PHEBO: PREDICTIVE FEEDBACK-DRIVEN HYPOTHESIS EVOLUTION FOR BAYESIAN OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Bayesian optimization (BO) offers a sample-efficient framework for scientific optimization with costly evaluations. Large language models (LLMs) can enrich this framework by proposing domain-informed hypotheses, but how to assess their reliability, translate them into numerical guidance, and revise them as evidence accumulates remains a key challenge. This paper introduces PHEBO, an LLM-enhanced BO framework in which predictive feedback drives the evolution of hypotheses population. These hypotheses are encoded in conditional Gaussian process models that quantify their numerical effects. Prospective forecast evaluation updates hypothesis credits, which govern acquisition and hypothesis selection. At scheduled interventions, the LLM uses accumulated observations and credits to revise existing hypotheses, recombine them, and propose new ones, closing the loop between domain reasoning and numerical optimization. Rather than relying on LLMs for in-context numerical prediction or self-reported confidence, PHEBO focuses the LLM on identifying and revising qualitative trends and relationships, guided by empirical predictive feedback. Across four offline scientific optimization tasks spanning analytical chemistry, materials design, and organic photovoltaics, PHEBO achieves better overall performance against GP-based and LLM-enhanced BO baselines. Beyond numerical optimization, the resulting domain beliefs, refined through observation and reasoning, have the potential to inform scientific discovery and support other LLM-enhanced workflows. Code is available at `https://anonymous.4open.science/r/PHEBO-28C1`

1 INTRODUCTION

Scientific discovery often requires selecting informative experiments under a limited evaluation budget. Although domain knowledge can guide this process, it is commonly expressed through qualitative and context-dependent relations: a variable may be beneficial, an interaction may be important, or a favorable operating regime may exist. Bayesian optimization (BO) provides an efficient framework for optimizing expensive objectives through surrogate modeling and sequential evaluation (Jones et al., 1998), but conventional numerical surrogates do not naturally incorporate knowledge stated in natural language or inferred from heterogeneous observations. Large language models (LLMs) offer a promising mechanism for interpreting these task descriptions and observations, motivating their use in surrogate prediction and experiment generation (Liu et al., 2024; Caldas Ramos et al., 2026). The key question is how to turn this contextual understanding into reliable numerical guidance for sequential scientific decision-making.

What role could an LLM play in BO? Its prior knowledge and contextual reasoning can identify trends and formulate candidate mechanisms, but recognizing a relationship and quantifying it are distinct tasks. In two regression case studies, Riegler et al. (2025) found that LLMs correctly identified association directions, illustrating their ability to recover qualitative relationships. However, Liu et al. (2025) identified architectural and training-related precision barriers in standard softmax Transformers on controlled least-squares tasks, and BO-ICL reports uncertainty-calibration challenges in its LLM surrogate (Caldas Ramos et al., 2026). These limitations motivate a division of labor: the LLM proposes qualitative trends and relationships from prior knowledge and observations, while explicit statistical inference quantifies their effects.

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.

How much influence should a hypothesis have on optimization? The value of LLM guidance depends partly on how strongly its hypotheses influence the search. LGBO converts LLM-generated point or region preferences into Gaussian process (GP) mean shifts, scaling self-reported confidence by posterior uncertainty (Yuan et al., 2026). Although this controls the magnitude of guidance, self-reported confidence need not reflect empirical reliability: Xiong et al. (2024) found that LLMs tend to be overconfident across several reasoning tasks. Following prequential forecast assessment (Dawid, 1984), we argue that a hypothesis should earn influence by anticipating improvement on unseen outcomes. This criterion aligns conceptually with expected improvement (EI), which measures the expected gain over the current incumbent (Jones et al., 1998).

How should LLM beliefs evolve as observations accumulate? Like a BO surrogate posterior, LLM beliefs should evolve with accumulating evidence. History-aware prompting enables iterative guidance in LGBO and BORA (Yuan et al., 2026; Cissé et al., 2025), while CAKE uses LLM-driven mutation and crossover to evolve GP kernels, evaluating population fitness with the Bayesian information criterion (BIC) and combining it with EI for kernel selection (Suwandi et al., 2025). These approaches motivate retaining history across optimization rounds while tracking the utility of competing beliefs. New evidence can then preserve supported knowledge while prompting reconsideration of weakened assumptions.

To this end, we introduce PHEBO, an LLM-enhanced BO framework in which predictive feedback drives hypothesis evolution. The LLM uses prior knowledge and observations to propose and revise scientific hypotheses, while conditional statistical models determine their numerical effects. Predictive credit uses subsequent experimental evidence to control both the hypotheses' influence on optimization and the evolution of the belief population. This closes the loop: the LLM refines domain beliefs, while BO adapts how they guide numerical decisions.

Our contributions connect the use and evolution of LLM-generated knowledge through predictive assessment as follows:

1. We introduce an *executable hypothesis interface* that translates LLM-generated hypotheses into numerical models, enabling their quantitative evaluation.
2. We develop a *credit-guided belief evolution mechanism* that scores hypotheses against subsequent observations. The resulting credits guide hypothesis selection and mixture-based acquisition, while empirical feedback drives their revision and recombination.
3. We evaluate PHEBO on *four scientific optimization tasks*, demonstrating better overall performance than baselines within limited experimental budgets, while also providing evidence for the transferability and potential scientific value of the constructed beliefs.

2 Related Work

2.1 LLM guidance and numerical inference

LLM-enhanced Bayesian optimization (BO) differs in how language informs numerical decisions. LLAMBO uses in-context learning for probabilistic prediction, warmstarting, and objective-conditioned candidate sampling (Liu et al., 2024); BO-ICL and LLANA likewise explore language-based surrogates (Caldas Ramos et al., 2026; Chen et al., 2024). However, contextual prediction does not ensure numerical reliability: controlled least-squares studies identify precision limitations in standard Transformers (Liu et al., 2025), and LLM prior-elicitation experiments recover association directions while struggling to calibrate prior widths (Riegler et al., 2025). Rather than predicting objective values, PHEBO uses the LLM to propose trends and candidate mechanisms and GP inference to quantify their effects.

Beyond surrogate prediction, LLMs can guide the next search action: generating candidate solutions from evaluated examples (Yang et al., 2024; Agarwal et al., 2025), proposing candidates alongside a statistical optimizer (Yin et al., 2024; Mahammadli & Ertekin, 2025), or selecting an acquisition function (Ngo et al., 2026). Here, language outputs specify candidates or search strategies. PHEBO instead maps LLM-expressed trends and mechanisms to GP experts, allowing the optimizer to derive actions from their guidance.

2.2 LLM KNOWLEDGE RETENTION AND EVOLUTION

Knowledge-guided BO includes fixed optimizer priors, hypothesis regions, and function beliefs (Hvarfner et al., 2022; Cissé et al., 2024; Hvarfner et al., 2024). BORA's experiments supply HypBO (Cissé et al., 2024) and ColaBO (Hvarfner et al., 2024) with LLM-generated inputs fixed before optimization (Cissé et al., 2025). More adaptive interfaces retain linguistic history: LGBO uses previous reasoning to refresh point or region preferences, shifting the GP mean without changing covariance (Yuan et al., 2026); BORA conditions new hypotheses on past comments and observations (Cissé et al., 2025). These methods already remember and revise guidance. These methods already remember and revise guidance; PHEBO instead retains beliefs as credited predictive experts with parent–offspring lineages, enabling continued assessment and genetic revision.

Methods also differ in how they assess guidance and determine its influence on optimization. LLINBO studies participation decay and surrogate-based checks on LLM suggestions (Chang et al., 2025), whereas CAKE uses LLM crossover and mutation to evolve GP kernels, ranks fitness by the Bayesian information criterion, and combines model fit with acquisition utility (Suwandi et al., 2025). PHEBO similarly uses genetic operations but recombines mechanism claims rather than covariance structures. Its selection pressure combines inherited credit with subsequent predictive evidence, linking belief participation and influence to forecasts issued before new outcomes. Thus, both belief content and assessed utility evolve with optimization.

3 PRELIMINARIES

We review Bayesian optimization and prequential forecast evaluation, which provide the optimization framework and the basis for assessing predictive reliability.

3.1 BAYESIAN OPTIMIZATION

Consider a black-box objective $f : \mathcal{X} \to \mathbb{R}$ on a bounded continuous domain $\mathcal{X} \subset \mathbb{R}^D$, whose evaluations are expensive, potentially noisy, and do not provide gradients. Let $p_1, \ldots, p_D$ denote the task parameters and $\mathbf{x} = (x_1, \ldots, x_D) \in \mathcal{X}$ an input configuration, where x_j is the value assigned to p_j. Bayesian optimization (BO) uses a probabilistic surrogate to guide the search for high objective values under a limited evaluation budget. Given initial observations $\mathcal{D}_0$, at round t the surrogate is conditioned on $\mathcal{D}_{t-1}$ and an acquisition function α_t selects the next query point $\mathbf{x}_t$, at which a noisy observation y_t is obtained:

$$\mathbf{x}_t \in \arg\max_{\mathbf{x}\in\mathcal{X}} \alpha_t(\mathbf{x}), \qquad y_t = f(\mathbf{x}_t)+\epsilon_t, \qquad \mathcal{D}_t = \mathcal{D}_{t-1}\cup\{(\mathbf{x}_t, y_t)\}, \qquad \epsilon_t \sim \mathcal{N}(0, \sigma^2_{n,t}). \quad (1)$$

We use index 0 for the baseline Gaussian process (GP), with prior mean $m_{0,t}$, covariance kernel k_t, and independent zero-mean Gaussian observation noise of variance $\sigma^2_{n,t}$ (Rasmussen & Williams, 2006). Conditioning on $\mathcal{D}_{t-1}$ with these parameters fixed yields

$$q^{\mathrm{obs}}_{0,t}(\cdot \mid \mathbf{x}) = \mathcal{N}\big(\mu_{0,t}(\mathbf{x}), v_{0,t}(\mathbf{x}) + \sigma^2_{n,t}\big)\,, \quad (2)$$

where $\mu_{0,t}$ and $v_{0,t}$ are the posterior mean and variance of the latent function value.

3.2 PREQUENTIAL FORECAST EVALUATION

The prequential principle evaluates predictive distributions on outcomes observed after the forecasts are issued (Dawid, 1984). For a predictor indexed by i, let $q^{\mathrm{obs}}_{i,t}(\cdot \mid \mathbf{x}_t)$ denote its forecast issued before y_t is revealed. Its logarithmic score is

$$s_{i,t} = \log q^{\mathrm{obs}}_{i,t}(y_t \mid \mathbf{x}_t). \quad (3)$$

Under standard integrability conditions, this score is strictly proper: its expectation is maximized by reporting the true predictive distribution (Gneiting & Raftery, 2007). Comparing scores on subsequent observations provides a basis for assessing predictors formed from earlier data.

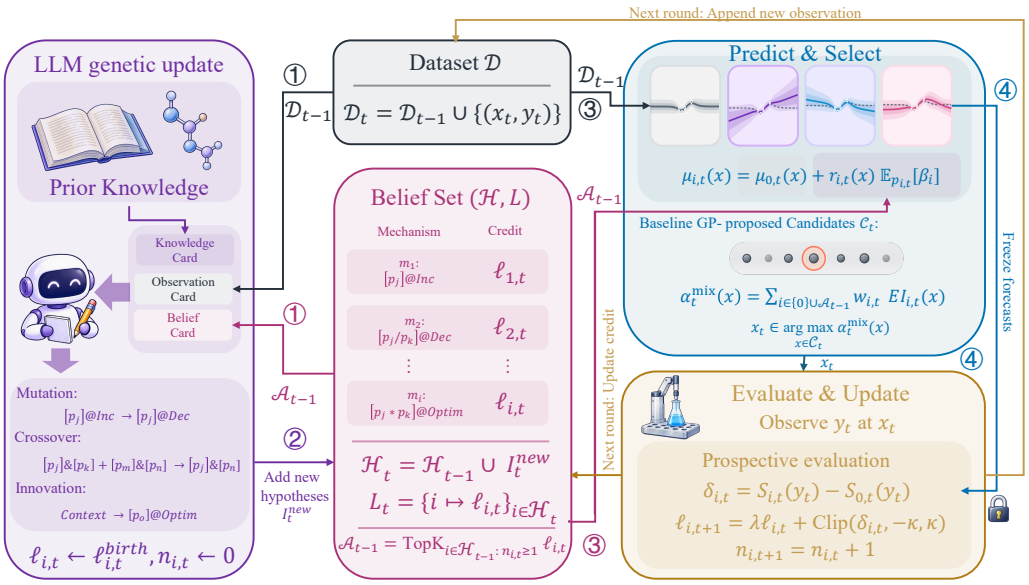

Figure 1: Overview of PHEBO. ① At scheduled interventions, the LLM uses domain knowledge, accumulated observations, and current active beliefs with credits to propose new hypotheses through mutation, crossover, or innovation. ② Accepted hypotheses enter the belief archive with inherited credit. ③ Credits select and weight conditional-GP experts in a mixture-EI acquisition over baseline-GP candidates. ④ Forecasts are frozen before evaluation, and the new outcome updates every archived hypothesis's credit relative to the baseline.

4 METHOD

PHEBO integrates LLM-based hypothesis evolution with Bayesian optimization through predictive feedback. At scheduled interventions, the LLM updates the belief set using domain context, accumulated observations, and belief credits.. Each belief is then instantiated as a GP-based expert whose credit is initialized from its lineage and subsequently updated based on predictive performance relative to a baseline GP. These credits determine active-set membership and predictive-mixture weights. The next evaluation maximizes expected improvement under this mixture over candidates proposed by the baseline GP, thereby connecting belief evolution to experiment selection. Figure 1 summarizes this closed loop.

4.1 EXECUTABLE DOMAIN HYPOTHESES

Hypothesis representation. We represent each hypothesis as a structured mechanism m_i, indexed by $i \geq 1$, together with a natural-language explanation; m_0 is reserved for the baseline GP. Each term specifies a feature—a parameter p_j, a product $p_j p_k$, or an ordered ratio p_j/p_k ($j \neq k$)—and pairs it with an increasing (Inc), decreasing (Dec), or optimum (Opt) preference. For example, $[p_j]$`@Inc` expresses the LLM's hypothesis that larger values of p_j are associated with higher objective values. The structured mechanism defines a mechanism basis h_i, while the explanation provides context for interpretation and revision.

From claims to preference functions. For a configuration $\mathbf{x}$, let $z_{i,r}(\mathbf{x})$ denote the value of the rth feature of m_i, $r = 1, \ldots, J_i$, obtained by substituting x_j for each parameter p_j in its expression. TThe paired preference defines a bounded function $\psi_{i,r}(\mathbf{x}) \in [\varepsilon_\psi, 1]$. Appendix A.1 specifies its construction for `@Inc`, `@Dec`, and `@Opt`. We construct the mechanism basis by taking the geometric mean of these term-level preferences and rescaling it:

$$h_i(\mathbf{x}) = 2\left[\prod_{r=1}^{J_i} \psi_{i,r}(\mathbf{x})\right]^{1/J_i} - 1. \tag{4}$$

Larger values of $h_i(\mathbf{x})$ indicate more favorable configurations under mechanism m_i. This basis supplies the mechanism-specific mean component of the GP expert introduced next.

4.2 CONDITIONAL GP EXPERTS

Mechanism-informed mean. Given the basis h_i compiled from mechanism m_i, we construct an expert that combines the mechanism's preference with a GP model. Following GP regression with explicit basis functions (Rasmussen & Williams, 2006), we use the mechanism basis h_i as an explicit mean component with an uncertain coefficient. At round t, we fit a baseline GP to $\mathcal{D}_{t-1}$ and hold its fitted prior mean $\mu_{0,t}^{\text{prior}}$, kernel k_t, and observation-noise variance $\sigma_{n,t}^2$ fixed across experts. The expert associated with mechanism m_i is

$$f_{i,t}(\mathbf{x}) = \mu_{0,t}^{\text{prior}}(\mathbf{x}) + g_t(\mathbf{x}) + \beta_i h_i(\mathbf{x}), \quad g_t \sim \mathcal{GP}(0, k_t), \quad \beta_i \sim \mathcal{N}(0, \tau_t^2)\big|_{[0,\infty)}. \tag{5}$$

Here g_t models residual variation and is independent of β_i a priori; $\tau_t > 0$ controls the coefficient prior scale. The constraint $\beta_i \geq 0$ prevents the mechanism contribution from reversing the ordering encoded by h_i. Appendix A.2 specifies the implementation conventions.

Conditional inference. Conditioning on $\mathcal{D}_{t-1}$ with h_i and the shared GP parameters fixed, yields the truncated-normal coefficient posterior $p_{i,t}(\beta) = \mathcal{N}(\beta; b_{i,t}, \nu_{i,t})\big|_{[0,\infty)}$, where $b_{i,t}$ and $\nu_{i,t}$ are the mean and variance of the Gaussian before truncation.

Let X collect the observed inputs, and define $H_i = h_i(X)$, $C_t = k_t(X, X) + \sigma_{n,t}^2 I$, and $k_{t,\mathbf{x}} = k_t(X, \mathbf{x})$. For a fixed coefficient β, the conditional latent prediction is Gaussian with mean $\mu_{0,t}(\mathbf{x}) + \beta r_{i,t}(\mathbf{x})$ and variance $v_{0,t}(\mathbf{x})$, where the baseline moments are defined in Equation (2) and

$$r_{i,t}(\mathbf{x}) = h_i(\mathbf{x}) - k_{t,\mathbf{x}}^\top C_t^{-1} H_i. \tag{6}$$

The subtraction accounts for the change in the GP residual after assigning the contribution βH_i to the mechanism at the observed inputs. Thus, the posterior mean correction is $\beta r_{i,t}(\mathbf{x})$, rather than $\beta h_i(\mathbf{x})$, and can have either sign despite $\beta \geq 0$. Appendix A.3 derives the coefficient posterior and this residual correction.

Marginal predictive distribution. We marginalize over the coefficient posterior to incorporate uncertainty in the mechanism's contribution:

$$q_{i,t}^{\text{lat}}(u \mid \mathbf{x}) = \int_0^\infty \mathcal{N}(u; \mu_{0,t}(\mathbf{x}) + \beta r_{i,t}(\mathbf{x}), v_{0,t}(\mathbf{x}))\, p_{i,t}(\beta)\, d\beta. \tag{7}$$

Although the prediction conditional on β is Gaussian, the marginal distribution need not be. When $r_{i,t}(\mathbf{x}) = 0$, it reduces to the baseline latent prediction at $\mathbf{x}$. The observation density $q_{i,t}^{\text{obs}}$ is obtained by replacing $v_{0,t}(\mathbf{x})$ inside the integral with $v_{0,t}(\mathbf{x}) + \sigma_{n,t}^2$.

4.3 PROSPECTIVE CREDIT AND ACTIVE PARTICIPATION

Prospective evaluation. We assess mechanisms through forecasts issued before the evaluated outcomes are observed. Let $\mathcal{H}_t$ index all mechanisms accepted before evaluation t. At the selected input $\mathbf{x}_t$, we freeze the observation forecasts of the baseline and every archived mechanism before observing y_t. To emphasize prediction of improvement, we score these forecasts relative to the incumbent $a_t = \max_{(\mathbf{x},y)\in\mathcal{D}_{t-1}} y$. Following the censored-likelihood construction of Diks et al. (2011), we retain the exact outcome when it exceeds a_t and otherwise record only non-improvement:

$$S_{i,t}(u) = \begin{cases} \log q_{i,t}^{\text{obs}}(u \mid \mathbf{x}_t), & u > a_t, \\ \log Q_{i,t}^{\text{obs}}(a_t \mid \mathbf{x}_t), & u \leq a_t. \end{cases} \tag{8}$$

Here $Q_{i,t}^{\text{obs}}$ is the observation CDF. Thus, non-improving outcomes are scored by their total predicted probability, irrespective of their magnitudes. Appendix A.4 derives the density and CDF.

Discounted predictive credit. Let $\ell_{i,t}$ denote the credit of mechanism m_i before evaluation t. After observing y_t, we update every archived mechanism relative to the baseline:

$$\delta_{i,t} = S_{i,t}(y_t) - S_{0,t}(y_t), \qquad \ell_{i,t+1} = \lambda \ell_{i,t} + \text{clip}(\delta_{i,t}, -\kappa, \kappa), \tag{9}$$

where $0 < \lambda < 1$ and $\kappa > 0$. Discounting emphasizes recent forecasts (Raftery et al., 2010), while clipping limits the contribution of a single outcome. A positive $\delta_{i,t}$ means that the expert assigned greater density to the realized improving outcome, or greater probability to non-improvement, than the baseline. Appendix A.5 establishes its bound and distinguishes the propriety of censored scoring from the properties of the clipped credit update.

Active participation. Let $n_{i,t}$ count the previously scored forecasts of m_i, with $n_{i,t+1} = n_{i,t} + 1$ after each evaluation. Only mechanisms with at least one scored forecast are eligible for the active set:

$$\mathcal{A}_{t-1} = \mathrm{TopK}_{i \in \mathcal{H}_{t-1}:\, n_{i,t} \geq 1} \ell_{i,t}. \tag{10}$$

Here TopK returns the K highest-credit eligible indices, or all eligible indices if fewer than K exist. Ties are resolved as specified in Appendix A.9. The active mechanisms participate in the predictive mixture; the baseline is included separately and does not occupy these K slots. Unscored mechanisms are pending, and scored mechanisms outside $\mathcal{A}_{t-1}$ are dormant. Both remain archived and continue to receive scores. A new mechanism therefore becomes eligible only after its first prospective score, and the first decision uses the baseline alone.

4.4 Hypothesis evolution and credit inheritance

Hypothesis generation. Given the domain context c and initial observations $\mathcal{D}_0$, the LLM proposes seed hypotheses. At scheduled interventions, it receives the domain context c, available observations $\mathcal{D}_{t-1}$, and active mechanisms $\mathcal{A}_{t-1}$ with their credits. Drawing on genetic algorithms (Holland, 1992), it proposes new hypotheses through mutation of one parent, crossover of terms from two parents, or parentless innovation, subject to the mechanism grammar. Each accepted hypothesis enters the archive as a new pending mechanism. Appendix A.9 specifies proposal validation and execution conventions.

Birth credit and subsequent evidence. Let $\mathcal{P}_i$ denote the parent set of m_i: it is empty for seeds and parentless innovations, contains one parent for mutation, and two for crossover. Its birth credit is

$$\ell_i^{\mathrm{birth}} = 0 \text{ if } \mathcal{P}_i = \varnothing; \quad \text{else } \ell_i^{\mathrm{birth}} = |\mathcal{P}_i|^{-1} \sum_{p \in \mathcal{P}_i} \ell_{p,t_i^{\mathrm{birth}}} \tag{11}$$

We initialize $\ell_{i,t_i^{\mathrm{birth}}} = \ell_i^{\mathrm{birth}}$ and $n_{i,t_i^{\mathrm{birth}}} = 0$. Thus, historical observations may inform the new expert's coefficient posterior, whereas credit increments use only post-birth forecasts; inherited credit decays under Equation (9), as shown in Appendix A.5.

4.5 Experiment selection under the predictive mixture

Credit-weighted prediction. Predictive credit determines how strongly each active expert influences experiment selection. For $M_t = |\mathcal{A}_{t-1}| > 0$, let

$$Z_t = M_t + \sum_{i \in \mathcal{A}_{t-1}} e^{\ell_{i,t}}, \qquad w_{0,t} = \frac{M_t}{Z_t}, \qquad w_{i,t} = \frac{e^{\ell_{i,t}}}{Z_t} \quad (i \in \mathcal{A}_{t-1}). \tag{12}$$

The factor M_t keeps the mechanism group's reference mass independent of its size. These weights define the latent predictive mixture

$$q_t^{\mathrm{mix}}(u \mid \mathbf{x}) = \sum_{i \in \{0\} \cup \mathcal{A}_{t-1}} w_{i,t} q_{i,t}^{\mathrm{lat}}(u \mid \mathbf{x}). \tag{13}$$

Mixture expected improvement. We select experiments by the expected improvement(EI) above the incumbent a_t (Jones et al., 1998). With $(z)_+ = \max\{z, 0\}$, define each expert's EI and, by linearity, the mixture acquisition function as

$$\mathrm{EI}_{i,t}(\mathbf{x}) = \mathbb{E}_{F \sim q_{i,t}^{\mathrm{lat}}(\cdot \mid \mathbf{x})}[(F - a_t)_+], \qquad \alpha_t^{\mathrm{mix}}(\mathbf{x}) = \sum_{i \in \{0\} \cup \mathcal{A}_{t-1}} w_{i,t}\, \mathrm{EI}_{i,t}(\mathbf{x}). \tag{14}$$

Thus, credit converts predictive evidence and inherited trust into a weighted improvement criterion. Appendices A.6 and A.7 give its computation and derivation. To highlight the value of the mixture

acquisition, we use EI under the baseline GP landscape to select at most B peak locations x, which form the candidates set $\mathcal{C}_t$(Appendix A.8). The experiment point for this round is selected by $\mathbf{x}_t = \arg\max_{\mathbf{x}\in\mathcal{C}_t} \alpha_t^{\text{mix}}(\mathbf{x})$.

Closed-loop update. Before observing y_t, we freeze the forecasts of the baseline and all archived mechanisms at $\mathbf{x}_t$. The new observation updates their relative credits through Equation (9) and augments $\mathcal{D}_{t-1}$ to form $\mathcal{D}_t$. These updates determine subsequent active participation and mixture weights, while scheduled LLM interventions use the accumulated observations and credits to revise hypotheses. The resulting loop links hypothesis evolution, experiment selection, and prospective evaluation. Algorithm 1 summarizes the procedure; Appendix A.9 provides execution details.

5 Experiments

We evaluate PHEBO in offline scientific optimization settings using publicly available datasets collected from real wet-lab experiments. We treat each dataset as a task. Each task consists of a finite set of design–response pairs, from which we identify the offline optimum. Although the offline optimum may differ from the true global optimum of the underlying physical system, it provides a consistent reference for comparing optimization algorithms across tasks.

5.1 Experimental settings

Baselines. We compare PHEBO with **GPBO**, **LLAMBO** (Liu et al., 2024), **CAKE** (Suwandi et al., 2025), and **LGBO** (Yuan et al., 2026). **GPBO** uses a Gaussian process surrogate to guide Bayesian optimization. **LLAMBO** uses in-context learning for candidate generation and surrogate prediction. **CAKE** evolves compositional kernels through LLM-guided mutation and crossover. **LGBO** incorporates LLM-derived preferences into GP predictions.

Implementation details. For GP methods with a fixed kernel, we use an RBF kernel and qLogEI as the baseline acquisition function. Our framework is implemented in double precision using BoTorch (Balandat et al., 2020). Each optimization run begins with five initial evaluations, followed by 25 sequential optimization rounds, for a total budget of 30 evaluations. Following their respective initialization protocols, PHEBO, LLAMBO, and LGBO share the same LLM-suggested initial points, while GPBO and CAKE use Sobol initialization. We repeat each experiment with five random seeds and report the mean $\pm$ standard deviation across runs. All LLM-based methods use Kimi-K2.6 (Moonshot AI, 2026). Beyond the task description and observed evaluations, prompts do not provide explicit descriptions of dataset patterns or statistical conclusions as prior guidance.

Hyperparameters. We set the LLM sampling temperature to 0.6. Each mechanism contains at most two terms involving at most four distinct variables. The active mechanism capacity is set to $K = 6$. At scheduled interventions, the LLM proposes at most three new hypotheses. Interventions occur at the beginning of BO rounds $t = 1, 4, 7, \ldots$, after every three completed BO evaluations. Credit updates use the improvement-censored score with clipping threshold $\kappa = 1$ and a half-life of six observations, corresponding to a discount factor of $\lambda = 2^{-1/6}$. We set the candidate capacity to $B = 5$; further numerical settings are given in Appendix A.8.

Tasks. We consider four offline benchmarks spanning analytical chemistry, materials design and organic photovoltaics: **HPLC** (Häse et al., 2021), which optimizes automated chromatographic sample handling to maximize peak area; **Concrete** (Tripathi, 2020), which optimizes mixture composition to maximize compressive strength; **Cross-barrel** (Gongora et al., 2020), which optimizes the geometry of 3D-printed structures to maximize mechanical toughness; and **PCE10** (Langner et al., 2020), which optimizes organic photovoltaic blend composition to minimize photodegradation. Dataset specifications and evaluation oracles are detailed in Appendix B.

5.2 Main Results

Figure 2 compares optimization trajectories across four scientific tasks. PHEBO achieves the highest final mean relative objective on HPLC, Cross-barrel, and Concrete, and ranks second on PCE10. On HPLC (a) and Cross-barrel (b), it leads the baselines over most rounds, with an advantage already apparent by rounds 5 and 10, respectively. These trajectories indicate earlier gains and better final performance on these two tasks within the evaluation budget. On Concrete (c), PHEBO's advantage

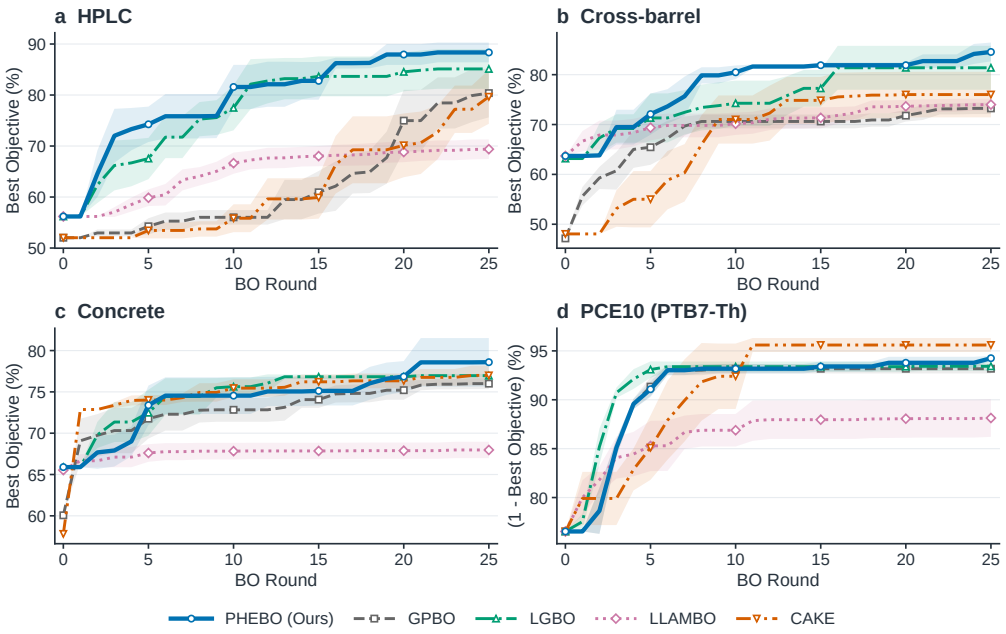

Figure 2: Optimization performance, expressed as normalized best-so-far objective values (%), computed as $(y^\star - y_{\min})/(y_{\max} - y_{\min})$, where $y^\star$ denotes the best objective value observed, and $y_{\min}$ and $y_{\max}$ are the minimum and maximum objective values in the corresponding dataset, respectively. Curves show means over five independent seeds; shaded bands indicate standard deviations. Higher values indicate better performance.

emerges late in the optimization process. Its mean relative objective rises from 76.9% at round 20 to 78.6% at round 21, while LGBO and CAKE plateau at approximately 76–77% from round 15 onward. On PCE10 (d), GPBO and LGBO plateau near 93%, while PHEBO makes modest late improvements and finishes at 94.3%. The three methods, which share the same acquisition policy over a fixed Sobol candidate set, exhibit similar trajectories with signs of stagnation. In contrast, CAKE continuously optimizes its acquisition function and achieves a higher score of 95.6%. Further analysis is provided in Appendix C.1. The results therefore support competitive mean performance across these tasks.

Further comparison of the optimization trajectories on the Cross-barrel task, shown in Figure 3(a), reveals that GPBO and PHEBO start from comparable locations. After reaching a broad plateau, GPBO becomes trapped in localized exploration. In contrast, by leveraging the gains from its early-stage exploration, PHEBO escapes the local plateau and ultimately reaches a higher-performing region. This contrast highlights how the LLM-informed belief effectively guides exploration while enabling better use of experience accumulated during the optimization process. Cost–benefit and hyperparameter sensitivity analyses are provided in Appendices C.2 and C.3, respectively.

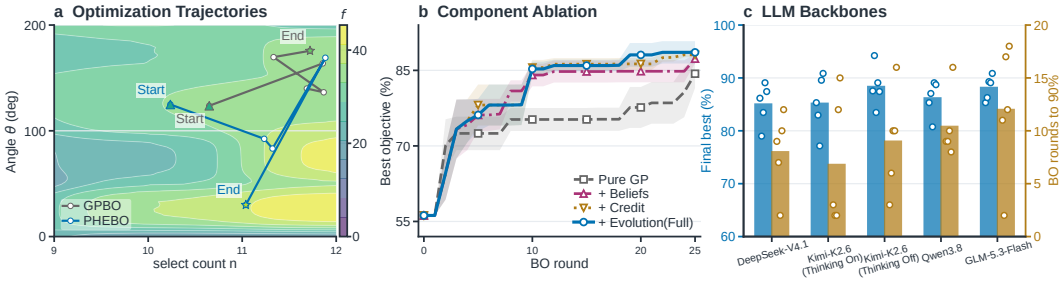

Figure 3: Optimization trajectories and ablation results.

5.3 Ablation Study

Component Ablation. To assess the contribution of individual components in PHEBO, we conduct incremental ablation experiments on the HPLC dataset. We construct four variants by progressively adding belief injection, predictive credit, and belief evolution. *Pure GP* performs Bayesian optimization without belief guidance. + *Beliefs* incorporates the initial LLM-generated beliefs, while keeping both the belief population and the acquisition mixture weights fixed. + *Credit* retains the same fixed belief population but adjusts its influence on acquisition using credit accumulated. Finally, + *Evolution(Full)* additionally enables credit-guided belief evolution. The three belief-based variants share the same initial belief population observations within each seed.

As shown in Figure 3(b), adding fixed beliefs raises the final objective, suggesting that conditional hypothesis injection can provide useful guidance even without subsequent adaptation. Predictive credit further increases the mean and reduces the final standard deviation from 1.83 to 0.95 percentage points, consistent with more reliable guidance with credit. The full method achieves the highest final objectives, 88.58%, suggesting an additional benefit from updating the belief population.

LLM backbones. To assess the robustness of our framework across different LLM backbones, we evaluated **DeepSeek-V4.1**, **Qwen3.8**, **GLM-5.3-Flash**, and thinking-on/off variants of **Kimi-K2.6**. As shown in Figure 3(c), all backbone configurations achieve comparable best objectives, demonstrating that the framework is not sensitive to the choice of LLM. Moreover, enabling thinking allows **Kimi-K2.6** to reach 90% of best performance in fewer rounds, suggesting that explicit reasoning can improve optimization efficiency. Overall, these results demonstrate the stable performance and broad adaptability of PHEBO across different LLM backbones and inference modes.

5.4 Transferability of the Beliefs

Beyond optimization performance, we investigate whether the evolved beliefs capture task knowledge that can inform a different framework. For each task, we select the three highest-credit hypotheses with evidence count $n_i > 5$ from the final belief set. We transfer these hypotheses, together with their associated mechanism descriptions, to LGBO as domain priors. At each iteration, LGBO uses these priors alongside the accumulated observations to generate preferences that guide candidate selection.

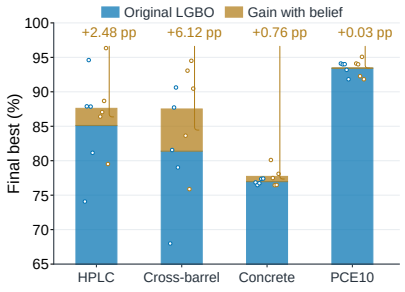

Figure 4: Belief-prior transfer to LGBO. Gold segments indicate gains over original LGBO.

Figure 4 shows that belief-enhanced LGBO achieves higher mean final performance across all four tasks, although the gains vary substantially and are marginal on PCE10. The consistently non-negative effect suggests that the evolved beliefs encode actionable domain knowledge rather than optimizer-specific search heuristics. Together, these results provide preliminary evidence that evolved beliefs can transfer beyond their originating framework, with the potential to inform scientific discovery and support other LLM-enhanced workflows.

6 Conclusion

This work introduces PHEBO, a Bayesian optimization framework in which predictive feedback drives the evolution of LLM-generated domain hypotheses. By translating qualitative hypotheses into conditional GP experts and updating their credits using subsequent predictive evidence, PHEBO links hypothesis evaluation, acquisition decisions, and hypothesis revision in a closed loop. Across four offline scientific optimization tasks, PHEBO outperforms GP-based and LLM-enhanced BO baselines overall. Ablation studies demonstrate the benefits of hypothesis injection, predictive credit, and hypothesis evolution. Cross-framework transfer results further suggest that the evolved hypotheses provide useful guidance beyond the framework in which they were developed. Together, these findings support predictive feedback as a basis for evaluating and refining LLM-generated hypotheses, improving optimization while producing reusable hypotheses for further scientific investigation.

REFERENCES

Dhruv Agarwal, Manoj Ghuhan Arivazhagan, Rajarshi Das, Sandesh Swamy, Sopan Khosla, and Rashmi Gangadharaiah. Searching for optimal solutions with LLMs via Bayesian optimization. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a79237d6c47feef24c1df723cb41f674-Abstract-Conference.html.

Sebastian Ament, Samuel Daulton, David Eriksson, Maximilian Balandat, and Eytan Bakshy. Unexpected improvements to expected improvement for Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 20577–20612, 2023. doi: 10.52202/075280-0904. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/419f72cbd568ad62183f8132a3605a2a-Paper-Conference.pdf.

Maximilian Balandat, Brian Karrer, Daniel R. Jiang, Samuel Daulton, Benjamin Letham, Andrew Gordon Wilson, and Eytan Bakshy. BoTorch: A framework for efficient monte-carlo Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f5b1b89d98b7286673128a5fb112cb9a-Paper.pdf.

Mayk Caldas Ramos, Shane S. Michtavy, Andrew D. White, and Marc D. Porosoff. Bayesian optimization of catalysis with in-context learning. *ACS Central Science*, 12(5):599–615, 2026. doi: 10.1021/acscentsci.5c02418. URL https://doi.org/10.1021/acscentsci.5c02418.

Chih-Yu Chang, Milad Azvar, Chinedum Okwudire, and Raed Al Kontar. LLINBO: Trustworthy LLM-in-the-loop Bayesian optimization. arXiv preprint arXiv:2505.14756, 2025. URL https://arxiv.org/abs/2505.14756v2.

Guojin Chen, Keren Zhu, Seunggeun Kim, Hanqing Zhu, Yao Lai, Bei Yu, and David Z. Pan. LLM-enhanced Bayesian optimization for efficient analog layout constraint generation. arXiv preprint arXiv:2406.05250, 2024. URL https://arxiv.org/abs/2406.05250v3.

Qiyuan Chen, Liangkui Jiang, Hantang Qin, and Raed Al Kontar. Multi-agent collaborative Bayesian optimization via constrained Gaussian processes. *Technometrics*, 67(1):32–45, 2025. doi: 10.1080/00401706.2024.2365732. URL https://doi.org/10.1080/00401706.2024.2365732.

Abdoulatif Cissé, Xenophon Evangelopoulos, Sam Carruthers, Vladimir V. Gusev, and Andrew I. Cooper. HypBO: Accelerating black-box scientific experiments using experts' hypotheses. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 3881–3889, 2024. doi: 10.24963/ijcai.2024/429. URL https://www.ijcai.org/proceedings/2024/429.

Abdoulatif Cissé, Xenophon Evangelopoulos, Vladimir V. Gusev, and Andrew I. Cooper. Language-based Bayesian optimization research assistant (BORA). In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 4967–4975, 2025. doi: 10.24963/ijcai.2025/553. URL https://www.ijcai.org/proceedings/2025/553.

A. Philip Dawid. Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society, Series A (General)*, 147(2):278–290, 1984. doi: 10.2307/2981683. URL https://academic.oup.com/jrssa/article/147/2/278/7106293.

Cees Diks, Valentyn Panchenko, and Dick van Dijk. Likelihood-based scoring rules for comparing density forecasts in tails. *Journal of Econometrics*, 163(2):215–230, 2011. doi: 10.1016/j.jeconom.2011.04.001. URL https://doi.org/10.1016/j.jeconom.2011.04.001.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437. URL https://doi.org/10.1198/016214506000001437.

Aldair E. Gongora, Bowen Xu, Wyatt Perry, Chika Okoye, Patrick Riley, Kristofer G. Reyes, Elise F. Morgan, and Keith A. Brown. A Bayesian experimental autonomous researcher for mechanical design. *Science Advances*, 6(15):eaaz1708, 2020. doi: 10.1126/sciadv.aaz1708. URL https://doi.org/10.1126/sciadv.aaz1708.

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, 2021. doi: 10.1088/2632-2153/abedc8. URL https://doi.org/10.1088/2632-2153/abedc8.

Matthew Hoffman, Eric Brochu, and Nando de Freitas. Portfolio allocation for Bayesian optimization. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pp. 327–336, 2011. URL https://mwhoffman.com/papers/hoffman_2011.pdf.

John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press, 1992. URL https://mitpress.mit.edu/9780262581110/adaptation-in-natural-and-artificial-systems/.

Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. πBO: Augmenting acquisition functions with user beliefs for Bayesian optimization. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2204.11051.

Carl Hvarfner, Frank Hutter, and Luigi Nardi. A general framework for user-guided Bayesian optimization. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/b95f3f8b6cff9e7ac6553a56b895cf70-Abstract-Conference.html.

Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998. doi: 10.1023/A:1008306431147. URL https://doi.org/10.1023/A:1008306431147.

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. doi: 10.1002/adma.201907801. URL https://doi.org/10.1002/adma.201907801.

Jerry Liu, Jessica Grogan, Owen Dugan, Ashish Rao, Simran Arora, Atri Rudra, and Christopher Ré. Towards learning high-precision least squares algorithms with sequence models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/11662b256280f35f885a341833a37b32-Abstract-Conference.html.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2402.03921.

Kanan Mahammadli and Seyda Ertekin. Sequential large language model-based hyper-parameter optimization. arXiv preprint arXiv:2410.20302, version 3, 2025. URL https://arxiv.org/abs/2410.20302v3.

Moonshot AI. Kimi K2.6. Model card, 2026. URL https://huggingface.co/moonshotai/Kimi-K2.6.

Giang Ngo, Dat Phan Trong, Dang Nguyen, Sunil Gupta, and Svetha Venkatesh. Adaptive acquisition selection for Bayesian optimization with large language models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/f1b7a29fe9759d4c9d23f78738e40b42-Abstract-Conference.html.

Adrian E. Raftery, Miroslav Kárný, and Pavel Ettler. Online prediction under model uncertainty via dynamic model averaging: Application to a cold rolling mill. *Technometrics*, 52(1):52–66, 2010. doi: 10.1198/TECH.2009.08104. URL https://sites.stat.washington.edu/people/raftery/Research/PDF/Karny2010.pdf.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. URL https://gaussianprocess.org/gpml/chapters/RW2.pdf.

Michael A. Riegler, Kristoffer H. Hellton, Vajira Thambawita, and Hugo L. Hammer. Using large language models to suggest informative prior distributions in Bayesian regression analysis. *Scientific Reports*, 15:33386, 2025. doi: 10.1038/s41598-025-18425-9. URL https://www.nature.com/articles/s41598-025-18425-9.

Loïc M. Roch, Florian Häse, Christoph Kreisbeck, Teresa Tamayo-Mendoza, Lars P. E. Yunker, Jason E. Hein, and Alán Aspuru-Guzik. ChemOS: An orchestration software to democratize autonomous discovery. *PLOS ONE*, 15(4):e0229862, 2020. doi: 10.1371/journal.pone.0229862. URL https://doi.org/10.1371/journal.pone.0229862.

Richard Cornelius Suwandi, Feng Yin, Juntao Wang, Renjie Li, Tsung-Hui Chang, and Sergios Theodoridis. Adaptive kernel design for Bayesian optimization is a piece of CAKE with LLMs. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/c03a2610bca2712b984b331fd4f7bb6f-Paper-Conference.pdf.

Pratham Tripathi. Calculate concrete strength. Kaggle dataset, 2020. URL https://www.kaggle.com/datasets/prathamtripathi/regression-with-neural-networking.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/6733cf15e10e2cd1d59af033c3bb8507-Abstract-Conference.html.

Wenjie Xu, Masaki Adachi, Colin N. Jones, and Michael A. Osborne. Principled Bayesian optimization in collaboration with human experts. In *Advances in Neural Information Processing Systems*, volume 37, pp. 104091–104137, 2024. doi: 10.52202/079017-3308. URL https://papers.nips.cc/paper_files/paper/2024/hash/bc82dbfbfa43232be85b8d9838f49c3e-Abstract-Conference.html.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/3339f19c5fcee3ad74502947a32be9e6-Abstract-Conference.html.

I-Cheng Yeh. Concrete compressive strength. UCI Machine Learning Repository, 1998a. URL https://doi.org/10.24432/C5PK67. Dataset.

I-Cheng Yeh. Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research*, 28(12):1797–1808, 1998b. doi: 10.1016/S0008-8846(98)00165-3. URL https://doi.org/10.1016/S0008-8846(98)00165-3.

Yuxuan Yin, Yu Wang, Boxun Xu, and Peng Li. ADO-LLM: Analog design Bayesian optimization with in-context learning of large language models. In *IEEE/ACM International Conference on Computer-Aided Design*, 2024. doi: 10.1145/3676536.3676816. URL https://doi.org/10.1145/3676536.3676816.

Xinzhe Yuan, Zhuo Chen, Jianshu Zhang, Huan Xiong, Nanyang Ye, Yuqiang Li, and Qinying Gu. Unleashing LLMs in Bayesian optimization: Preference-guided framework for scientific discovery. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/85381f4549b5ddf1d48e2e287d7d3d15-Abstract-Conference.html.

LLM USAGE

We used large language models (LLMs) as auxiliary tools for writing assistance, language polishing, and literature retrieval and discovery. Specifically, LLMs were employed to improve the readability, grammar, and presentation of the text and to help retrieve and identify relevant prior work. The authors are responsible for the research ideas, experimental designs, and scientific contributions and take full responsibility for the content of this paper.

LIMITATIONS

This work has several limitations. Our evaluation uses offline benchmarks with deterministic oracles, leaving performance in prospective wet-lab optimization untested. Future work could evaluate the framework under experimental noise and laboratory constraints. The current implementation also restricts each mechanism to a predefined grammar with at most two terms and four distinct variables. Extending this representation while maintaining reliable proposal validation and tractable inference could accommodate more complex relationships. Finally, the fixed candidate set can limit attainable performance, as illustrated by the PCE10 results. Adaptive candidate generation or continuous acquisition optimization could improve search-space coverage.

APPENDIX CONTENTS

A **Derivations and implementation conventions**
A.1 Feature evaluation and preference construction
A.2 Coordinates, scaling, and the shared GP
A.3 Conditional expert inference
A.4 Observation density and non-improvement probability
A.5 Censored scoring and credit dynamics
A.6 Expected improvement and quadrature
A.7 Expected improvement of the predictive mixture
A.8 Baseline candidate construction
A.9 Closed-loop algorithm and execution conventions

B **Benchmark datasets and evaluation oracles**
B.1 HPLC
B.2 Concrete
B.3 Cross-barrel
B.4 PCE10

C **Additional experiments and analysis**
C.1 Reachability ceiling
C.2 Cost-benefit analysis
C.3 Hyperparameter sensitivity

D **Prompts for belief update**

E **Case study: Belief evolution and sampling interventions**
E.1 Trajectory selection and background
E.2 A traceable lineage of executable beliefs
E.3 How mixture EI changes the sampled point

A Derivations and implementation conventions

A.1 Feature evaluation and preference construction

For term r of mechanism m_i, let $z_{i,r}(\mathbf{x})$ evaluate its parameter expression and let $o_{i,r} \in \{\mathrm{Inc}, \mathrm{Dec}, \mathrm{Opt}\}$ denote its preference. Parameters p_j are symbolic names; their values are the components x_j of $\mathbf{x}$. Let $[a_{i,r}, b_{i,r}]$ be the feature range induced by the input bounds and set

$$\bar{z}_{i,r}(\mathbf{x}) = \mathrm{clip}\left(\frac{z_{i,r}(\mathbf{x}) - a_{i,r}}{b_{i,r} - a_{i,r}}, 0, 1\right).$$

The term preference is

$$\psi_{i,r}(\mathbf{x}) = \max\left\{\varepsilon_\psi, \begin{cases} \bar{z}_{i,r}(\mathbf{x}), & o_{i,r} = \mathrm{Inc}, \\ 1 - \bar{z}_{i,r}(\mathbf{x}), & o_{i,r} = \mathrm{Dec}, \\ \exp\left[-\dfrac{(z_{i,r}(\mathbf{x}) - c_{i,r})^2}{2\rho_{i,r}^2}\right], & o_{i,r} = \mathrm{Opt}. \end{cases}\right\} \tag{15}$$

Here $0 < \varepsilon_\psi < 1$, $c_{i,r} \in [a_{i,r}, b_{i,r}]$, and $\rho_{i,r} > 0$. For an Opt preference on a composite feature, the Gaussian preference width is fixed at

$$\rho_{i,r} = 0.2\big(b_{i,r} - a_{i,r}\big).$$

For an original parameter, the width $\rho_{i,r}$ is proposed by the LLM within the allowed range. Ratio features are accepted only when the denominator's parameter range excludes zero. Equation (4) combines these preferences into the mechanism basis.

A.2 Coordinates, scaling, and the shared GP

Section 4.2 expresses the expert model and its predictions in objective units. This section specifies the input transformation, outcome standardization, and shared GP fitting used in the implementation.

Input and outcome transformations. For $\mathcal{X} \subseteq \prod_{j=1}^{D}[a_j, b_j]$, define

$$T_j(\mathbf{x}) = \mathrm{clip}\left(\frac{x_j - a_j}{b_j - a_j}, \varepsilon_x, 1 - \varepsilon_x\right). \tag{16}$$

For composition-constrained tasks, clipping to $[0, 1]$ instead preserves valid boundary inputs. Let X collect the observed inputs and $U = T(X)$. For $n = |\mathcal{D}_{t-1}| \geq 2$, let $\bar{y}_t$ and $\widehat{\gamma}_t$ be the sample mean and sample standard deviation of the observed outcomes, with denominator $n - 1$ for the latter. Set $\gamma_t = \widehat{\gamma}_t$ if $\widehat{\gamma}_t \geq \varepsilon_s$, and $\gamma_t = 1$ otherwise. The standardized observations and objective are

$$\widetilde{y}_{s,t} = \frac{y_s - \bar{y}_t}{\gamma_t}, \qquad \widetilde{f}_t(\mathbf{x}) = \frac{f(\mathbf{x}) - \bar{y}_t}{\gamma_t}. \tag{17}$$

Minimization objectives are negated before standardization. The same transformation is applied to the new outcome, $\widetilde{y}_t = (y_t - \bar{y}_t)/\gamma_t$, with $\bar{y}_t$ and γ_t fixed before y_t is observed. We use s to index observations, j, k to index parameters, i to index mechanisms, and r to index their terms.

Shared GP fitting and parameter conversion. Let $\widetilde{\mathbf{y}}_t$ collect the standardized outcomes in $\mathcal{D}_{t-1}$. Writing θ for the GP hyperparameters, the implementation obtains the point estimate

$$\widehat{\theta}_t \in \arg\max_{\theta} \left\{\log p(\widetilde{\mathbf{y}}_t \mid U, \theta) + \log p_{\mathrm{GP}}(\theta)\right\}, \tag{18}$$

where p_{GP} denotes the registered hyperparameter priors. All mechanism experts use this fitted parameter estimate. Let $\widetilde{\mu}_{0,t}^{\mathrm{prior}}$, $\widetilde{k}_t$, and $\widetilde{\sigma}_{n,t}^2$ denote the fitted prior mean, kernel, and noise variance in numerical coordinates. Their counterparts in objective units are

$$\begin{aligned} \mu_{0,t}^{\mathrm{prior}}(\mathbf{x}) &= \bar{y}_t + \gamma_t \widetilde{\mu}_{0,t}^{\mathrm{prior}}(T(\mathbf{x})), \\ k_t(\mathbf{x}, \mathbf{x}') &= \gamma_t^2 \widetilde{k}_t(T(\mathbf{x}), T(\mathbf{x}')), \\ \sigma_{n,t}^2 &= \gamma_t^2 \widetilde{\sigma}_{n,t}^2, \qquad \beta_i = \gamma_t \widetilde{\beta}_i, \qquad \tau_t = \gamma_t \tau. \end{aligned} \tag{19}$$

Here $\widetilde{\beta}_i$ has a half-normal prior with scale τ in standardized units; the corresponding scale in Equation (5) is τ_t. These transformations preserve the fitted model and coefficient prior.

Baseline predictions and numerical evaluation. Let $\mathbf{y}_{t-1}$ collect the observed outcomes and define

$$e_t = \mathbf{y}_{t-1} - \mu_{0,t}^{\text{prior}}(X), \qquad C_t = k_t(X,X) + \sigma_{n,t}^2 I, \qquad k_{t,\mathbf{x}} = k_t(X,\mathbf{x}).$$

The baseline latent posterior has mean and variance

$$\begin{aligned} \mu_{0,t}(\mathbf{x}) &= \mu_{0,t}^{\text{prior}}(\mathbf{x}) + k_{t,\mathbf{x}}^\top C_t^{-1} e_t, \\ v_{0,t}(\mathbf{x}) &= k_t(\mathbf{x},\mathbf{x}) - k_{t,\mathbf{x}}^\top C_t^{-1} k_{t,\mathbf{x}}. \end{aligned} \tag{20}$$

For numerical evaluation, the implementation applies a positive floor ε_v to latent variance in standardized units before conversion to objective units. For either latent or observation predictions, let $\widetilde{q}_{i,t}$ denote the density in standardized units. Then

$$q_{i,t}(u \mid \mathbf{x}) = \gamma_t^{-1} \widetilde{q}_{i,t}\left(\frac{u-\bar{y}_t}{\gamma_t} \,\middle|\, \mathbf{x}\right), \qquad \text{EI}_{i,t}(\mathbf{x}) = \gamma_t \widetilde{\text{EI}}_{i,t}(\mathbf{x}). \tag{21}$$

The standardized EI uses the threshold $\widetilde{a}_t = (a_t - \bar{y}_t)/\gamma_t$. The common positive factor preserves mixture-EI rankings. The density Jacobian cancels in relative log-score comparisons, while CDF probabilities are invariant under the same increasing affine transformation.

A.3 Conditional expert inference

We derive the coefficient posterior and predictive distribution for Equation (5), conditional on the specified basis h_i and fitted shared GP parameters. The data-dependent generation of the mechanism is not integrated into this posterior. Throughout, all quantities are expressed in objective units.

Coefficient posterior. Starting from Equation (5), we omit the round and mechanism indices for clarity and write the observation model as

$$\mathbf{y} = \mu_0^{\text{prior}}(X) + \beta h(X) + g(X) + \boldsymbol{\epsilon},$$

where X collects the observed inputs, $g(X) \sim \mathcal{N}(0, k(X,X))$, and $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma_n^2 I)$. The GP residual and observation noise are independent of each other and of β a priori. Hence,

$$g(X) + \boldsymbol{\epsilon} \sim \mathcal{N}(0, C), \qquad C = k(X,X) + \sigma_n^2 I,$$

which gives

$$\mathbf{y} \mid \beta \sim \mathcal{N}\Big(\mu_0^{\text{prior}}(X) + \beta h(X), C\Big).$$

Define the residual vector and the basis vector as

$$e = \mathbf{y} - \mu_0^{\text{prior}}(X), \qquad H = h(X).$$

Then $e \mid \beta \sim \mathcal{N}(\beta H, C)$. Assuming C is positive definite, the likelihood is

$$p(e \mid \beta) = (2\pi)^{-n/2} |C|^{-1/2} \exp\left[-\frac{1}{2}(e-\beta H)^\top C^{-1}(e-\beta H)\right],$$

where n is the number of observations. Holding the fitted GP parameters fixed and omitting factors independent of β, we obtain

$$p(e \mid \beta) \propto \exp\left[-\frac{1}{2}(e-\beta H)^\top C^{-1}(e-\beta H)\right].$$

The coefficient has a half-normal prior,

$$p(\beta) = \frac{2}{\tau}\phi\left(\frac{\beta}{\tau}\right)\mathbf{1}_{\{\beta \geq 0\}}, \qquad \tau > 0,$$

where ϕ denotes the standard normal density. By Bayes' rule,

$$\begin{aligned} p(\beta \mid e) &\propto p(e \mid \beta) p(\beta) \\ &\propto \mathbf{1}_{\{\beta \geq 0\}} \exp\left[-\frac{1}{2}\left\{(e-\beta H)^\top C^{-1}(e-\beta H) + \tau^{-2}\beta^2\right\}\right]. \end{aligned} \tag{22}$$

Expanding the quadratic form yields

$$(e-\beta H)^\top C^{-1}(e-\beta H)+\tau^{-2}\beta^2 \\ = e^\top C^{-1}e - 2\beta H^\top C^{-1}e + \beta^2\left(H^\top C^{-1}H+\tau^{-2}\right). \tag{23}$$

Since $e^\top C^{-1}e$ is independent of β, it can be absorbed into the proportionality constant. Define

$$A = H^\top C^{-1}H+\tau^{-2}, \qquad B = H^\top C^{-1}e.$$

Completing the square gives

$$A\beta^2 - 2B\beta = A\left(\beta - \frac{B}{A}\right)^2 - \frac{B^2}{A}.$$

The last term is also independent of β. Therefore,

$$p(\beta \mid e) \propto \mathbf{1}_{\{\beta\geq 0\}} \exp\left[-\frac{(\beta-b)^2}{2\nu}\right],$$

where

$$\nu = \left(\tau^{-2}+H^\top C^{-1}H\right)^{-1}, \qquad b = \nu H^\top C^{-1}e. \tag{24}$$

Thus, the coefficient posterior is a Gaussian distribution truncated to the nonnegative half-line:

$$\beta \mid e \sim \mathcal{N}(b,\nu)\big|_{[0,\infty)}.$$

Here ν^{-1} combines the prior precision τ^{-2} with the information $H^\top C^{-1}H$ supplied by the likelihood. The parameter b is the unconstrained minimizer of

$$(e-\beta H)^\top C^{-1}(e-\beta H)+\tau^{-2}\beta^2,$$

and therefore represents a regularized estimate of the coefficient before imposing the nonnegativity constraint. Importantly, b and ν are the mean and variance parameters of the Gaussian before truncation, not the mean and variance of the truncated posterior.

To normalize the posterior, note that

$$\int_0^\infty \frac{1}{\sqrt{\nu}}\phi\left(\frac{\beta-b}{\sqrt{\nu}}\right)\,d\beta = \Phi\left(\frac{b}{\sqrt{\nu}}\right),$$

where Φ denotes the standard normal CDF. The normalized posterior density is therefore

$$p(\beta \mid e) = \frac{\phi((\beta-b)/\sqrt{\nu})}{\sqrt{\nu}\,\Phi(b/\sqrt{\nu})}\mathbf{1}_{\{\beta\geq 0\}}. \tag{25}$$

The factor $\Phi(b/\sqrt{\nu})$ is the probability mass of the untruncated Gaussian on $[0,\infty)$; dividing by this factor ensures that the posterior density integrates to one.

Conditional prediction and residual correction. For a fixed coefficient $\beta_i = \beta$, the residual observations are

$$e_t - \beta H_i = g_t(X) + \epsilon, \qquad \epsilon \sim \mathcal{N}(0,\sigma_{n,t}^2 I).$$

With $k_{t,\mathbf{x}} = k_t(X,\mathbf{x})$, Gaussian conditioning gives

$$g_t(\mathbf{x}) \mid \beta, \mathcal{D}_{t-1} \sim \mathcal{N}\left(k_{t,\mathbf{x}}^\top C_t^{-1}(e_t-\beta H_i), v_{0,t}(\mathbf{x})\right). \tag{26}$$

The conditional covariance is unchanged because βH_i is deterministic for fixed β. Adding the prior mean and the mechanism contribution at $\mathbf{x}$ gives

$$\begin{aligned}\mathbb{E}[f_{i,t}(\mathbf{x}) \mid \beta, \mathcal{D}_{t-1}] &= \mu_{0,t}^{\text{prior}}(\mathbf{x}) + k_{t,\mathbf{x}}^\top C_t^{-1}(e_t-\beta H_i) + \beta h_i(\mathbf{x}) \\ &= \mu_{0,t}(\mathbf{x}) + \beta\left[h_i(\mathbf{x}) - k_{t,\mathbf{x}}^\top C_t^{-1}H_i\right] \\ &= \mu_{0,t}(\mathbf{x}) + \beta r_{i,t}(\mathbf{x}).\end{aligned} \tag{27}$$

The second equality uses the baseline posterior mean in Equation (20), and the final equality uses the residual basis defined in Equation (6). Its subtraction term accounts for the adjustment to the

GP residual after assigning βH_i to the mechanism at the observed inputs. The conditional latent prediction is therefore

$$f_{i,t}(\mathbf{x}) \mid \beta, \mathcal{D}_{t-1} \sim \mathcal{N}(\mu_{0,t}(\mathbf{x}) + \beta r_{i,t}(\mathbf{x}), v_{0,t}(\mathbf{x}))\,. \tag{28}$$

Marginal prediction and coefficient uncertainty. Integrating the conditional prediction against $p_{i,t}(\beta)$ gives Equation (7). By the law of total variance,

$$\mathrm{Var}[f_{i,t}(\mathbf{x}) \mid \mathcal{D}_{t-1}] = v_{0,t}(\mathbf{x}) + r_{i,t}(\mathbf{x})^2 \,\mathrm{Var}_{p_{i,t}}[\beta]. \tag{29}$$

The second term arises from marginalizing over β. It must not also be added to the conditional variance inside the predictive integral. For a noisy observation, the conditional variance becomes $v_{0,t}(\mathbf{x}) + \sigma^2_{n,t}$ before marginalization.

A.4 Observation density and non-improvement probability

We derive the observation density and CDF from the predictive distribution in Section 4.2. Fix a round, a mechanism, and a query point, and abbreviate

$$\mu = \mu_{0,t}(\mathbf{x}), \qquad r = r_{i,t}(\mathbf{x}), \qquad V = v_{0,t}(\mathbf{x}) + \sigma^2_{n,t}, \qquad b = b_{i,t}, \qquad \nu = \nu_{i,t}.$$

Introduce an auxiliary untruncated coefficient $B \sim \mathcal{N}(b, \nu)$ and let $Y \mid B \sim \mathcal{N}(\mu + rB, V)$. Conditioning this joint distribution on $B \geq 0$ recovers the expert's observation law. Before truncation, the joint Gaussian implies

$$\begin{aligned} A = V + r^2\nu, \qquad d = u - \mu - rb, \\ B \mid Y = u \sim \mathcal{N}(b^+,\, \nu^+), \qquad b^+ = b + \frac{\nu r}{A} d, \qquad \nu^+ = \frac{\nu V}{A}. \end{aligned} \tag{30}$$

The expert's observation law conditions this joint distribution on $B \geq 0$. Factoring the joint density therefore gives

$$\begin{aligned} q_i^{\mathrm{obs}}(u) &= \mathcal{N}(u; \mu + rb, A) \frac{\Pr(B \geq 0 \mid Y = u)}{\Pr(B \geq 0)}, \\ \log q_i^{\mathrm{obs}}(u) &= -\frac{1}{2}\left[\log(2\pi A) + \frac{d^2}{A}\right] + \log \Phi\left(\frac{b^+}{\sqrt{\nu^+}}\right) - \log \Phi\left(\frac{b}{\sqrt{\nu}}\right). \end{aligned} \tag{31}$$

All quantities on the right are determined by the archived pre-outcome forecast and the subsequently observed u; no GP refit is needed.

For censoring at a, integrate the Gaussian observation CDF over the truncated coefficient:

$$Q_i^{\mathrm{obs}}(a) = \frac{1}{\Phi(b/\sqrt{\nu})} \int_0^\infty \Phi\left(\frac{a - \mu - r\beta}{\sqrt{V}}\right) \frac{\phi((\beta - b)/\sqrt{\nu})}{\sqrt{\nu}}\, d\beta. \tag{32}$$

The baseline uses $Q_0^{\mathrm{obs}}(a) = \Phi((a - \mu)/\sqrt{V})$. The implementation evaluates the expert integral with adaptive quadrature after scaling its log-concave integrand around its mode, and uses stable log-CDF evaluation for Equation (31).

A.5 Censored scoring and credit dynamics

Propriety of the censored score. Condition on all information available before evaluation t, including the selected input $\mathbf{x}_t$ and incumbent a_t. Let p be the true observation density, $P(a_t)$ its probability of non-improvement, and q a candidate predictive density with CDF Q. Censoring maps all outcomes $u \leq a_t$ to a single symbol $\perp$, while retaining outcomes above a_t. The resulting predictive law has mass $Q(a_t)$ at $\perp$ and density $q(u)$ for $u > a_t$. Its logarithmic score is Equation (8) (Diks et al., 2011). Under the integrability conditions for the logarithmic score,

$$\begin{aligned} \mathbb{E}_p[S(p, Y) - S(q, Y)] &= P(a_t) \log \frac{P(a_t)}{Q(a_t)} + \int_{a_t}^\infty p(u) \log \frac{p(u)}{q(u)}\, du \\ &= D_{\mathrm{KL}}(P^{\mathrm{cens}} \| Q^{\mathrm{cens}}) \geq 0. \end{aligned} \tag{33}$$

Thus, the score is strictly proper for the censored outcome law. It does not distinguish full predictive distributions that agree above a_t and assign the same total probability below it. The argument remains valid when $\mathbf{x}_t$ and a_t depend on past observations, since both are fixed before y_t is revealed. The credit update clips the score difference relative to the baseline. The propriety of the censored score therefore does not automatically extend to this nonlinear update.

Invariance to outcome standardization. Using the transformation in Appendix A.2, let $\widetilde{y}_t = (y_t - \bar{y}_t)/\gamma_t$ and $\widetilde{a}_t = (a_t - \bar{y}_t)/\gamma_t$. For an improving outcome,

$$\log q_{i,t}^{\mathrm{obs}}(y_t \mid \mathbf{x}_t) = -\log \gamma_t + \log \widetilde{q}_{i,t}^{\mathrm{obs}}(\widetilde{y}_t \mid \mathbf{x}_t).$$

The common Jacobian term cancels in the comparison with the baseline. For a non-improving outcome,

$$Q_{i,t}^{\mathrm{obs}}(a_t \mid \mathbf{x}_t) = \widetilde{Q}_{i,t}^{\mathrm{obs}}(\widetilde{a}_t \mid \mathbf{x}_t).$$

Consequently, if $\widetilde{S}_{i,t}$ denotes the score in standardized units,

$$S_{i,t}(y_t) - S_{0,t}(y_t) = \widetilde{S}_{i,t}(\widetilde{y}_t) - \widetilde{S}_{0,t}(\widetilde{y}_t). \tag{34}$$

Credit updates are therefore unchanged by this transformation, provided the forecast, threshold, and outcome use the same pre-evaluation scaling.

Inheritance, discounting, and bounded credit. For a mechanism born in round $b = t_i^{\mathrm{birth}}$, Equation (11) copies the parent's credit under mutation and averages the two parent credits under crossover. For crossover,

$$\exp(\ell_i^{\mathrm{birth}}) = \exp\left(\frac{\ell_{a_1,b} + \ell_{a_2,b}}{2}\right) = \sqrt{\exp(\ell_{a_1,b})\exp(\ell_{a_2,b})}. \tag{35}$$

Thus, arithmetic averaging of the stored credits corresponds to geometric averaging of their exponentiated values. Both inheritance operations preserve the range of parental credits, including negative values. The birth value is applied once; parental forecast counts and observation identifiers are not transferred. Let $\Delta_{i,t} = \mathrm{clip}(\delta_{i,t}, -\kappa, \kappa)$. Since each archived mechanism is scored on every subsequent evaluation, unrolling Equation (9) from $\ell_{i,b} = \ell_i^{\mathrm{birth}}$ gives

$$\ell_{i,t+1} = \lambda^n \ell_i^{\mathrm{birth}} + \sum_{s=b}^{t} \lambda^{t-s} \Delta_{i,s}, \qquad n = n_{i,t+1} = t - b + 1. \tag{36}$$

Hence

$$|\ell_{i,t+1}| \le \lambda^n |\ell_i^{\mathrm{birth}}| + \kappa \frac{1-\lambda^n}{1-\lambda}. \tag{37}$$

The inherited contribution decays with the number of the child's own scored forecasts. Because seeds and parentless innovations start at zero, and mutation and crossover preserve parental credit ranges, induction over births and evaluations gives

$$|\ell_{i,t}| \le \frac{\kappa}{1-\lambda} \tag{38}$$

throughout a run, assuming the stated birth rules and common λ, κ. The discount factor can be specified by a credit half-life h_{credit} through $\lambda = 2^{-1/h_{\mathrm{credit}}}$. Clipping applies to each increment $\delta_{i,t}$, not separately to the inherited value. Neither the bound nor inheritance guarantees predictive accuracy or equalizes evidence across mechanisms of different ages. An inaccurate descendant may retain an inherited advantage after its first score; the benefit of inheritance therefore requires empirical evaluation.

A.6 Expected improvement and quadrature

For $F \sim \mathcal{N}(\mu, v)$, $v > 0$, substitute $z = (F - \mu)/\sqrt{v}$ into $\mathbb{E}[(F - a)_+]$:

$$\begin{aligned}\mathcal{E}(\mu, v; a) &= \int_{(a-\mu)/\sqrt{v}}^{\infty} (\mu - a + \sqrt{v}\, z)\phi(z)\, dz \\ &= (\mu - a)\Phi\left(\frac{\mu - a}{\sqrt{v}}\right) + \sqrt{v}\,\phi\left(\frac{\mu - a}{\sqrt{v}}\right).\end{aligned} \tag{39}$$

This is the Gaussian EI expression (Jones et al., 1998). Conditional expectation under Equation (7) gives

$$\mathrm{EI}_{i,t}(\mathbf{x}) = \int_0^\infty \mathcal{E}(\mu_{0,t}(\mathbf{x}) + r_{i,t}(\mathbf{x})\beta, v_{0,t}(\mathbf{x}); a_t)\, p_{i,t}(\beta)\, d\beta. \tag{40}$$

The Gaussian EI inside the integral uses the conditional latent variance $v_{0,t}(\mathbf{x})$. Coefficient uncertainty is incorporated by integration against $p_{i,t}(\beta)$; neither observation noise nor the additional marginal variance from coefficient uncertainty is added to this conditional variance. For the baseline, the Gaussian EI expression applies directly with $\mu = \mu_{0,t}(\mathbf{x})$, $v = v_{0,t}(\mathbf{x})$, and threshold a_t.

Let $F_{i,t}^\beta$ be the coefficient posterior CDF and let $(u_j, \omega_j)_{j=1}^Q$ be Gauss–Legendre nodes and weights on $[0, 1]$. Changing variables to the coefficient quantile yields the approximation

$$\mathrm{EI}_{i,t}(\mathbf{x}) \approx \sum_{j=1}^{Q} \omega_j \mathcal{E}\Big(\mu_{0,t}(\mathbf{x}) + r_{i,t}(\mathbf{x})(F_{i,t}^\beta)^{-1}(u_j), v_{0,t}(\mathbf{x}); a_t\Big). \tag{41}$$

The approximation integrates coefficient uncertainty, without replacing the resulting expert distribution by a moment-matched Gaussian. In the implementation, this quadrature is evaluated in standardized units.

A.7 Expected improvement of the predictive mixture

Fix a round and a candidate $\mathbf{x}$, and let $\mathcal{I}_t = \{0\} \cup \mathcal{A}_t$. The weights in Equation (12) define the predictive mixture

$$q_t^{\mathrm{mix}}(u \mid \mathbf{x}) = \sum_{i \in \mathcal{I}_t} w_{i,t} q_{i,t}^{\mathrm{lat}}(u \mid \mathbf{x}). \tag{42}$$

The weights are nonnegative and sum to one. By linearity of integration,

$$\begin{aligned}\mathbb{E}_{F \sim q_t^{\mathrm{mix}}(\cdot|\mathbf{x})}[(F - a_t)_+] &= \int (u - a_t)_+ \sum_{i \in \mathcal{I}_t} w_{i,t} q_{i,t}^{\mathrm{lat}}(u \mid \mathbf{x})\, du \\ &= \sum_{i \in \mathcal{I}_t} w_{i,t} \int (u - a_t)_+ q_{i,t}^{\mathrm{lat}}(u \mid \mathbf{x})\, du = \sum_{i \in \mathcal{I}_t} w_{i,t}\, \mathrm{EI}_{i,t}(\mathbf{x}).\end{aligned} \tag{43}$$

No independence between experts is needed: each expert specifies an alternative predictive law. The implementation evaluates their EIs separately and takes this weighted sum. The identity establishes the criterion implied by the mixture and improvement utility; it does not establish an advantage over other acquisition strategies.

If the active set is empty, the criterion reduces to baseline analytic EI on $\mathcal{C}_t$. If every active expert has zero residual at each candidate, the mixture criterion equals baseline EI there because the weights sum to one. These statements concern the generated candidate set; they do not establish a maximizer over $\mathcal{G}_t$ or $\mathcal{X}$. The baseline reference mass $c_{0,t} = 1/2$ is not a lower bound on $w_{0,t}$. The weights measure predictive trust along the selected trajectory, including inherited credit; they are not posterior probabilities of scientific truth.

A.8 Baseline candidate construction

Candidate scores. The baseline GP constructs a set $\mathcal{C}_t$ with at most B inputs. Let $\alpha_t^{\mathrm{base}}(\mathbf{x})$ be its qLogEI value on the scrambled Sobol grid $\mathcal{G}_t$, computed with singleton batches and quasi-Monte Carlo samples (Ament et al., 2023). This score proposes candidates; Equation (14) ranks the resulting set for selection. The former uses the library's smoothed sample-based acquisition in raw objective units. The implementation computes mixture EI using analytic Gaussian EI and coefficient quadrature in standardized units. The two acquisition constructions need not rank grid points identically.

Local peaks and prominence. Build a symmetric k_{nn}-nearest-neighbor graph on the normalized grid: connect two vertices if either selects the other as a neighbor. From each vertex, repeatedly move to the largest-α_t^{base} vertex among itself and its neighbors, resolving ties by the smaller grid index. The terminal vertices define the local peaks.

Prominence measures how far a peak rises above the level at which it connects to a higher peak. Insert finite-valued vertices in descending α_t^{base} order. When two connected components merge at level α_{merge}, assign the lower peak $\mathbf{x}_{\mathrm{peak}}$ prominence $\alpha_t^{\mathrm{base}}(\mathbf{x}_{\mathrm{peak}}) - \alpha_{\mathrm{merge}}$. A surviving peak that is never eliminated has infinite prominence. Peaks with prominence at least ζ are retained for candidate admission.

Eligibility and candidate admission. Use normalized RMS distance $d_{\mathrm{RMS}}(\mathbf{u}, \mathbf{v}) = \|\mathbf{u}-\mathbf{v}\|_2/\sqrt{D}$ for both observed-point exclusion and candidate separation. A grid point is eligible if $\alpha_t^{\mathrm{base}}(\mathbf{x})$ is finite and its distance from every observed input is at least η_{obs}. Initially, it must also satisfy the relative acquisition floor

$$\alpha_t^{\mathrm{base}}(\mathbf{x}) \geq \max_{\mathbf{z}\in\mathcal{G}_t} \alpha_t^{\mathrm{base}}(\mathbf{z}) + \log \varrho, \qquad 0 < \varrho \leq 1. \tag{44}$$

Starting with $\mathcal{C}_t = \varnothing$, stop at capacity B and apply:

1. **Admit prominent peaks.** Add eligible prominent peaks in descending acquisition order.
2. **Fill remaining slots.** Scan the other eligible grid points in the same order, accepting a point only if it is sufficiently separated from every selected candidate. Repeat with decreasing separation thresholds, ending at zero.
3. **Relax the acquisition floor if needed.** If the set is still under capacity, remove Equation (44), reconsider the remaining prominent peaks, and repeat the filling step.

Observed-point exclusion is never relaxed. Candidate separation applies to filling; prominent peaks are admitted by prominence and eligibility.

Candidate ordering. Order the final candidates by descending α_t^{base}, breaking ties by the smaller grid index, and assign identifiers in that order.

Numerical settings. We evaluate baseline qLogEI (Ament et al., 2023) on 2048 scrambled Sobol points using 256 quasi-Monte Carlo samples and singleton batches. The Sobol seed is held fixed within each run. We use a symmetric 16-nearest-neighbor graph, a log-prominence threshold of $\log(1.1)$, and an initial relative acquisition floor of 10^{-6}.

For the conditional experts, we set the half-normal coefficient scale to $\tau = 1$ in standardized units and integrate coefficient uncertainty using 128 Gauss–Legendre nodes in posterior-quantile coordinates (Appendix A.6). The baseline and mechanism group each receive half of the reference mass.

A.9 Closed-loop algorithm and execution conventions

Algorithm 1 summarizes the optimization loop. Each round refreshes the active mechanisms, applies any scheduled revisions, selects an experiment, and updates credit from its outcome. In the pseudocode, $\mathcal{H}$ indexes all archived mechanisms m_i. Maps $L = \{i \mapsto \ell_i\}_{i\in\mathcal{H}}$ and $N = \{i \mapsto n_i\}_{i\in\mathcal{H}}$ store credits and scored-forecast counts. Their entries are updated in place, with the round index omitted: (ℓ_i, n_i) changes from $(\ell_{i,t}, n_{i,t})$ to $(\ell_{i,t+1}, n_{i,t+1})$ when y_t is scored.

Only previously scored mechanisms guide selection, but every accepted mechanism is evaluated on the new outcome. Predictions are archived before the experiment, with the GP parameters, scaling, and incumbent fixed. Newly accepted mechanisms therefore receive their first prospective score before they can influence a later decision. When the archive is empty, the LLM first proposes seed hypotheses.

The intervention schedule is defined by

$$\mathrm{Due}(t) = \mathbf{1}\{t > 1 \text{ and } (t-1) \bmod 3 = 0\}.$$

Seed initialization is handled separately when the archive is empty. In the pseudocode, $\mathrm{Due}(t)$ marks a scheduled revision, $\mathcal{O}_t$ specifies allowed proposal operations, and $\mathrm{Validate}(\mathcal{P}_t)$ returns accepted new mechanisms m_i, each assigned a fresh index i. The $\mathrm{Candidates}$ routine is defined in Appendix A.8. The fitted state $\mathcal{M}_t$ retains the shared GP, scaling $(\bar{y}_t, \gamma_t)$, and incumbent a_t. The pseudocode uses predictions and outcomes in objective units. Internally, the observation mean $\bar{y}_t = |\mathcal{D}_{t-1}|^{-1} \sum_{(\mathbf{x},y)\in\mathcal{D}_{t-1}} y$ and scale γ_t are computed before observing y_t and remain fixed when scoring it. Expert predictions use the conditional laws in Section 4.2.

Selection and participation. Exact ties in mixture EI favor the larger candidate identifier assigned by the construction in Appendix A.8. For active membership, credit ties favor existing active mechanisms, then descending identifier order. The acquisition and evolution capacities coincide, so every active mechanism participates in selection.

Proposal validation and evidence timing. Evolution is scheduled by observation count and uses bounded validation retries. Valid proposals may be retained even if others in the same batch fail. An empty archive triggers another initialization attempt on a later round. Revising only a mechanism's explanation preserves its identity and credit.

A newly accepted mechanism starts with the birth credit in Equation (11) and a zero scored-forecast count. It does not participate in the current decision, but its forecast at the selected input is archived before the outcome is observed. That outcome supplies its first score, making it eligible for active-set selection in the following round. GP parameters, outcome scaling, and the incumbent remain fixed when evaluating the archived forecast.

Recovery and final bookkeeping. Stored observation identifiers ensure that replay or recovery does not score or discount the same outcome twice. Birth-credit records restore inherited values once without creating forecast counts. After the evaluation budget is exhausted, final bookkeeping may also process a due evolution event and record any descendants' birth credits. This step makes no additional objective query and is omitted from Algorithm 1.

Adaptation and computational scope. Sharing fitted GP parameters restricts expert adaptation to the mechanism contribution and its coefficient uncertainty. The active-set capacity bounds the number of experts used to score candidates, but all accepted mechanisms are scored at the selected input. Maintaining prospective evaluation of a growing archive therefore adds computation even when the active set remains bounded.

Algorithm 1 PHEBO closed-loop optimization

Input: initial observations $\mathcal{D}_0$, domain context c, evaluation budget T, capacities K, B, credit parameters λ, κ.

Output: best observed input and mechanism archive with credits and counts.

$\mathcal{H} \leftarrow \varnothing$; $L, N \leftarrow \varnothing$; $\mathcal{A}_0 \leftarrow \varnothing$.
for $t = 1, \dots, T$ **do**
  Refresh participation and evolve hypotheses
  $\mathcal{A}_{t-1} \leftarrow \mathrm{TopK}_{i \in \mathcal{H}_{t-1}: n_{i,t} \geq 1} \ell_{i,t}$.
  if $\mathcal{H} = \varnothing$ or $\mathrm{Due}(t)$ **then**
    $\mathcal{O}_t \leftarrow \begin{cases} \{\mathrm{seed}\}, & \mathcal{H} = \varnothing, \\ \{\mathrm{mutation}, \mathrm{crossover}, \mathrm{innovation}\}, & \text{otherwise.} \end{cases}$
    $\mathcal{P}_t \leftarrow \mathrm{LLM}(\mathcal{D}_{t-1}, c, \mathcal{A}_{t-1}, L, N; \mathcal{O}_t)$.
    for each new mechanism m_i in $\mathrm{Validate}(\mathcal{P}_t)$ **do**
      $\mathcal{H} \leftarrow \mathcal{H} \cup \{i\}$; $\ell_i \leftarrow \ell_i^{\mathrm{birth}}$ (Eq. (11)); $n_i \leftarrow 0$.
    end for
  end if
  Construct experts and select an experiment
  $\mathcal{M}_t \leftarrow \mathrm{FitGP}(\mathcal{D}_{t-1})$ (Appendix A.2).
  $a_t \leftarrow \max_{(\mathbf{x},y) \in \mathcal{D}_{t-1}} y$.
  Construct experts for $i \in \mathcal{H}$ using $\mathcal{M}_t$ and $\mathcal{D}_{t-1}$ (Section 4.2).
  $\mathcal{C}_t \leftarrow \mathrm{Candidates}(\mathcal{M}_t, \mathcal{D}_{t-1}; B)$ (Appendix A.8).
  $\mathbf{w}_t \leftarrow \mathrm{Weights}(L, \mathcal{A}_{t-1})$ (Eq. (12)); set $w_{0,t} = 1$ if $\mathcal{A}_{t-1} = \varnothing$.
  Compute $\alpha_t^{\mathrm{mix}}(\mathbf{x})$ for $\mathbf{x} \in \mathcal{C}_t$ using Eqs. (14).
  $\mathbf{x}_t \in \arg\max_{\mathbf{x} \in \mathcal{C}_t} \alpha_t^{\mathrm{mix}}(\mathbf{x})$.
  Freeze forecasts, then observe
  $\mathcal{F}_t \leftarrow \{q_{i,t}^{\mathrm{obs}}(\cdot \mid \mathbf{x}_t)\}_{i \in \{0\} \cup \mathcal{H}}$.
  $\mathrm{Archive}(\mathcal{F}_t, \mathcal{M}_t, a_t, L)$.
  $y_t \leftarrow \mathrm{Evaluate}(\mathbf{x}_t)$; $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(\mathbf{x}_t, y_t)\}$.
  Update credit from the frozen forecasts
  for each $i \in \mathcal{H}$ **do**
    $\delta_{i,t} \leftarrow S_{i,t}(y_t) - S_{0,t}(y_t)$ using $\mathcal{F}_t$ and a_t (Eq. (8)).
    $\ell_i \leftarrow \lambda \ell_i + \mathrm{clip}(\delta_{i,t}, -\kappa, \kappa)$; $n_i \leftarrow n_i + 1$.
  end for
end for
return $\mathbf{x}^\star, \mathcal{H}, L, N$, where $(\mathbf{x}^\star, y^\star) \in \arg\max_{(\mathbf{x},y) \in \mathcal{D}_T} y$.

B BENCHMARK DATASETS AND EVALUATION ORACLES

We use deterministic evaluation oracles based on tabular data or released pretrained models. HPLC, Concrete, and CB maximize their responses, whereas PCE10 minimizes photodegradation. Record counts describe the underlying datasets, rather than the sequential evaluation budget.

B.1 HPLC

High-performance liquid chromatography (HPLC) is used here as a benchmark for calibrating automated sample handling. In the original ChemOS experiment, a robotic sampling system transferred a solution of 1,3,5-trimethoxybenzene in acetonitrile through a sample loop, an inline mixer, and an injection loop before HPLC analysis (Roch et al., 2020). The chromatographic peak area measures the amount of analyte delivered to the instrument. The optimization therefore tunes sample-transfer volumes, pump settings, and waiting time to maximize the analytical response. Efficient calibration of this workflow supports automated reaction monitoring by improving the delivery of samples to the measurement system. For optimization, the task couples several fluid-handling controls whose joint settings determine the observed response.

The HPLC dataset distributed through Olympus (Häse et al., 2021) contains 1,386 records with six continuous inputs. Filtering for finite values and the bounds in Table 1 retains 1,319 records at 957 distinct input configurations. Repeated measurements at the same configuration are aggregated while retaining their counts.

Table 1: HPLC search space.

Parameter	Unit	Range
Sample-loop volume	mL	$[0, 0.08]$
Additional volume	mL	$[0, 0.06]$
Tubing volume	mL	$[0.1, 0.9]$
Sample flow	mL/min	$[0.5, 2.5]$
Push speed	Hz	$[80, 150]$
Wait time	s	$[1, 10]$

For a query away from stored inputs, the oracle selects the 12 nearest distinct configurations in normalized Euclidean distance. Gaussian weights use bandwidths $(0.08, 0.26, 0.05, 0.15, 0.45, 0.50)$ in the parameter order of Table 1 and account for each configuration's measurement count. A query within 10^{-12} of a stored configuration returns its mean response. Outputs are clipped below at zero.

B.2 CONCRETE

Concrete mixture design seeks ingredient proportions that provide adequate mechanical performance for structural applications. Compressive strength depends on the amounts of cement, supplementary cementitious materials, water, aggregates, and chemical admixtures, as well as curing age. Their combined effects make strength prediction and formulation selection nonlinear problems (Yeh, 1998b). The benchmark thus represents an empirical formulation task in which optimization must identify promising combinations of ingredients from limited strength feedback. Its scientific relevance lies in exploring composition–property relationships while reducing the number of candidate mixtures requiring casting, curing, and mechanical testing.

We use the publicly distributed Concrete Compressive Strength dataset (Tripathi, 2020; Yeh, 1998a). The UCI description lists seven ingredient quantities in $\mathrm{kg/m^3}$, curing age in days, and compressive strength in MPa (Yeh, 1998a). Our benchmark excludes curing age and uses the seven continuous mixture quantities in Table 2. All 1,030 records pass the finite-value and bounds checks; they correspond to 427 distinct input configurations after age is excluded. Consequently, records sharing the same retained inputs can differ in curing age, and the resulting oracle combines responses across the ages represented in the source data. It should be interpreted as a composition-only empirical response, rather than strength at a specified curing age. The search space imposes the listed coordinate bounds, without an additional constraint on the total mixture mass or volume.

Table 2: Concrete search space. All ingredient quantities are in $\mathrm{kg/m^3}$.

Parameter	Range
Cement	$[102, 540]$
Blast-furnace slag	$[0, 359.4]$
Fly ash	$[0, 200.1]$
Water	$[121.8, 247]$
Superplasticizer	$[0, 32.2]$
Coarse aggregate	$[801, 1145]$
Fine aggregate	$[594, 992.6]$

The oracle uses inverse-distance weighting over the 12 nearest records in the original input coordinates, with exponent 2. Repeated records occupy separate neighbor slots. If the nearest distance is at most 10^{-12}, it returns the mean response of records within this tolerance among the selected neighbors.

B.3 CROSS-BARREL

The Cross-barrel task concerns the design of energy-absorbing structures studied with the Bayesian experimental autonomous researcher (BEAR) of Gongora et al. (2020). This platform combined automated 3D printing, robotic handling, and compression testing to explore a family of crossed-barrel geometries. Each structure consists of two platforms connected by hollow columns, with the design controlled by column number, twist angle, radius, and wall thickness. In the original study, toughness was quantified by the energy absorbed along the measured force–displacement curve. This response is relevant to protective structures that must absorb mechanical energy under loading. Geometric changes can alter buckling, self-contact, and failure behavior, making the relation between design parameters and toughness difficult to predict from individual parameters alone (Gongora et al., 2020).

The local dataset used in our benchmark contains 600 records at 600 distinct configurations, all within the search bounds. The geometry is specified by the number of columns n, twist angle θ, outer radius r, and wall thickness t. Our benchmark treats all four inputs as continuous, including n. Thus, the evaluator defines a continuous relaxation of the recorded design space: a noninteger column count is an oracle input and does not directly specify a fabricable structure. The optimization objective is to maximize the toughness response returned by this evaluator.

Table 3: Cross-barrel search space.

Parameter	Unit	Range
Number of columns, n	count	$[6, 12]$
Twist angle, θ	degrees	$[0, 200]$
Outer radius, r	mm	$[1.5, 2.5]$
Wall thickness, t	mm	$[0.7, 1.4]$

The oracle follows the HPLC construction: 12 nearest distinct configurations in normalized coordinates, Gaussian weighting, an exact-match tolerance of 10^{-12}, and nonnegative outputs. The bandwidths are $(0.126, 0.047, 0.038, 0.189)$ in the order (n, θ, r, t).

B.4 PCE10

The PCE10 task targets the photostability of multicomponent organic photovoltaic blends studied by Langner et al. (2020). The system combines the polymer donors P3HT and PTB7-Th (also known as PCE10) with the fullerene acceptor PCBM and the nonfullerene acceptor oIDTBR. In the original experiments, a liquid-handling robot prepared blend films whose absorption spectra were measured before and after 18 hours of illumination at one-sun intensity in ambient air. Photodegradation was quantified by the relative loss of integrated absorbance (Langner et al., 2020). Improving resistance to light-induced degradation is relevant to the stability of organic semiconductor materials. Because

blending can modify the stability of the constituent materials, the task tests the identification of stable compositions in a coupled mixture space. The response measures film photostability; it does not directly measure photovoltaic power-conversion efficiency or device operational lifetime.

We use the authors' released PTB7-Th (PCE10) system, with input order P3HT, PTB7-Th, PCBM, and oIDTBR as listed in Table 4. The released preprocessed dataset contains 1,040 records with positive degradation values at 1,020 distinct compositions; repeated measurements are retained. The objective is to minimize photodegradation.

Table 4: PCE10 search space. All inputs are mass fractions and sum to one.

Input	Material	Range
x_1	P3HT	$[0, 1]$
x_2	PTB7-Th (PCE10)	$[0, 1]$
x_3	PCBM	$[0, 1]$
x_4	oIDTBR	$[0, 1]$

The search domain is the simplex $\mathcal{X} = \{x \in [0, 1]^4 : \sum_{i=1}^{4} x_i = 1\}$, with three independent degrees of freedom. The baseline candidate grid is constructed by Euclidean projection of four-dimensional Sobol points onto this simplex. The domain includes pure-component vertices and does not impose an additional donor-to-acceptor ratio constraint.

Pretrained evaluation oracle. We evaluate compositions using the authors' released five-fold Bayesian neural network (BNN) ensemble for the PTB7-Th system.[1] Each network takes four inputs and has three hidden layers of 120 units, with leaky ReLU activations (slope 0.2) and a ReLU output. Our implementation loads the released parameters without retraining, standardizes inputs using the stored means and standard deviations, and returns degradation in the model's original output units.

For each network, we average 128 stochastic forward passes using a fixed bank of Gaussian draws for local reparameterization, then average across the five networks. The draws are generated with seed 1 and reused for every query and optimization run. Dropout is disabled and no observation noise is sampled, yielding a deterministic Monte Carlo approximation to the ensemble's predictive mean. To match the maximization convention in Method, PHEBO uses negative degradation as utility before outcome standardization.

[1] https://github.com/the-matter-lab/quaterny_opvs

C Additional Experiments and Analysis

C.1 Reachability Ceiling

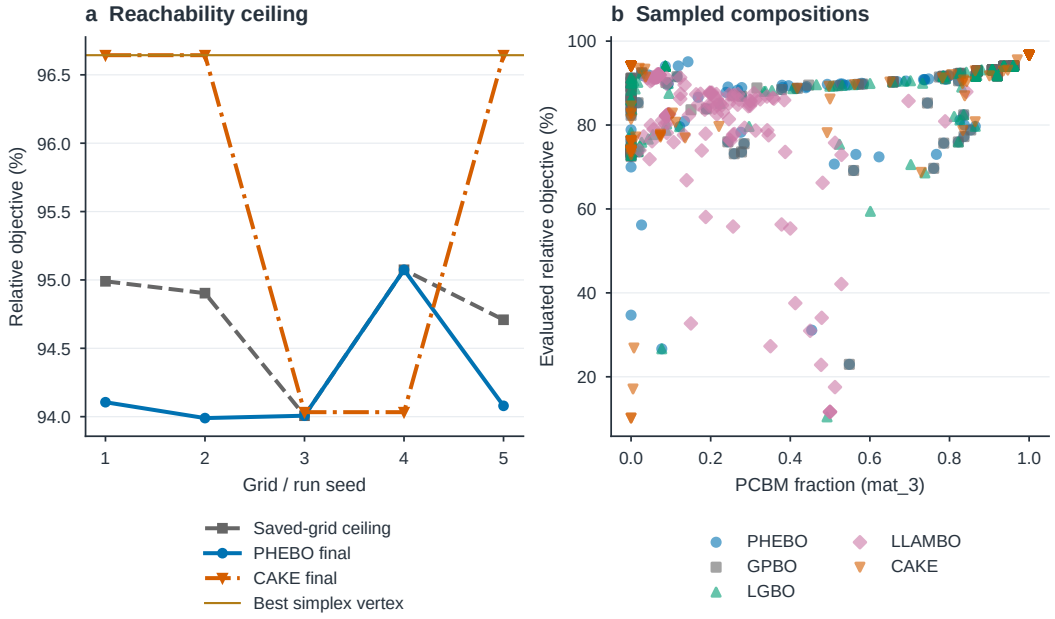

Figure 5: Candidate-set reachability and sampled compositions on the PCE10 task.

As shown in Figure 5, we investigate the factors limiting PHEBO's performance on the PCE10 task. Under our experimental setup, each run uses a fixed set of 2,048 Sobol candidate points. The gray dashed line in panel (a) denotes the highest score attainable within each candidate set. PHEBO reaches this ceiling for seeds 3 and 4, while leaving room for improvement for seeds 1, 2, and 5. Nevertheless, all five candidate-set ceilings remain below the score of the best simplex vertex, corresponding to pure PCBM, which CAKE reaches for seeds 1, 2, and 5.

Panel (b) shows the relationship between the PCBM mass fraction of queried compositions and their relative objective values. CAKE samples PCBM-rich compositions, including the pure-PCBM vertex at a mass fraction of 1.0, which attains the highest score shown under the fixed BNN oracle. This result reflects the simulated photodegradation objective and does not establish the superiority of pure PCBM under real experimental conditions. Thus, although PHEBO does not consistently attain its candidate-set ceiling, its suboptimal performance is partly attributable to the limited coverage of the fixed candidate set.

This access to boundary compositions is enabled by CAKE's use of BoTorch's `optimize_acqf`, which continuously optimizes the acquisition function rather than restricting candidate selection to a fixed Sobol set. With the simplex boundaries included in the feasible domain, the pure-PCBM vertex, $\mathbf{x} = (0, 0, 1, 0)$, is an admissible candidate and can be reached even when it is absent from the initialization points. This explains how CAKE can overcome the fixed candidate-set ceiling, although reaching this vertex in any particular run still depends on the learned acquisition landscape and the outcome of numerical optimization.

C.2 Cost-benefit analysis

Comparison at a common performance level. LLM guidance incurs additional request costs but may reduce the number of expensive objective evaluations. We retrospectively assess this trade-off for GPBO, LGBO, and PHEBO using five recorded HPLC runs per method. Let $b_{m,s}(t)$ denote the best peak area obtained by method m in run s after five initial evaluations and t optimization rounds. We use GPBO's final mean performance as the common reference level and define the first round at

which each method's mean trajectory reaches this level as

$$y_{\mathrm{ref}} = \frac{1}{5}\sum_{s=1}^{5} b_{\mathrm{GPBO},s}(25) \simeq 1906.73, \qquad t_m = \min\left\{ t \in \{0,\dots,25\} : \frac{1}{5}\sum_{s=1}^{5} b_{m,s}(t) \geq y_{\mathrm{ref}} \right\}. \tag{45}$$

Thus, t_m is the first crossing of the *mean trajectory*, not the mean of individual runs' hitting times.

To illustrate an expensive-experiment scenario, we assume $c_{\mathrm{eval}} = \$100$ per objective evaluation and $c_{\mathrm{LLM}} = \$1$ per recorded LLM query. These are illustrative unit costs, not measured laboratory expenses or API prices. With one evaluation per optimization round, the mean cost is

$$\overline{C}_m(t) = (5+t)c_{\mathrm{eval}} + \overline{Q}_m(t)c_{\mathrm{LLM}}, \tag{46}$$

where $\overline{Q}_m(t)$ is the mean cumulative recorded query count through round t. For PHEBO, we count all logged LLM requests, including those yielding rejected responses and validation retries. For LGBO, we count each round's recorded preference response. Operations after the stopping round are excluded, including PHEBO's final bookkeeping after round 25.

Table 5: HPLC costs at the common target of GPBO's final mean peak area (1906.73). Query counts, peak areas, and costs are five-run means at each method's first mean-trajectory crossing. Evaluations include initialization.

Method	Optimization rounds	Evaluations	LLM queries	Peak area	Cost ($)
GPBO	25	30	0	1906.73	3000.00
LGBO	11	16	11.0	1948.39	1611.00
PHEBO	10	15	9.8	1935.15	**1509.80**

As shown in Table 5, the mean PHEBO trajectory first crosses the reference level at round 10. At this round, each run has used 15 objective evaluations, and the mean recorded LLM request count is 9.8. Under the illustrative unit costs, the corresponding mean cost is \$1509.80, which is 49.67% below GPBO's \$3000.00 and 6.28% below LGBO's \$1611.00. The request counts across the five PHEBO runs through round 10 are 8, 12, 11, 10, and 8. For these observed crossing rounds and recorded request counts, the GPBO–PHEBO cost difference is $15c_{\mathrm{eval}} - 9.8c_{\mathrm{LLM}}$. Thus, the estimated saving is positive when $c_{\mathrm{eval}} > (9.8/15)c_{\mathrm{LLM}}$.

Theoretical LLM query budgets over 25 rounds. We compare theoretical query budgets over $T = 25$ optimization rounds, starting from a supplied initial dataset and assuming one request per prescribed LLM operation, with no retries. These budgets exclude the historical generation of the supplied initial points and post-budget bookkeeping.

GPBO requires no LLM queries. LGBO requests one preference per round, giving $Q_{\mathrm{LGBO}}(T) = T$ (Yuan et al., 2026, Section 3.3). Generating its warm-start points would require an additional Q_{init} queries; a total of 26 therefore assumes $Q_{\mathrm{init}} = 1$.

CAKE makes n_c crossover requests per round and one mutation request with probability p_m (Suwandi et al., 2025, Algorithm 1). Its query budget is

$$Q_{\mathrm{CAKE}}(T) = Tn_c + \sum_{t=1}^{T} Z_t, \qquad Z_t \sim \mathrm{Bernoulli}(p_m), \qquad \mathbb{E}[Q_{\mathrm{CAKE}}(T)] = T(n_c + p_m). \tag{47}$$

The original configuration uses $n_c = 5$ and $p_m = 0.7$ (Suwandi et al., 2025), yielding 142.5 expected queries over 25 rounds, with possible counts from 125 to 150. System-prompt updates require no separate request.

For PHEBO, one seed-hypothesis batch is requested before round 1, followed by updates before rounds $4, 7, 10, 13, 16, 19, 22, 25$. With an evolution interval of three and successful validation on the first attempt, the budget is

$$Q^{\mathrm{nominal}}_{PHEBO}(25) = 1 + \left\lfloor \frac{25-1}{3} \right\rfloor = 9. \tag{48}$$

Each request may return multiple proposals, so this count is not multiplied by the mechanism population size. These nominal counts assume successful validation on the first attempt. For PHEBO,

the total of nine comprises one seed-hypothesis request and eight scheduled evolution requests. Appendix C.3 reports logged evolution attempts, including retries but excluding seed-hypothesis generation. Under that accounting convention, the default configuration averaged 25.4 evolution requests over rounds 2–25.

Table 6: Theoretical LLM query budgets over 25 optimization rounds, excluding retries and generation of the supplied initial points. CAKE values are expectations because mutation is probabilistic.

Method / setting	Theoretical queries
GPBO	0
LGBO	25
CAKE, original paper ($n_c = 5$)	142.5
PHEBO, interval 3	9

C.3 Hyperparameter sensitivity

We assess sensitivity to mechanism capacity, candidate capacity, credit memory, and evolution frequency on HPLC using eight configurations, each varying one specified hyperparameter from the default.

Configurations and shared protocol. Let K denote the active mechanism capacity, B the candidate capacity, E the interval between scheduled LLM evolution calls, and h the credit half-life in observations, with $\lambda = 2^{-1/h}$. Table 7 lists the nine configurations, including the default. Each configuration uses seeds 1–5, the same five initial HPLC observations, and 25 subsequent BO rounds. Within each seed, all configurations share the three initial hypotheses from the archived default run. Subsequent hypotheses are generated anew under each configuration so that hypothesis evolution can respond to its observed outcomes and predictive feedback. Newly generated hypotheses must receive a prospective score before participating in acquisition. The LLM backbone, sampling temperature, objective oracle, GP settings, and numerical approximation budgets are held at their nominal default settings.

Table 7: Sensitivity configurations. Bold values indicate changes from the default. Each configuration uses five seeds and the same evaluation budget.

Configuration	K	B	E	h	κ
Default	6	5	3	6	1
K-small	**3**	5	3	6	1
K-large	**12**	5	3	6	1
B-small	6	**3**	3	6	1
B-large	6	**10**	3	6	1
Short memory	6	5	3	**3**	1
Long memory	6	5	3	**12**	1
Frequent evolution	6	5	**1**	6	1
Sparse evolution	6	5	**6**	6	1

Mechanism and candidate capacities. The grid $K \in \{3, 6, 12\}$ tests whether a smaller active set restricts participation by useful hypotheses and whether a larger set adds useful diversity or dilutes their influence. The active-set capacities used for hypothesis evolution and acquisition are both set to K, while the initial hypothesis count and per-call proposal limit remain fixed. Accepted hypotheses remain in the archive under the same predictive-credit policy.

The grid $B \in \{3, 5, 10\}$ tests whether a narrow candidate set limits belief-guided selection and whether a broader set improves performance. Because K and B are upper bounds, we also record the realized active-set and candidate counts. We exclude $B = 1$ from this sensitivity analysis because beliefs cannot alter selection by reranking a single candidate. The absence of belief guidance is examined separately in the Pure GP component ablation.

Credit memory and evolution frequency. We compare credit half-lives of 3, 6, and 12 observations, corresponding to $\lambda \approx 0.794$, 0.891, and 0.944. This grid contrasts rapid adaptation with longer retention of predictive evidence.

The intervals $E \in \{1, 3, 6\}$ represent frequent, default, and sparse LLM evolution. Starting from a shared initial population, subsequent calls occur at the beginning of rounds $1+E, 1+2E, \ldots$ within the budget. Evolution frequency refers to this call schedule, not how often mixture EI changes the GP-preferred choice. The proposal limit remains three per call, so changing E changes both the timing and the opportunity to propose hypotheses. These comparisons therefore assess scheduling policies rather than timing alone under a fixed proposal budget.

Parameters held fixed. We fix $\kappa = 1$ without testing alternative clipping thresholds or removing clipping. Replaying saved forecasts from the five default runs identifies 12 clipped increments among 1,697 mechanism-level score updates (approximately 0.71%), affecting eight of the 125 BO rounds. This low activation rate motivates prioritizing parameters that act throughout optimization. Appendix A.5 discusses this bounded-increment property.

We also fix the LLM temperature at 0.6, allow at most two terms and four distinct variables per mechanism, retain a composite-feature optimum width of 0.2 times its range, and permit at most three new hypotheses per call. Temperature controls generation randomness; the structural limits and width define the hypothesis representation; and the proposal limit controls another aspect of the generation budget.

Evaluation and interpretation. The primary outcomes are the final best objective and the area under the best-so-far curve divided by 25 (AUC/25). We normalize peak area using the fixed HPLC reference range $[0, 2372.24939]$, obtained after averaging repeated measurements at identical inputs, and express the normalized values as percentages. For seed s, let $b_s(t)$ denote the normalized best objective after round t, with initialization at $t = 0$. We compute

$$\mathrm{AUC}_s = \sum_{t=1}^{25} \frac{b_s(t-1) + b_s(t)}{2}.$$

Table 8 reports means and sample standard deviations of $b_s(25)$ and $\mathrm{AUC}_s/25$ across the five seeds. Saved analysis files contain individual values and within-seed differences from the default. Saved acquisition snapshots record active-set occupancy, candidate counts, acquisition weights, and candidate rankings for diagnostic inspection.

For cost comparisons, we count logged LLM evolution attempts in rounds 2–25, including retries. These counts exclude shared initial-hypothesis generation and post-budget calls.

Table 8: HPLC sensitivity results over five seeds (mean $\pm$ sample standard deviation). Final best and AUC/25 use the same normalized percentage scale. Bold and underlined values indicate the best and second-best means in each column, respectively. Higher is better for Final best and AUC/25, whereas lower is better for LLM requests. Requests count LLM evolution attempts in rounds 2–25, including retries, excluding shared initialization and post-budget calls.

Configuration	Final best (%)	AUC/25 (%)	LLM requests
Default	$\mathbf{88.35} \pm 3.89$	$\underline{79.86} \pm 4.73$	25.40 ± 5.22
K-small	86.90 ± 2.07	79.36 ± 3.71	28.00 ± 2.92
K-large	88.22 ± 0.87	78.57 ± 5.15	19.20 ± 1.92
B-small	83.16 ± 9.91	77.50 ± 7.28	24.60 ± 2.30
B-large	87.44 ± 1.69	76.36 ± 5.45	21.80 ± 0.45
Short memory	87.12 ± 5.05	78.44 ± 6.72	$\underline{18.40} \pm 2.70$
Long memory	86.35 ± 7.35	77.69 ± 8.38	22.80 ± 4.76
Frequent evolution	$\underline{88.28} \pm 1.39$	$\mathbf{80.24} \pm 2.29$	98.60 ± 6.58
Sparse evolution	83.91 ± 6.75	78.29 ± 6.83	$\mathbf{7.40} \pm 1.52$

Capacity sensitivity. The $K = 3$ configuration achieved a mean final objective of 86.90%, compared with 88.35% for the default ($K = 6$). The $K = 12$ configuration achieved a similar final mean (88.22%) with lower between-seed variability, but its AUC/25 was lower (78.57% versus 79.86%). A larger active mechanism capacity was therefore not associated with better mean performance on either metric in these runs.

Larger differences in the final mean were observed across the candidate-capacity settings. The $B = 3$ configuration had a final mean 5.19 percentage points below the default, with a sample standard deviation of 9.91 percentage points. The $B = 10$ configuration also had a lower final mean

(87.44%) and a lower AUC/25 (76.36%). Among the three tested candidate capacities, the default ($B = 5$) had the highest observed means on both metrics. These descriptive comparisons do not establish its optimality.

Credit memory and evolution cost. Shortening and lengthening the credit half-life lower the mean final objective by 1.23 and 2.00 percentage points, respectively; the longer half-life also increases variability. These results are consistent with a useful intermediate memory horizon, but five runs do not establish a statistically significant advantage for $h = 6$.

Evolution every round yields a final mean close to the default (88.28% versus 88.35%) and a modestly higher AUC/25 (80.24% versus 79.86%). However, it requires an average of 98.6 evolution requests, compared with 25.4 for the default, a 3.88-fold increase. Evolution every six rounds reduces the mean request count to 7.4 but lowers the final objective to 83.91%. The default therefore offers a more economical observed trade-off than per-round evolution, whereas sparse evolution reduces final quality on average.

For $E = 1, 3, 6$, mean API-reported token usage (input plus output) during rounds 2–25 is approximately 974,703, 197,435, and 56,354 tokens, respectively. For the two new interval variants, mean logged elapsed time from run opening through round 25 is 20.50 minutes for $E = 1$ and 2.14 minutes for $E = 6$. These times include local computation and service latency; they are descriptive measurements, not hardware-controlled benchmarks. Post-budget calls average 4.4, 5.6, and 2.4 for $E = 1, 3, 6$, respectively, and are excluded from the optimization request counts above.

D PROMPTS FOR BELIEF UPDATE

This appendix presents the prompts used to initialize and evolve the mechanism population in Section 4.3. The system message is reproduced verbatim; the user message is a schematic excerpt of the JSON input, reorganized into blocks for readability. Angle-bracketed placeholders replace task-specific and round-specific content. The template follows the composite-feature, predictive-credit archive configuration used in this paper; serialization and detailed validation fields are abbreviated.

System prompt. The fixed system message defines the LLM's role as a proposer of interpretable hypotheses, constrains its output to typed claims, and leaves numerical scoring and experiment selection to the optimization pipeline.

```
You propose a compact population of scientifically interpretable
mechanisms for Bayesian optimization. Return JSON only. You may
propose only the typed claims in the supplied schema. Do not score,
select acquisition cards, or invent observations.
```

User prompt template. The user message supplies domain context, parameter bounds, the objective, and available observations. Active mechanisms are accompanied by their claims, explanations, predictive credits, and prediction counts; pending mechanisms and previously used claims provide additional context. The requirements specify admissible evolutionary operations and explain how credit should be interpreted. On a validation retry, the message also includes the invalid response, error feedback, and already accepted actions. A separate JSON response schema enforces the output structure.

```
[Task and background]
- Task: <seed initialization or evolution>; round: <round index>.
- Domain context: <background, parameter meanings, and units>.
- Search space: <parameters and bounds>; objective: <target/direction>.

[Observations and belief state]
- Observations: <available input-output records>.
- Active mechanisms: <IDs, claims, explanations, credits, counts>.
- Pending mechanisms: <unscored mechanisms>.
- Allowed parents: <active IDs>; forbidden IDs/claims: <used entries>.

[Claim grammar and evolution]
- Use supplied variables/target and <allowed feature catalog>.
- One term: single; two terms: conjunction. Respect <variable limit>.
- Relations: increasing, decreasing, optimum; centers stay in bounds.
  Composite optimum radii must match <catalog radii>.
- Seed: no parents, initialization only; innovation: no parents,
  after initialization only. Mutation: one active parent, one edit.
- Crossover: one term per active parent; refine at most one term
  without changing its input expression.
- Optional description revisions change only active explanations;
  keep them separate from proposals and empty at initialization.
- Avoid duplicate claims; prefer interpretable, testable hypotheses.

[Predictive credit]
- Credit measures prediction relative to the GP, not probability
  or proof of optimization benefit; never invent credits/counts.
- Mutation inherits parental log credit; crossover inherits its
  parental mean. Seeds/innovations start at zero.
- Inheritance is a lineage prior. New mechanisms have zero counts
  and weights; activation requires a scored future observation.
- Dormant mechanisms remain scored and can return to active status.
- Credit policy: <evidence half-life and improvement scoring rule>.

[Output and validation]
- Return JSON: decision_summary, description_revisions, proposals.
- First attempt: 1-3 proposals; one proposal-plan entry per proposal.
- New ID prefix: <required prefix>; use the supplied response schema.
- Retry feedback: <invalid response, errors, suggested fixes>.
  Fill <remaining slots>; do not repeat <staged valid actions>.
```

Output schema. The response contains a population diagnosis, optional description revisions, and new mechanism proposals. The following compact schema summarizes the actual structured output; angle brackets denote typed placeholders, and | denotes alternatives rather than literal JSON content. All fields of the selected object type are required, including nullable fields, and additional properties are disallowed. Credit, prediction counts, and acquisition weights are maintained by the optimization pipeline and are not output fields.

```
{
  "decision_summary": {
    "population_diagnosis": "<brief assessment of current beliefs>",
    "proposal_plan": ["<one plan entry per returned proposal>", ...]
  },
  "description_revisions": [{
    "target_id": "<active mechanism ID>",
    "mechanism": "<revised explanation>", "reason": "<revision reason>"
  }, ...],
  "proposals": [{
    "id": "<new mechanism ID with required prefix>",
    "claim": {
      "target": "<objective name>",
      "terms": [<Term>, ...], "composition": "<single | conjunction>"
    },
    "mechanism": "<scientific explanation of the proposed claim>",
    "lineage": {
      "operation": "<seed | mutation | crossover | innovation>",
      "parents": [<parent IDs>], "created_round": <nonnegative integer>,
      "mutation_edit": <MutationEdit | null>,
      "contributions": [<Contribution>, ...]
    }
  }, ...]
}
```

A `Term` uses either an original variable or a catalogued composite feature, with the same relation fields. For `optimum`, the center and positive radius are numeric; both are `null` for monotone relations. One term requires `single`; two require `conjunction`.

```
Term = {
  "variable": "<parameter name>",
  "relation": "<increasing | decreasing | optimum>",
  "center": <number | null>, "radius": <positive number | null>
}
For a composite term, replace "variable" with:
  "feature": {"op": "<product | ratio>",
              "variables": ["<first parameter>", "<second parameter>"]}

MutationEdit = {
  "kind": "<change_relation | adjust_optimum | replace_term |
            add_term | remove_term>",
  "before": <Term | null>, "after": <Term | null>
}
Contribution = {
  "parent_id": "<active parent ID>", "source_term_index": <0 | 1>,
  "child_term_index": <0 | 1>,
  "transformation": "<copied | relation_refined | optimum_refined>"
}
For an original-variable contribution, "source_variable":
  "<parameter name>" may replace "source_term_index".
```

Seeds and innovations use `parents=[]`, `mutation_edit=null`, and `contributions=[]`. Mutation specifies one parent and one edit, with empty contributions: `add_term` uses `before=null`, `remove_term` uses `after=null`, and other edits require both terms. Crossover specifies two distinct active parents, a null edit, and two contributions mapping those parents to different child terms; at most one contribution may be refined. Description revisions use an empty array when absent and at initialization. The first attempt returns one to three proposals; retries return only the remaining actions. The validators also check bounds, novelty, and consistency with the cited parents.

E CASE STUDY: BELIEF EVOLUTION AND SAMPLING INTERVENTIONS

E.1 TRAJECTORY SELECTION AND BACKGROUND

We examine the seed-2 trajectory from the experiments on HPLC task, in which the five runs share five initial observations, whose best peak area is 1332.96, followed by 25 sequential evaluations of the deterministic HPLC oracle (Appendix B.1). Each run uses a fixed Sobol grid of 2,048 points, at most five basin candidates, six active mechanisms, and scheduled evolution every three rounds. Credit uses improvement-censored scoring, a half-life of six observations, clipping at one, and birth-credit inheritance.

For this analysis, an *intervention* means that mixture EI selects a different candidate from the maximizer of analytic baseline GP EI on the *same available candidate set*. This is a local decision comparison, not a comparison with a separately executed GP-only trajectory.

E.2 A TRACEABLE LINEAGE OF EXECUTABLE BELIEFS

Table 9 names six hypotheses used below. Let v_ℓ, v_t, and v_a denote sample-loop, tubing, and additional volumes in mL, and let p denote push speed in Hz. $\mathrm{Opt}(z; c, \rho)$ denotes the implemented preference centered at c with width ρ; these are hypothesis parameters, not estimated physical optima or confidence intervals. The conjunction symbol combines two preference terms as in Equation (4).

Table 9: Selected mechanisms in seed 2. IDs are those in the archived records; R denotes the birth round.

Alias	Archived ID	R	Executable claim	Origin
H1	`M-R001-A01-001`	1	$\mathrm{Opt}(v_\ell; 0.04, 0.02)$	Seed
H2	`M-R001-A01-003`	1	$\mathrm{Opt}(v_t; 0.45, 0.20) \wedge \mathrm{Dec}(v_a)$	Seed
H3	`M-R004-A01-001`	4	$\mathrm{Opt}(p; 115, 35)$	Replace H1's term
H4	`M-R004-A01-002`	4	$\mathrm{Opt}(v_\ell; 0.04, 0.02) \wedge \mathrm{Opt}(v_t; 0.45, 0.20)$	Crossover H1, H2
H5	`M-R007-A01-003`	7	$\mathrm{Opt}(p; 130, 20)$	Adjust H3's optimum
H6	`M-R013-A01-002`	13	$\mathrm{Opt}(v_\ell; 0.04, 0.02) \wedge \mathrm{Opt}(p; 130, 20)$	Crossover H1, H5

The lineage illustrates both revision and retention. H3 replaces H1's sample-loop term with a push-speed term; H5 subsequently shifts and narrows that preference. H6 combines the original sample-loop preference with H5's push-speed preference. These operations create distinct identities: H3 remains available when H5 is introduced, allowing both preferences to compete through subsequent forecasts. Across rounds 1–25, the archive accepts 26 mechanisms: three seeds, six mutations, four crossovers, and 13 parentless innovations. Two further mechanisms are created by the terminal update labeled round 26; they have no subsequent evaluation and are excluded from explanations of the 25 sampling decisions.

Credit inheritance and delayed participation. At round 1 all three seeds are pending, so the baseline alone chooses the first point. H3 is born at round 4 with H1's credit, 1.4114, but has zero weight in that round. After its first scored forecast it participates at round 5 with credit 1.2168 and weight 0.1834. H5 similarly inherits 0.8364 at round 7 and first participates at round 8. H6 inherits the mean parental credit, 0.4685, at round 13 and first participates at round 14. Thus inherited trust affects the starting credit, while admission still requires the child's own prospective evaluation.

Retention does not imply permanent influence. After the round-19 improvement, H3's credit rises from 0.1716 to 0.7173, and H6's from 0.2403 to 0.8546; these updated values are used at round 20. H1 instead falls from 0.2547 to 0.1138 and becomes dormant at round 20. It re-enters the active set at round 23, before becoming dormant again at round 25. Figure 6 distinguishes continuous credit tracking from the discrete admission decision. Credit measures the expert's forecast of the realized outcome, so even an expert that favored a different candidate can gain credit at the point actually evaluated.

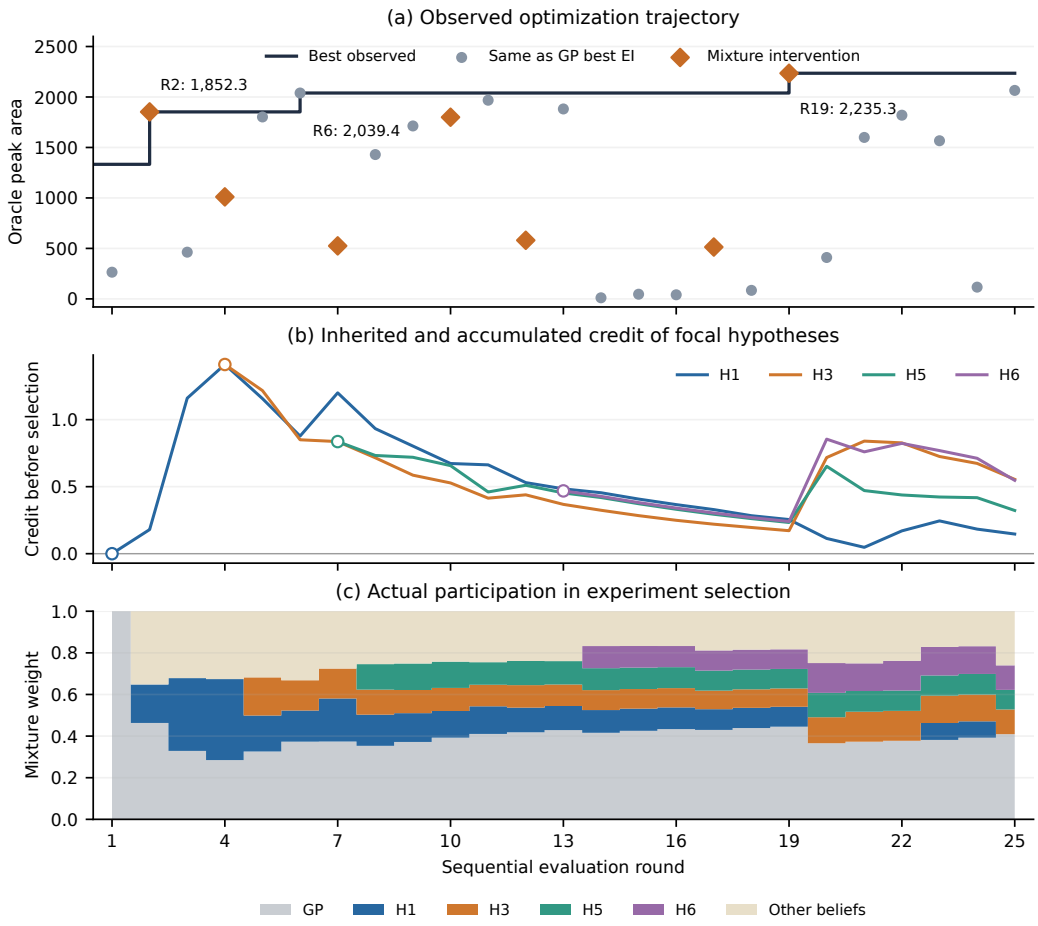

Figure 6: Seed 2, reconstructed from archived observations and acquisition snapshots. (a) Actual oracle responses and best-so-far response; orange diamonds mark all seven interventions. (b) Credit *before* each selection for the focal lineage in Table 9; open circles mark birth, including inherited initial credit. (c) Actual mixture weights; other active mechanisms are grouped together. Newborn hypotheses have zero weight until a future observation is scored, and dormant hypotheses retain credit while having zero selection weight.

E.3 How mixture EI changes the sampled point

Table 10 reports every intervention, including those without immediate improvement. The best response rises from 1332.96 to 1852.35 at round 2, to 2039.40 at round 6 (without an intervention), and to 2235.27 at round 19. The final best is 67.69% above the shared initial best. The other five interventions do not improve the incumbent. In particular, round 17 has the largest relative mixture-EI margin among the interventions, yet returns only 513.51: an acquisition preference is not a guarantee of realized improvement.

Early intervention: a seed hypothesis changes the ranking. At round 2, the baseline favors `C-R002-01`, whereas the mixture selects `C-R002-03`, ranked third by GP EI. Their sample-loop volumes are 0.07198 and 0.03446 mL, respectively; the latter is nearer H1's hypothesized center of 0.04 mL. H1's conditional expert assigns EI 0.11401 to the baseline choice and 0.35721 to the selected point. With weight 0.18483, its contribution to the selected-minus-baseline mixture-EI difference is $+0.04495$. This outweighs the weighted baseline loss of -0.01558 and the net opposition of the remaining experts, leaving a margin of $+0.01599$. The queried outcome is 1852.35, improving the incumbent by 519.39. After scoring this outcome, H1's credit increases from 0.1799 to 1.1603 for round 3.

Table 10: All interventions in seed 2. Rank is the selected candidate's rank by analytic GP EI. EI values are in each round's standardized utility units; compare them within a row, not across rounds. B denotes the GP-best candidate and S the selected candidate.

Round	Rank	GP EI(B)	GP EI(S)	Mix EI(B)	Mix EI(S)	Outcome	Gain
2	3	.184407	.150765	.181598	.197584	1852.35	519.39
4	3	.130490	.095459	.146572	.146953	1010.51	0
7	2	.056562	.047863	.058389	.060283	525.12	0
10	2	.026490	.022503	.025255	.027718	1799.55	0
12	3	.008431	.006767	.007240	.007921	580.52	0
17	2	.001303	.001107	.001215	.002170	513.51	0
19	2	.044464	.044010	.050765	.051023	2235.27	195.87

Late intervention: a retained parent changes the decision. At round 19, the mixture selects `C-R019-02` over `C-R019-01`. In the parameter order of Table 1, the selected input is

$$\mathbf{x}_{19} \simeq (0.05511, 0.04477, 0.30716, 2.27609, 106.51859, 7.41820).$$

Its push speed is lower than the baseline choice's 136.26050 Hz. H3, the retained 115 Hz preference, assigns EI 0.07181 to the selected point versus 0.04521 to the baseline choice. Its weight is only 0.08808, the smallest among the six active mechanisms, but its weighted EI difference is $+0.002343$. The baseline and all five other active mechanisms favor the alternative, contributing -0.002085 in total. The resulting $+0.000258$ margin reverses the decision and the observed outcome raises the best response by 195.87. The three new round-19 hypotheses are pending and have no role in this selection. Holding the saved forecasts fixed and removing H3's mixture term makes `C-R019-01` the winner again; the analogous removal of H1 reverses the round-2 intervention. These are local score decompositions, not reruns of the optimization process.