings, seed construction, optimizer hyperparameters, and software environments are documented in Appendix B. The exact single-objective normalization endpoints and task-level results are reported in Appendix C, and the oracle-retraining and measured-table replay protocols are detailed in Appendix D. Released data, configurations, and initial-design assets are accompanied by SHA-256 checksums so that the inputs to a reproduced run can be verified independently.

REFERENCES

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Thomas Bäck and Hans-Paul Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary computation*, 1(1):1–23, 1993.

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538, 2020.

Daniil Bash, Yongqiang Cai, Vijila Chellappan, Swee Liang Wong, Xu Yang, Pawan Kumar, Jin Da Tan, Anas Abutaha, Jayce JW Cheng, Yee-Fun Lim, et al. Multi-fidelity high-throughput optimization of electrical conductivity in p3ht-cnt composites. *Advanced Functional Materials*, 31 (36):2102606, 2021.

Syrine Belakaria, Aryan Deshwal, and Janardhan Rao Doppa. Max-value entropy search for multi-objective bayesian optimization. *Advances in neural information processing systems*, 32, 2019.

Eric Bradford, Artur M Schweidtmann, and Alexei Lapkin. Efficient multiobjective optimization employing gaussian processes, spectral sampling and a genetic algorithm. *Journal of global optimization*, 71(2):407–438, 2018.

Benjamin Burger, Phillip M Maffettone, Vladimir V Gusev, Catherine M Aitchison, Yang Bai, Xiaoyan Wang, Xiaobo Li, Ben M Alston, Buyi Li, Rob Clowes, et al. A mobile robotic chemist. *Nature*, 583(7815):237–241, 2020.

Adam D Clayton, Edward O Pyzer-Knapp, Mark Purdie, Martin F Jones, Alexandre Barthelme, John Pavey, Nikil Kapur, Thomas W Chamberlain, A John Blacker, and Richard A Bourne. Bayesian self-optimization for telescoped continuous flow synthesis. *Angewandte Chemie International Edition*, 62(3):e202214511, 2023.

Alexander I Cowen-Rivers, Wenlong Lyu, Rasul Tutunov, Zhi Wang, Antoine Grosnit, Ryan Rhys Griffiths, Alexandre Max Maraval, Hao Jianye, Jun Wang, Jan Peters, et al. Hebo: Pushing the limits of sample-efficient hyper-parameter optimisation. *Journal of Artificial Intelligence Research*, 74:1269–1349, 2022.

Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. *Advances in neural information processing systems*, 34:2187–2200, 2021.

Adarsh Dave, Jared Mitchell, Sven Burke, Hongyi Lin, Jay Whitacre, and Venkatasubramanian Viswanathan. Autonomous optimization of non-aqueous li-ion battery electrolytes via robotic experimentation and machine learning coupling. *Nature communications*, 13(1):5454, 2022.

Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, 6(2): 182–197, 2002.

Johannes P Dürholt, Thomas S Asche, Johanna Kleinekorte, Gabriel Mancino-Ball, Benjamin Schiller, Simon Sung, Julian Keupp, Aaron Osburg, Toby Boyne, Ruth Misener, et al. Bofire: Bayesian optimization framework intended for real experiments. *Journal of Machine Learning Research*, 26(204):1–7, 2025.

Nick Erickson, Jonas Mueller, Alexander Shirkov, Hang Zhang, Pedro Larroy, Mu Li, and Alexander Smola. Autogluon-tabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505*, 2020.

Timothy Erps, Michael Foshey, Mina Konaković Luković, Wan Shou, Hanns Hagen Goetzke, Herve Dietsch, Klaus Stoll, Bernhard von Vacano, and Wojciech Matusik. Accelerated discovery of 3d printing materials using data-driven multiobjective optimization. *Science advances*, 7(42): eabf7435, 2021.

Kobi C Felton, Jan G Rittig, and Alexei A Lapkin. Summit: benchmarking machine learning methods for reaction optimisation. *Chemistry-Methods*, 1(2):116–122, 2021.

Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

Nikolaus Hansen, Steffen Finck, Raymond Ros, and Anne Auger. *Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions*. PhD thesis, INRIA, 2009.

Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. Coco: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software*, 36(1):114–144, 2021.

Florian Häse, Matteo Aldeghi, Riley J Hickman, Loïc M Roch, and Alán Aspuru-Guzik. Gryffin: An algorithm for bayesian optimization of categorical variables informed by expert knowledge. *Applied Physics Reviews*, 8(3), 2021a.

Florian Häse, Matteo Aldeghi, Riley J Hickman, Loïc M Roch, Melodie Christensen, Elena Liles, Jason E Hein, and Alán Aspuru-Guzik. Olympus: a benchmarking framework for noisy optimization and experiment planning. *Machine Learning: Science and Technology*, 2(3):035021, 2021b.

Riley Hickman, Priyansh Parakh, Austin Cheng, Qianxiang Ai, Joshua Schrier, Matteo Aldeghi, and Alán Aspuru-Guzik. Olympus, enhanced: benchmarking mixed-parameter and multi-objective optimization in chemistry and materials science. 2023.

Riley J Hickman, Matteo Aldeghi, Florian Häse, and Alán Aspuru-Guzik. Bayesian optimization with known experimental and design constraints for chemistry applications. *Digital Discovery*, 1 (5):732–744, 2022.

Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization*, pp. 507–523. Springer, 2011.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

Luise F Kaven, Artur M Schweidtmann, Jan Keil, Jana Israel, Nadja Wolter, and Alexander Mitsos. Data-driven product-process optimization of n-isopropylacrylamide microgel flow-synthesis. *Chemical Engineering Journal*, 479:147567, 2024.

Stefan Langner, Florian Häse, José Darío Perea, Tobias Stubhan, Jens Hauch, Loïc M Roch, Thomas Heumueller, Alán Aspuru-Guzik, and Christoph J Brabec. Beyond ternary opv: high-throughput experimentation and self-driving laboratories optimize multicomponent systems. *Advanced Materials*, 32(14):1907801, 2020.

Artem I Leonov, Alexander JS Hammer, Slawomir Lach, S Hessam M Mehr, Dario Caramelli, Davide Angelone, Aamir Khan, Steven O'Sullivan, Matthew Craven, Liam Wilbraham, et al. An integrated self-optimizing programmable chemical synthesis and reaction engine. *Nature Communications*, 15(1):1240, 2024.

Qiaohao Liang, Aldair E Gongora, Zekun Ren, Armi Tiihonen, Zhe Liu, Shijing Sun, James R Deneault, Daniil Bash, Flore Mekki-Berrada, Saif A Khan, et al. Benchmarking the performance of bayesian optimization across multiple experimental materials science domains. *npj Computational Materials*, 7(1):188, 2021.

Andre KY Low, Flore Mekki-Berrada, Abhishek Gupta, Aleksandr Ostudin, Jiaxun Xie, Eleonore Vissol-Gaudin, Yee-Fun Lim, Qianxiao Li, Yew Soon Ong, Saif A Khan, et al. Evolution-guided bayesian optimization for constrained multi-objective optimization in self-driving labs. *npj Computational Materials*, 10(1):104, 2024.

Man Luo, Zikai Xie, Huirong Li, Baicheng Zhang, Jiaqi Cao, Yan Huang, Hang Qu, Qing Zhu, Linjiang Chen, Jun Jiang, et al. Physics-informed, dual-objective optimization of high-entropy-alloy nanozymes by a robotic ai chemist. *Matter*, 8(4), 2025.

Benjamin P MacLeod, Fraser GL Parlane, Connor C Rupnow, Kevan E Dettelbach, Michael S Elliott, Thomas D Morrissey, Ted H Haley, Oleksii Proskurin, Michael B Rooney, Nina Taherimakhsousi, et al. A self-driving laboratory advances the pareto front for material properties. *Nature communications*, 13(1):995, 2022.

Art B Owen. Scrambling sobol'and niederreiter–xing points. *Journal of complexity*, 14(4):466–489, 1998.

Florian Pfisterer, Lennart Schneider, Julia Moosbauer, Martin Binder, and Bernd Bischl. Yahpo gym-an efficient multi-objective multi-fidelity benchmark for hyperparameter optimization. In *International Conference on Automated Machine Learning*, pp. 3–1. PMLR, 2022.

Hong Qian, Yiyi Zhu, Xiang Shu, Shuo Liu, Yaolin Wen, Xin An, Huakang Lu, Aimin Zhou, Ke Tang, and Yang Yu. Soo-bench: Benchmarks for evaluating the stability of offline black-box optimization. In *International Conference on Learning Representations*, volume 2025, pp. 81720–81784, 2025.

Mohamadreza Ramezani, Poulomi Nandi, Pablo Antonio De La Fuente-Moreno, and Majid Beidaghi. Brine: a cost-effective electrochemical self-driving laboratory for accelerated discovery of high-performance electrolytes. *Digital Discovery*, 5(1):397–406, 2026.

Artur M Schweidtmann, Adam D Clayton, Nicholas Holmes, Eric Bradford, Richard A Bourne, and Alexei A Lapkin. Machine learning meets continuous flow chemistry: Automated optimization towards the pareto front of multiple objectives. *Chemical Engineering Journal*, 352:277–282, 2018.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1): 148–175, 2015.

Benjamin J Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I Martinez Alvarado, Jacob M Janey, Ryan P Adams, and Abigail G Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021.

Naoto Sugisawa, Hiroki Sugisawa, Yuma Otake, Roman V Krems, Hiroyuki Nakamura, and Shinichiro Fuse. Rapid and mild one-flow synthetic approach to unsymmetrical sulfamides guided by bayesian optimization. *Chemistry-Methods*, 1(11):484–490, 2021.

Brandon Trabucco, Xinyang Geng, Aviral Kumar, and Sergey Levine. Design-bench: Benchmarks for data-driven offline model-based optimization. In *International Conference on Machine Learning*, pp. 21658–21676. PMLR, 2022.

Ben Tu, Axel Gandy, Nikolas Kantas, and Behrang Shafei. Joint entropy search for multi-objective bayesian optimization. *Advances in Neural Information Processing Systems*, 35:9922–9938, 2022.

Florian Wagner, Peter Sagmeister, Clemens E Jusner, Thomas G Tampone, Vidhyadhar Manee, Frederic G Buono, Jason D Williams, and C Oliver Kappe. A slug flow platform with multiple process analytics facilitates flexible reaction optimization. *Advanced Science*, 11(13):2308034, 2024a.

Florian L Wagner, Peter Sagmeister, Thomas G Tampone, Vidhyadhar Manee, Dauzhan Yerkozhanov, Frederic G Buono, Jason D Williams, and C Oliver Kappe. Self-optimizing flow reactions for sustainability: an experimental bayesian optimization study. *ACS Sustainable Chemistry & Engineering*, 12(26):10002–10010, 2024b.

Jia Wu, Xiu-Yun Chen, Hao Zhang, Li-Dong Xiong, Hang Lei, and Si-Hao Deng. Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*, 17(1):26–40, 2019.

Ruyu Yang, Donglai Zhou, Zijin Jia, Yulan Han, Lianyou Tang, Zifan Jiang, Xiaolin Tai, Yuhai Cai, Wenhui Zhong, Yue Lin, et al. Toward practical design of high-entropy catalysts for chlorine evolution reaction via pareto-guided multi-objective bayesian optimization enabled by a robotic ai-chemist. *Angewandte Chemie International Edition*, 65(27):e8794274, 2026.

Ye Yu, Zikai Xie, Man Luo, Adam Redfearn, Arianna Brandolese, Chenxi Sheng, Andrew P Dove, Xiaolong Zhang, Linjiang Chen, and Jun Jiang. Optimizing toward discovery: Ai-driven exploration of lewis acid–base catalysts for pet glycolysis. *Journal of the American Chemical Society*, 148(4):4635–4644, 2026.

Arber Zela, Julien Siems, Lucas Zimmer, Jovita Lukasik, Margret Keuper, and Frank Hutter. Surrogate nas benchmarks: Going beyond the limited search spaces of tabular nas benchmarks. *arXiv preprint arXiv:2008.09777*, 2020.

A Data source and licensing

Table 6: Summary of papers, licenses, and sources

No.	Paper	License	Source
1	An Integrated Self-Optimizing Programmable Chemical Synthesis and Reaction Engine (Leonov et al., 2024)	CC BY 4.0	Nature Communications
2	Bayesian Self-Optimization for Telescoped Continuous Flow Synthesis (Clayton et al., 2023)	CC BY 4.0	Angewandte Chemie
3	Rapid and Mild One-Flow Synthetic Approach to Unsymmetrical Sulfamides Guided by Bayesian Optimization (Sugisawa et al., 2021)	GPL-2.0	GitHub
4	A Mobile Robotic Chemist (Burger et al., 2020)	Apache License	GitHub
5	A Self-Driving Laboratory Advances the Pareto Front for Material Properties (MacLeod et al., 2022)	CC BY 4.0	Nature Communications
6	A Slug Flow Platform with Multiple Process Analytics Facilitates Flexible Reaction Optimization (Wagner et al., 2024a)	CC BY 4.0	Advanced Science
7	Accelerated Discovery of 3D Printing Materials Using Data-Driven Multiobjective Optimization (Erps et al., 2021)	CC BY-NC 4.0	Science Advances
8	Autonomous Optimization of Non-Aqueous Li-Ion Battery Electrolytes via Robotic Experimentation and Machine Learning Coupling (Dave et al., 2022)	CC BY 4.0	Nature Communications
9	Beyond Ternary OPV: High-Throughput Experimentation and Self-Driving Laboratories Optimize Multicomponent Systems (Langner et al., 2020)	CC BY-NC 4.0	Advanced Materials
10	BRINE: A Cost-Effective Electrochemical Self-Driving Laboratory for Accelerated Discovery of High-Performance Electrolytes (Ramezani et al., 2026)	CC BY 3.0	Digital Discovery
11	Data-Driven Product-Process Optimization of N-Isopropylacrylamide Microgel Flow-Synthesis (Kaven et al., 2024)	CC BY 4.0	arXiv
12	Evolution-Guided Bayesian Optimization for Constrained Multi-Objective Optimization in Self-Driving Labs (Low et al., 2024)	CC BY 4.0	npj Computational Materials
13	Machine Learning Meets Continuous Flow Chemistry: Automated Optimization Towards the Pareto Front of Multiple Objectives (Schweidtmann et al., 2018)	CC BY 4.0	Chemical Engineering Journal
14	Multi-Fidelity High-Throughput Optimization of Electrical Conductivity in P3HT-CNT Composites (Bash et al., 2021)	MIT License	GitHub
15	Self-Optimizing Flow Reactions for Sustainability: An Experimental Bayesian Optimization Study (Wagner et al., 2024b)	CC BY 4.0	ACS Sustainable Chemistry & Engineering
16	Toward Practical Design of High-Entropy Catalysts for Chlorine Evolution Reaction via Pareto-Guided Multi-Objective Bayesian Optimization Enabled by a Robotic AI-Chemist (Yang et al., 2026)	MIT License	GitHub
17	Physics-Informed, Dual-Objective Optimization of High-Entropy-Alloy Nanozymes by a Robotic AI Chemist (Luo et al., 2025)	MIT License	GitHub

License abbreviations: CC BY = Creative Commons Attribution; CC BY-NC = Creative Commons Attribution–NonCommercial.

B IMPLEMENTATION DETAILS

B.1 ORACLE SCREENING, FITTING, AND RELEASE ARTIFACTS

Candidate targets were evaluated on five train–test splits with seeds 42–46. Random 80/20 splits were used except for the Li-ion electrolyte and P3HT–CNT tasks, for which rows sharing the same input vector were kept in the same split. The R^2, Pearson correlation, and range-normalized RMSE values in Table 3 are the means over these five splits; the positive-split count records the number of splits with $R^2 > 0$. Screening was performed before the final oracle was fitted, so the formal optimization histories were not used in task admission or model selection.

The same final fitting protocol was used for every admitted target. AutoGluon TabularPredictor was run with `medium_quality`, a 15-s fitting budget, root-mean-squared error as the evaluation metric, two CPU cores, sequential fitting, and candidate model families `RF` and `XT`. The selected `WeightedEnsemble_L2` was then refitted on all valid labeled rows using `refit_full`; the deployed model name is `WeightedEnsemble_L2_FULL` for all 14 tasks. Predictions are point predictions and no stochastic observation-noise term is added. For a multi-objective query, the relevant frozen single-output predictors receive the same validated candidate and their outputs are concatenated in the declared objective order.

Each task release includes a copy of its task configuration, source metadata and constraints, the pre-refit AutoGluon leaderboard, smoke-test predictions, and a `training_manifest.json`. The manifest records the training-row count, input columns, fitting configuration, selected and deployed model names, software versions, and full SHA-256 hashes of the processed data, metadata, and constraint files. Candidate-level validation results are retained in `repeated_split_results.csv` and `repeated_model_screening.csv`; the revalidated task decisions are retained in `latest_model_screening.csv`. These machine-readable files provide the unrounded values underlying Table 3.

B.2 TASK PARAMETERIZATION AND FEASIBLE-DOMAIN MAPPINGS

For the formal comparison, every task was represented by a latent unit cube $[0, 1]^d$ and a deterministic task-specific decoder. This provided every optimizer with the same feasible parameterization, irrespective of whether the optimizer library natively implements categorical, compositional, or joint constraints. Table 7 summarizes the external and latent dimensions. Complete bounds, units, category levels, step sizes, and constraint tolerances are stored in the released `tasks.json` and `multi_tasks.json` files.

Table 7: Task domains and the common feasible parameterizations used in the formal experiments. p is the number of externally submitted variables and d is the dimension of the latent unit cube.

Task	Tier	p	d	Variable types	Decoder and feasibility conditions
AgNP reaction rate	Core	5	5	Continuous	Box scaling followed by validation of $\mathrm{AgNO_3/AA} \geq 0.3$ and $\mathrm{AgNO_3/AA + seed/AgNO_3} \geq 2$.
OPV degradation	Core	5	5	One categorical; four continuous	The first latent coordinate selects PBQ-QF or PTB7-Th; four exponential weights are normalized to a composition summing to one.
Mobile hydrogen evolution	Core	10	11	Step-grid numeric	P10-MIX1 is mapped to a 0.2-mg grid. Nine liquid additions are mapped to 0.25-mL grids with total volume at most 5 mL and L-cysteine at least 0.25 mL; water is derived to complete 5 mL.
BRINE campaign 2	Core	5	6	Continuous	Bounded composition with total dispensed volume at most 330 μL.
Amide yield, reaction 1	Core	5	5	Continuous	Independent box bounds.
Printing compression modulus	Core	6	6	Continuous composition	Bounded simplex with the six ink fractions summing to 100 wt%.
N-benzylation impurity	Core	4	4	Continuous	Independent box bounds.
SNAr E-factor	Core	4	4	Continuous	Independent box bounds.
P3HT–CNT conductivity	Core	5	5	Continuous composition	Bounded simplex with the five component fractions summing to 100 wt%.
Printing toughness	Core	6	6	Continuous composition	Bounded simplex with the six ink fractions summing to 100 wt%.
Li-ion electrolyte conductivity	Conditional	3	3	Continuous	Independent box bounds.
Amide yield, reaction 4	Conditional	5	5	Continuous	Independent box bounds.
Amide yield, reaction 2	Conditional	5	5	Continuous	Independent box bounds.
BRINE campaign 1	Conditional	4	5	Continuous	Bounded composition with total dispensed volume at most 330 μL.
Amide reactions 1/2/4	MO	5	5	Continuous	Intersection of the three child-task box domains; three maximization objectives.
Printing mechanical properties	MO	6	6	Continuous composition	Six ink fractions sum to 100 wt%; compression modulus and toughness are maximized.

For an unconstrained numerical variable, decoding is linear between its stored lower and upper bounds. Categorical values occupy equal intervals of a latent coordinate. For a sum-equality constraint, positive weights $w_j = -\log u_j$ are normalized and then projected onto the bounded simplex. For a sum-at-most constraint over k variables, one latent coordinate controls the used fraction of the available total through $u_0^{1/k}$ and k further coordinates allocate that amount using normalized exponential weights. This adds one latent radial coordinate, explaining $d = p + 1$ for the BRINE tasks. The mobile-chemist task similarly uses one solid-amount coordinate, one total-liquid-volume coordinate, and nine liquid-allocation coordinates, giving 11 latent coordinates for 10 submitted variables. Step-constrained outputs are mapped to their stored grids before deriving water. Normalized exponential allocation is redundant: scaling every positive weight by the same factor leaves the composition unchanged.

B.3 Initial designs, seeds, and repeated proposals

The single-objective release contains 280 frozen initialization files: 14 tasks by 20 trials, each with ten feasible conditions. They were generated using task-aware scrambled Sobol sampling. Within-task duplicates were forbidden both within and across trials, step grids and joint constraints were checked after a CSV round trip, and every OPV trial contains five conditions from each categorical level. For a fixed task and trial, all ten algorithms read the same file in the same row order.

Post-initialization optimizer seeds were constructed as

$$s = \mathrm{uint32}(\mathrm{SHA256}(\texttt{chembench-evaluation-stream-v1}\,|\,\texttt{task}\,|\,\texttt{trial}\,|\,\texttt{algorithm})_{1:4}),$$

where the first four digest bytes were interpreted in big-endian order. Thus, Random Search and Sobol Search start new streams rather than continuing the stream used to construct the common initialization. The multi-objective runner contains 40 frozen Latin-hypercube initialization files (two tasks by 20 trials). It uses an analogous SHA-256 construction with base seed 20260820 and stores the resulting seed in each run summary.

For single-objective runs, candidate identity uses exact categorical strings and numerical values rounded to 12 decimal places. A repeated decoded proposal is resampled, while all time spent on rejected suggestions remains in the recorded suggestion time. After 100 repeated suggestions, a deterministic legal fallback stream derived from the run seed and iteration is used; the optimizer is told the decoded condition that was actually evaluated. For multi-objective runs, repeated proposals are evaluated and remain part of the 110-evaluation budget.

B.4 Single-objective optimizer configurations

All single-objective optimizers received the ten common observations before their first formal suggestion. Minimization responses were sign-flipped inside the optimizer interface, so every method operated under a maximization convention. Unless specified below, library defaults were retained.

B.5 Multi-objective optimizer configurations

The two multi-objective tasks contain only continuous external variables, with the printing compositions decoded onto the same bounded simplex for every method. For qNEHVI, qNParEGO, TSEMO, MESMO, and JES, a double-precision SingleTaskGP with a vector-valued standardized outcome transform was refitted by exact marginal likelihood after every observation.

MESMO and JES acquisition optimization used $q = 1$, five restarts, 128 raw samples, batch limit 1, initialization batch limit 16, and at most 200 iterations. If a posterior path contained fewer than ten detected Pareto points, only that path was redrawn; the requested front width was reduced through 10, 8, 6, 4, 2, and 1 only when necessary, with population 1,024 and 20 search attempts for replacement draws. Numerical acquisition failures used, in order, ten deterministic Sobol starting points, the best finite acquisition value among 2,048 Sobol candidates, and finally the point of maximum finite posterior variance. Every fallback event is recorded in its run summary.

An audit of the 35,200 multi-objective evaluations found four repeated candidates (three from qNParEGO and one from JES), corresponding to 0.011% of the recorded budget. The NSGA-II interface reconstructed its starting population through the latent simplex representation; the maximum

Table 8: Single-objective implementations and fixed hyperparameters.

Method	Implementation used in the formal benchmark
Random Search	Independent NumPy uniform draw in $[0, 1]^d$ at each step.
Sobol Search	SciPy scrambled Sobol sequence with the run-specific seed.
RF/SMAC-like EI	Random forest with 128 trees, `max_features`=0.8, and seed $s + n$ after n observations. EI is evaluated on 4,096 uniform candidates using the across-tree mean and standard deviation and an exploration offset $0.01\,\mathrm{sd}(y)$. This is an in-house SMAC-inspired implementation, not the official SMAC software.
BoTorch LogEI	Double-precision SingleTaskGP with standardized outcomes and exact marginal-likelihood fitting; LogExpectedImprovement with current best response; acquisition optimization with $q = 1$, eight restarts, and 128 raw samples.
BoTorch UCB	The same GP and acquisition optimization as BoTorch LogEI, with UpperConfidenceBound and $\beta = 0.2$.
BoFire LogEI	BoFire single-objective Bayesian strategy (SoboStrategy) on $[0, 1]^d$ with qLogEI and the run-specific seed; all common initial observations are supplied through the BoFire tell interface.
Optuna TPE	TPESampler with the run-specific seed and `n_startup_trials`=0; the ten common observations are inserted as completed trials.
CMA-ES	Initial mean equal to the mean latent coordinate of the ten common conditions; initial step size 0.25, bounds $[0, 1]^d$, population size 5, and run-specific seed.
HEBO	Numerical HEBO design space on $[0, 1]^d$, `rand_sample`=0, and `scramble_seed` equal to the run seed; the common observations are loaded before the first suggestion.
Gryffin	Continuous latent parameters on $[0, 1]^d$; one CPU, `boosted`=false, 64 random samples, 200 epochs, 128 draws, learning rate 0.05, and sampling strategy 1.

Table 9: Multi-objective implementations and fixed hyperparameters.

Method	Implementation used in the formal benchmark
Random Search	Independent uniform draws in the latent unit cube.
Sobol Search	Scrambled Sobol sequence with the algorithm seed and a 1,024-point fast-forward before the first formal suggestion.
NSGA-II	pymoo NSGA-II with population size 10, the common initialization as the starting population, duplicate elimination enabled, and termination at 110 oracle evaluations.
qNEHVI	BoTorch qLogNoisyExpectedHypervolumeImprovement. The acquisition reference is the componentwise minimum of the ten initial responses minus 0.1 times their componentwise range. Baseline pruning is enabled; $q = 1$, five restarts, and 128 raw samples are used.
qNParEGO	BoTorch qLogNParEGO with a new seeded nonnegative scalarization vector normalized to sum to one at every step; baseline pruning, $q = 1$, five restarts, and 128 raw samples.
TSEMO	One joint posterior sample is drawn on 4,096 scrambled-Sobol candidates; a point is sampled from the nondominated set of that posterior draw.
MESMO	Eight posterior Pareto samples with ten requested Pareto points, population 512, and ten search attempts; qLowerBoundMultiObjectiveMaxValueEntropySearch with 64 acquisition samples.
JES	The same Pareto-sampling configuration as MESMO; qLowerBoundMultiObjectiveJointEntropySearch with 64 acquisition samples.

coordinate difference from the stored initialization was 2.13×10^{-14}. All initial oracle responses were identical except for one printing response, for which a tree-model threshold changed the predicted toughness by 43.87 Pa (0.067%). The formal analysis uses the responses actually recorded in each history. As a sensitivity check, we excluded printing trial 5 for all eight algorithms. MESMO and qNEHVI retained mean printing ranks of 1.842 and 2.053, respectively, with qNParEGO next at 2.789.

For multi-objective inference, algorithms were ranked by final HV within each of the 20 paired task–trial blocks. Within each task, the method with the largest mean final HV was compared with each of the other seven methods using a one-sided paired Wilcoxon signed-rank test.

B.6 SOFTWARE AND COMPUTE ENVIRONMENT

The frozen single-objective optimizer environment used Python 3.10 and included BoFire 0.3.1, BoTorch 0.16.1, CMA 3.2.2, HEBO 0.3.6, Optuna 4.9.0, scikit-learn 1.7.2, SciPy 1.12.0, pandas 2.3.3, NumPy 1.24.4, pymoo 0.6.0, and CPU PyTorch 2.13.0. Gryffin was installed from the bundled source revision `fb149d18f9c81b96179cc682c46e05db60bc7229`. The isolated oracle environment used AutoGluon 1.5.0, NumPy 2.2.6, pandas 2.3.3, SciPy 1.15.3, and scikit-learn 1.7.2. The multi-objective bundle fixed NumPy 1.26.4 and pymoo 0.6.1.5 and used BoTorch 0.16.1. The released server bundles include the environment specifications; the single-objective and oracle environments also include full transitive dependency locks.

Each formal task–algorithm–trial run was an independent Slurm job assigned two CPU cores and 8 GB RAM on a dual-socket AMD EPYC 7702 node with 256 GB DDR4. The learned oracle ran in a separate process from the optimizer environment. Single-objective suggestion time was measured around optimizer candidate generation only; oracle evaluation and scheduler queueing were excluded. Multi-objective wall time covers the complete run process but excludes scheduler queueing.

C SINGLE-OBJECTIVE NORMALIZATION AND TASK-LEVEL RESULTS

C.1 FROZEN NORMALIZATION ENDPOINTS

Table 10 gives the exact endpoints used in the normalized-regret definition in the main text. They were computed once from the frozen union of all 2,800 formal single-objective histories (10 algorithms, 14 tasks, and 20 trials), after converting minimization tasks to maximization by multiplying their responses by -1. These are post hoc scale factors only: they were not supplied to an optimizer and could not affect a candidate trajectory. The signs in this table refer to transformed objective values. For example, the OPV best transformed response 0.015795 corresponds to a recorded oracle prediction of approximately -0.015795 in the original minimization units.

Table 10: Per-task pooled-history endpoints under the common maximization convention. The denominator of normalized regret is $z_d^{\text{best}} - z_d^{\text{worst}}$.

Task	Original direction	z_d^{best}	z_d^{worst}
AgNP reaction rate	Max	0.047508	0.004150
OPV degradation	Min	0.015795	−0.709751
Mobile hydrogen evolution	Max	19.982903	0.200546
BRINE campaign 2	Max	25.925722	11.595119
Amide yield, reaction 1	Max	97.753334	21.826666
Printing compression modulus	Max	2.760167	0.233667
N-benzylation impurity	Min	−2.291167	−9.816167
SNAr E-factor	Min	−0.247800	−1.919633
P3HT–CNT conductivity	Max	742.199402	9.429895
Printing toughness	Max	97438.640625	38700.308594
Li-ion electrolyte conductivity	Max	13.827521	4.084336
Amide yield, reaction 4	Max	48.230000	0.803333
Amide yield, reaction 2	Max	98.386665	51.549999
BRINE campaign 1	Max	28.430731	15.822099

C.2 PER-TASK FINAL PERFORMANCE AND TIER SENSITIVITY

Table 11 reports the task-level medians that were ranked to produce Table 4. Exact zeroes indicate that the median gap is zero; this occurs when at least 11 of the 20 trials attain the frozen best-observed response. Positive values below 0.001 are printed as < 0.001; exact zeroes are printed as 0.000. The complete-precision values and all 20 trial-level histories are included in the release.

Table 11: Median final normalized simple regret over 20 trials for each single-objective task. Lower is better. BT-L, RF, BT-U, and RS denote BoTorch LogEI, RF/SMAC-like EI, BoTorch UCB, and Random Search, respectively.

Task	HEBO	BoFire	BT-L	RF	BT-U	CMA	TPE	Gryffin	Sobol	RS
AgNP reaction rate	.065	**.045**	.052	.091	.112	.105	.128	.269	.236	.253
OPV degradation	**.000**	**.000**	**.000**	.047	**.000**	.048	.055	.093	.093	.098
Mobile hydrogen evolution	.183	.131	.351	.212	.292	**.120**	.364	.589	.572	.641
BRINE campaign 2	.055	.059	.063	.052	.057	.057	**.049**	.134	.133	.152
Amide yield, reaction 1	**.003**	.008	.009	.009	.008	.009	.006	.017	.020	.021
Printing compression modulus	**.019**	**.019**	**.019**	.210	**.019**	.217	.216	.229	.277	.275
N-benzylation impurity	$<$**.001**	.004	.007	.014	.004	.019	.030	.045	.061	.067
SNAr E-factor	$<$**.001**	.002	.002	.011	.002	.007	.022	.083	.108	.108
P3HT–CNT conductivity	.060	.202	**.046**	.154	.454	.154	.206	.237	.247	.240
Printing toughness	**.061**	.215	.202	.090	.251	.116	.096	.181	.193	.172
Li-ion electrolyte conductivity	$<$**.001**	.015	.013	.014	.018	.011	.027	.085	.065	.074
Amide yield, reaction 4	**.001**	.001	.007	.068	.006	.028	.100	.293	.269	.330
Amide yield, reaction 2	**.002**	.055	.062	.039	.036	.047	.033	.105	.105	.149
BRINE campaign 1	**.085**	.092	.093	.103	.092	.091	.110	.160	.203	.205

The tier sensitivity analysis repeated the within-task median ranking separately for the ten Core and four Conditional tasks. The resulting mean ranks are shown in Table 12; their Spearman correlation is 0.794.

Table 12: Mean within-task rank by oracle-admission tier. Lower is better.

Algorithm	Core (10 tasks)	Conditional (4 tasks)
HEBO	2.00	1.00
BoFire LogEI	3.85	4.25
BoTorch LogEI	4.10	4.75
RF/SMAC-like EI	4.20	5.00
BoTorch UCB	4.75	3.75
CMA-ES	5.00	3.50
Optuna TPE	5.30	5.75
Gryffin	8.10	9.00
Sobol Search	8.60	8.25
Random Search	9.10	9.75

The single-objective omnibus comparison used a Friedman test with the 14 tasks as blocks and the task-level median final regrets as observations. The follow-up comparisons reported in the main text are two-sided Wilcoxon signed-rank tests between HEBO and each alternative over the same 14 task medians.

C.3 Optimization trajectories

Figures 1–4 show the complete single-objective trajectories underlying the endpoint summaries in the main text. For each task, the left panel is the normalized best-so-far simple regret at evaluation t, and the right panel is the running normalized trapezoidal AUC from evaluation 10 through t. At $t = 10$, the latter is defined by continuity as the regret at evaluation 10. Its value at $t = 110$ therefore equals the normalized regret AUC reported in the main text. Solid lines show means over the 20 trials, and shaded regions show one sample standard deviation around the mean; the lower edge of each band is clipped at zero for display. Lower values are better in all single-objective panels.

Figure 5 reports the corresponding multi-objective trajectories. Hypervolume at each budget was computed using the fixed task-level objective scaling and physical reference point defined in the main text. Running AUC-HV is the trapezoidal average from evaluation 10 through the current evaluation, with its value at evaluation 10 defined by continuity. Solid lines and shaded regions again denote the mean and one sample standard deviation over 20 trials. Higher values are better in all multi-objective panels.

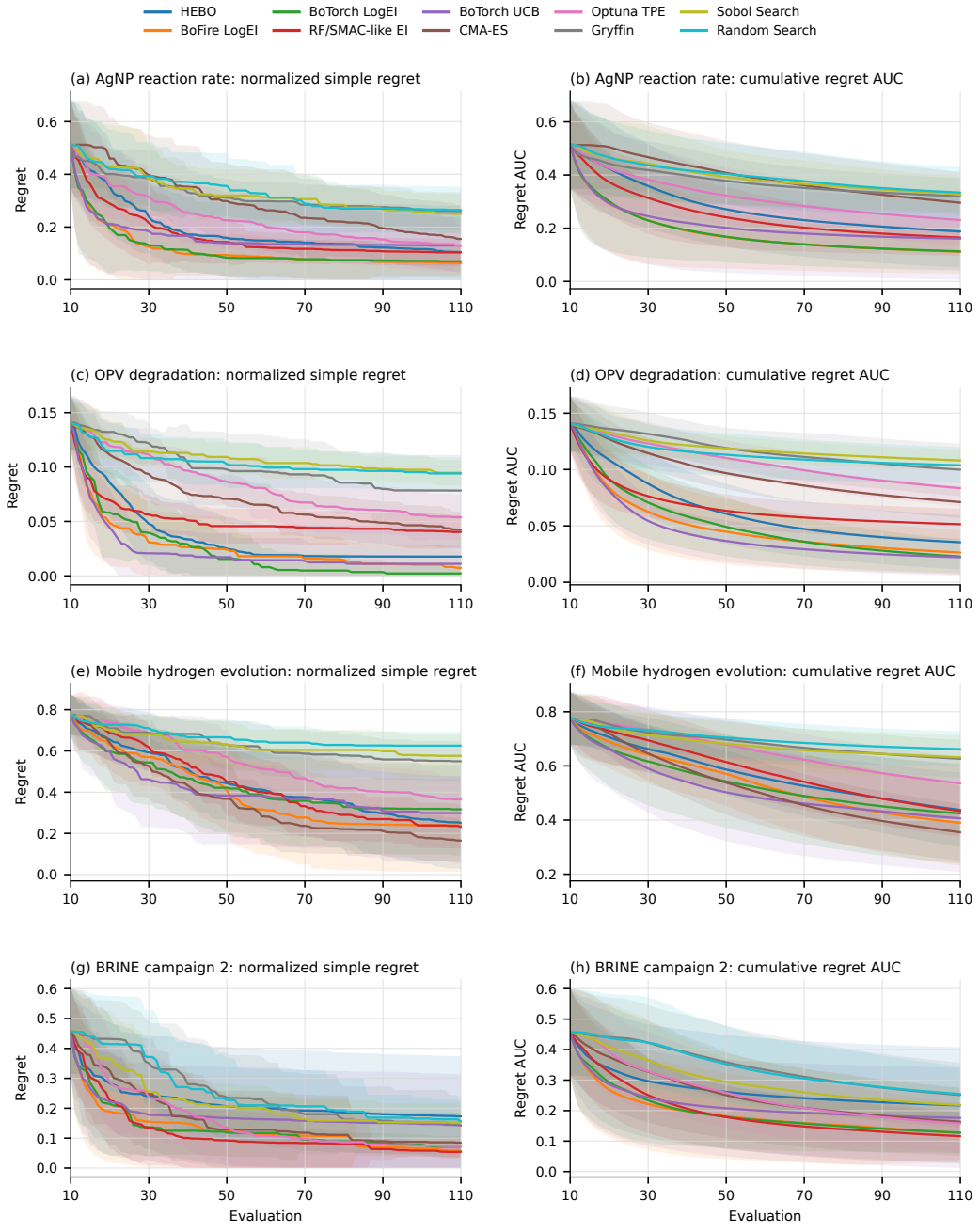


Figure 1: Mean single-objective optimization trajectories (solid lines) with one sample standard deviation across 20 trials (shaded regions) for AgNP reaction rate, OPV degradation, mobile hydrogen evolution, and BRINE campaign 2. Left panels show normalized best-so-far simple regret; right panels show running normalized regret AUC. Lower values are better.

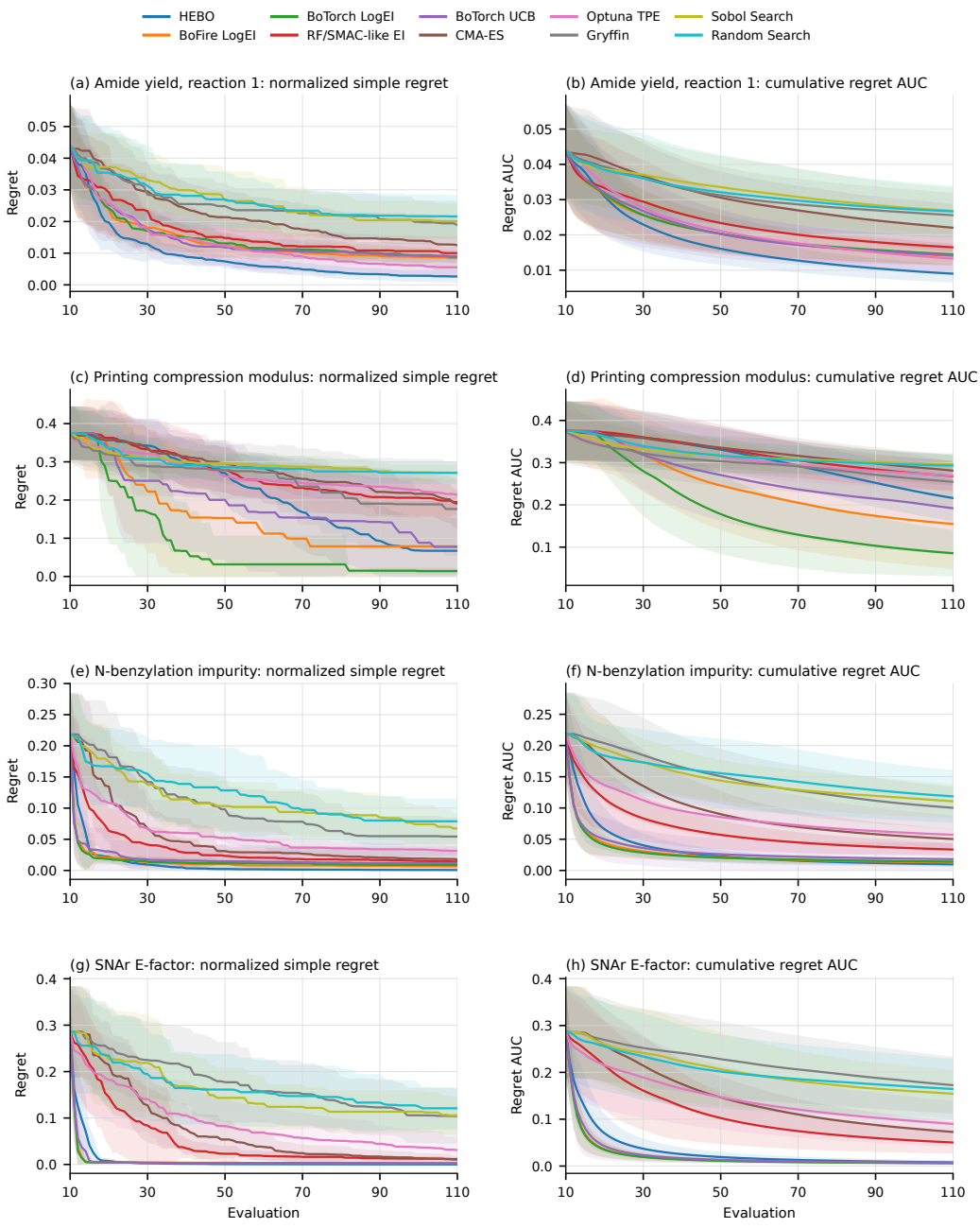


Figure 2: Mean single-objective optimization trajectories (solid lines) with one sample standard deviation across 20 trials (shaded regions) for amide yield (reaction 1), printing compression modulus, N-benzylation impurity, and SNAr E-factor. Left panels show normalized best-so-far simple regret; right panels show running normalized regret AUC. Lower values are better.

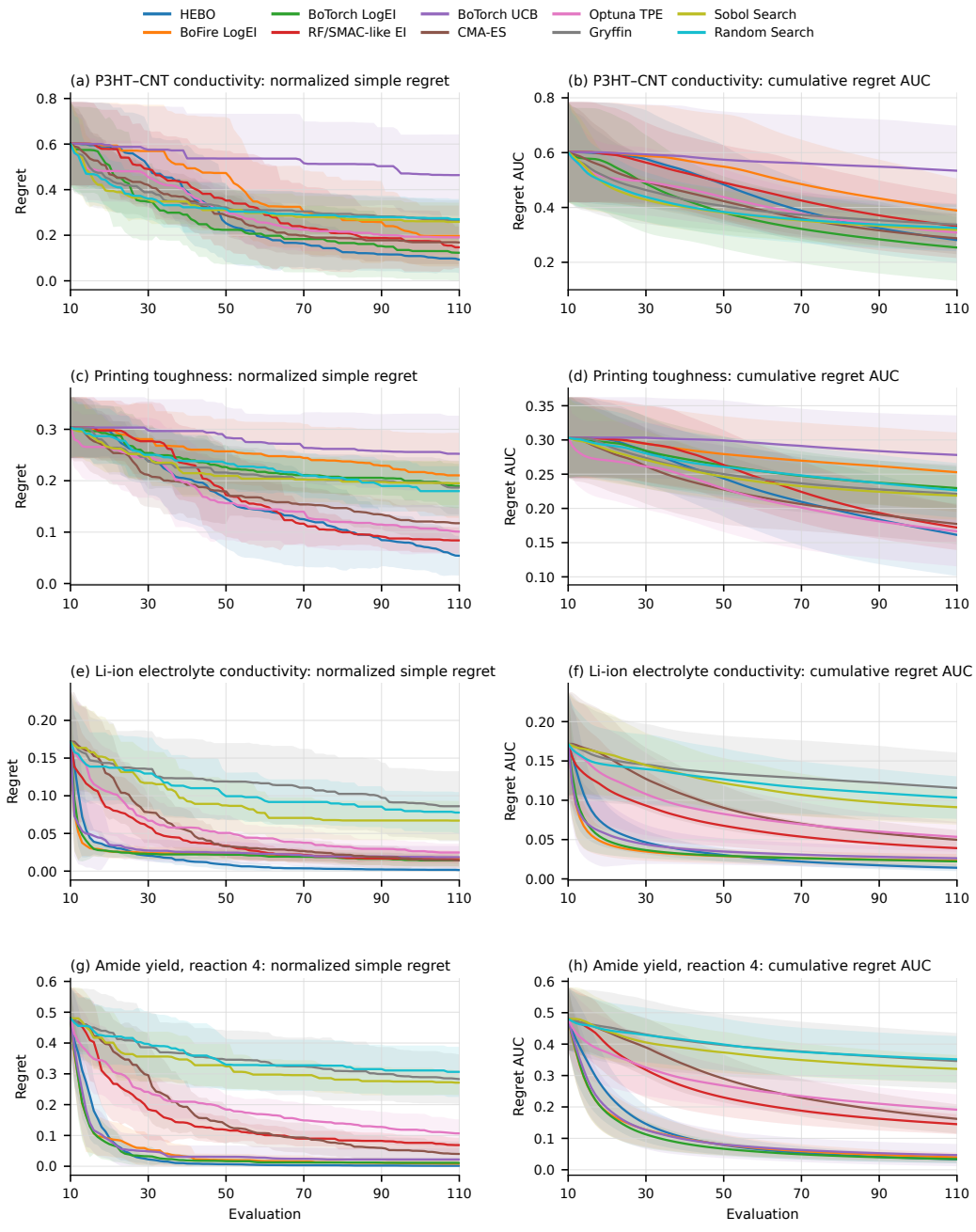


Figure 3: Mean single-objective optimization trajectories (solid lines) with one sample standard deviation across 20 trials (shaded regions) for P3HT–CNT conductivity, printing toughness, Li-ion electrolyte conductivity, and amide yield (reaction 4). Left panels show normalized best-so-far simple regret; right panels show running normalized regret AUC. Lower values are better.

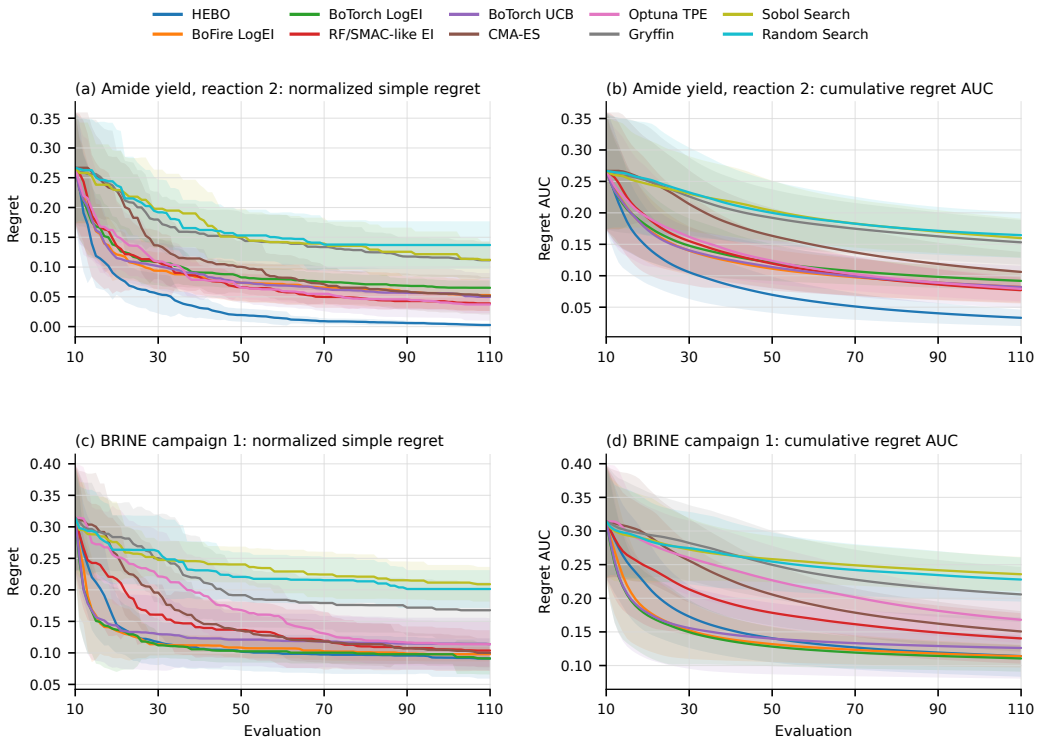


Figure 4: Mean single-objective optimization trajectories (solid lines) with one sample standard deviation across 20 trials (shaded regions) for amide yield (reaction 2) and BRINE campaign 1. Left panels show normalized best-so-far simple regret; right panels show running normalized regret AUC. Lower values are better.

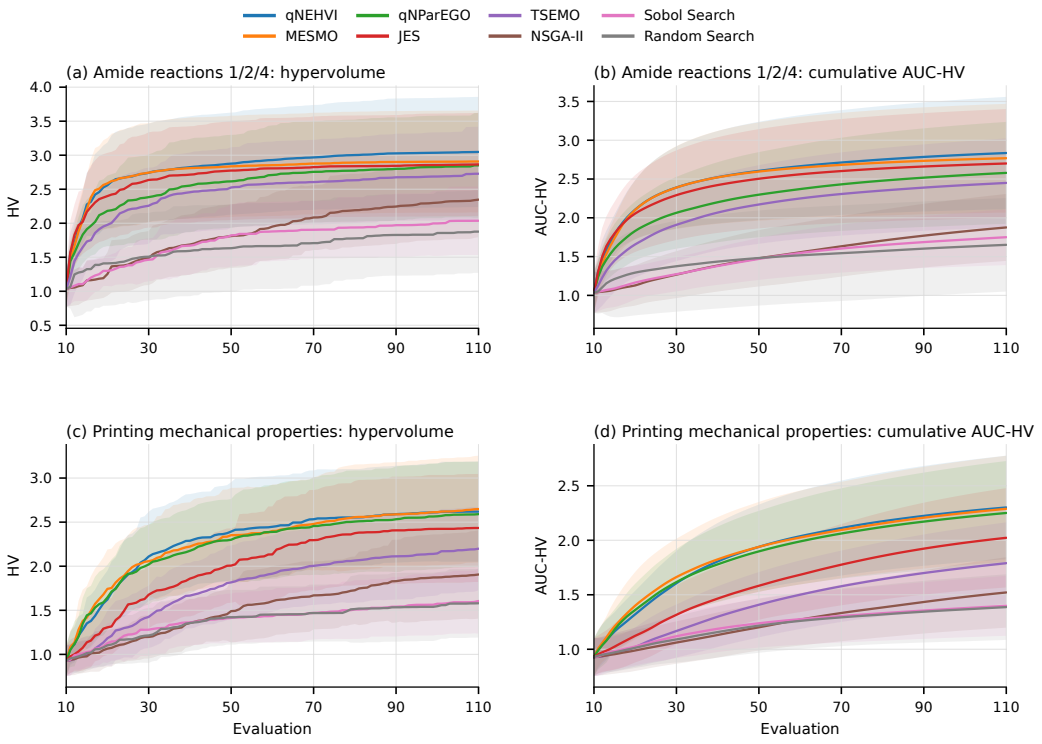


Figure 5: Mean multi-objective optimization trajectories (solid lines) with one sample standard deviation across 20 trials (shaded regions) for the combined amide-reaction task and the printing-mechanical-properties task. Left panels show hypervolume after fixed task-level objective normalization; right panels show running AUC-HV. Higher values are better.

D RELIABILITY-ANALYSIS DETAILS

D.1 INDEPENDENT ORACLE RETRAINING

For seeds 2026, 2027, and 2028, Python's random module, NumPy, `PYTHONHASHSEED`, and the random-forest and extra-trees model seeds were set to that value. Six tasks were then refitted independently with the same AutoGluon protocol described in Appendix B. Each seed-specific model was evaluated with the same 10 algorithms, 20 trials, common initialization, and post-initialization optimizer streams, yielding 3,600 complete runs and 396,000 oracle evaluations. Rankings were calculated within task–trial blocks and then averaged over the six tasks.

Table 13: Agreement between the ten-algorithm rankings obtained from independently retrained oracles.

Seed A	Seed B	Spearman ρ	Kendall τ_b	Top-three overlap
2026	2027	0.964	0.911	3
2026	2028	1.000	1.000	3
2027	2028	0.964	0.911	3

Numerical agreement was assessed on identical response-independent candidates from the Random and Sobol histories. For each task and seed pair, Pearson and Spearman correlations and a mean absolute error normalized by the pooled prediction range were computed. Candidate identity was 1.0 in every comparison. Table 14 aggregates the three seed pairs for each task.

Table 14: Prediction agreement across independent oracle fits. NMAE is mean absolute error divided by the task prediction range.

Task	Mean Pearson	Minimum Spearman	Mean NMAE	Maximum NMAE
Mobile hydrogen evolution	0.9863	0.9760	0.0155	0.0179
AgNP reaction rate	0.9978	0.9833	0.0088	0.0123
BRINE campaign 2	0.9959	0.9951	0.0111	0.0118
Printing compression modulus	0.9976	0.9963	0.0103	0.0115
Amide yield, reaction 1	0.9986	0.9976	0.0086	0.0090
OPV degradation	0.9989	0.9982	0.0046	0.0050

D.2 MEASURED-TABLE REPLAY

Measured replay operates on a finite pool of unique experimental conditions. Rows with identical model inputs were collapsed to one condition by averaging their responses; the replicate count and response standard deviation remain in the pool. Each task–trial pair begins from ten frozen measured rows shared by all algorithms. At each later step, an optimizer proposes a condition in the same external domain used by the continuous oracle. For numerical variable j, the component matching distance between proposal x and measured row x' is

$$\delta_j(x, x') = \frac{|x_j - x'_j|}{u_j - l_j},$$

and for a categorical variable it is the zero–one mismatch indicator. The row distance is the unweighted mean $D(x, x') = p^{-1} \sum_{j=1}^{p} \delta_j(x, x')$. The closest unused row is selected; numerical ties are ordered by the stored opaque row identifier. That row is removed from the pool, and the optimizer is told the matched measured condition and measured response rather than its original proposal. Consequently, no measured row can be evaluated twice in a run.

Matching diagnostics were calculated over the post-initialization evaluations of all ten algorithms and 20 trials. Table 16 reports the median of run-level median distances, the mean run-level 95th percentile, the largest observed distance, and the fraction of matches exceeding 0.25.

At 60 evaluations, 1,200 runs covered all six replay tasks and ten algorithms. At 110 evaluations, 400 runs covered OPV degradation and mobile hydrogen evolution. The agreement statistics in

Table 15: Measured replay pool construction. The 60-evaluation comparison uses all six tasks; the 110-evaluation comparison uses the two pools with at least 110 unique conditions after initialization.

Task	Valid rows	Unique conditions	Collapsed rows	60	110
AgNP reaction rate	144	130	14	Yes	No
OPV degradation	2322	2272	50	Yes	Yes
Mobile hydrogen evolution	595	521	74	Yes	Yes
BRINE campaign 2	121	121	0	Yes	No
Amide yield, reaction 1	117	116	1	Yes	No
Printing compression modulus	143	143	0	Yes	No

Table 16: Measured-table replay matching distances by task and total budget.

Task	Budget	Median run median	Mean run P95	Maximum	Fraction > 0.25
AgNP reaction rate	60	0.141	0.239	0.452	0.068
Amide yield, reaction 1	60	0.231	0.338	0.510	0.375
BRINE campaign 2	60	0.041	0.084	0.183	0.000
Mobile hydrogen evolution	60	0.068	0.103	0.166	0.000
OPV degradation	60	0.016	0.056	0.136	0.000
Printing compression modulus	60	0.109	0.191	0.301	0.024
Mobile hydrogen evolution	110	0.080	0.122	0.196	0.000
OPV degradation	110	0.020	0.074	0.162	0.000

Table 17 compare the three-seed oracle aggregate with measured replay at the same budget. The two rows have different task sets and therefore should be interpreted as separate comparisons rather than as a controlled estimate of the effect of budget alone.

Table 17: Rank agreement between learned-oracle optimization and measured-table replay.

Budget	Tasks	Spearman ρ	Kendall τ_b	Top-three overlap
60	6	0.758	0.556	2
110	2	0.455	0.333	2