Expert-Guided Bayesian Optimization for Sustainable Protein Formulation

Anonymous Authors[1]

Abstract

Autonomous scientific discovery systems increasingly use LLMs to narrow design spaces before experiments are run, but this practice is double-edged: when the LLM is right, sample efficiency can improve dramatically; when it is wrong, the system can underperform random search. We formalize Expert-Guided Bayesian Optimization (EGBO), in which an expert, e.g. a human or LLM, selects a low-dimensional subspace for BO and may adaptively expand it over time. We decompose EGBO's suboptimality into a selection gap and an optimization gap, and characterize the coverage–dimension tradeoff governing when expert guidance helps. To support *in silico* prototyping before costly real-world deployment, we introduce FORMULATEBENCH, a suite of 24 plant-based formulation tasks, on which LLM-guided EGBO outperforms all tested baselines. When deployed to optimize two plant-based dairy products, EGBO improves utility, as assessed by a trained human panel, by 29% and 26% in 10 iterations each. In a comparison with a professional human food scientist given the same time budget, EGBO achieved near-perfect utility of 0.992, vs. 0.850 for the food scientist.

1. Introduction

Scientific discovery is increasingly moving toward closed-loop workflows in which AI systems propose candidates and experimental platforms evaluate them. Recent systems use LLMs and tool-augmented agents to plan experiments, generate hypotheses, and guide molecular or genetic design, often in conjunction with laboratory automation or simulation (Boiko et al., 2023; M. Bran et al., 2024; Lu et al., 2024; Roohani et al., 2025; Ghafarollahi & Buehler, 2024). In many such pipelines, an LLM or human expert performs a critical act of judgment by restricting a vast design space to a tractable subset before expensive evaluation begins. When this restriction captures the relevant variables, downstream optimization can be highly sample-efficient; when it does not, the search is confined to a misspecified subspace, leading to degraded performance and, in some cases, worse outcomes than simple baselines such as random sampling.

The downstream optimization subroutine in many such loops is Bayesian optimization (BO) (Snoek et al., 2012; Frazier, 2018), whose sample efficiency deteriorates sharply in the ambient dimension N (Srinivas et al., 2012; Bull, 2011). Many of the most consequential applications - plant-based food formulation with thousands of candidate ingredients (van den Bedem et al., 2026), combinatorial fragment libraries in drug discovery (Irwin et al., 2012), alloy and catalyst design (Raccuglia et al., 2016) - are inherently high-dimensional. Yet a high-quality solution can often be found using a small subset of the available variables. For example, a successful plant-based formulation typically uses 5-15 ingredients. Importantly, this is a statement about the existence of a good sparse solution, not about the function itself: f generally depends on all coordinates, and the best possible recipe may use a few more ingredients than the best sparse one. However, the gap between the two is typically small, and the sparse solution is often preferable on other grounds, e.g. consumer acceptance.

Our work is particularly motivated by the challenges of discovering sustainable proteins: animal-free protein sources such as plant-based, fermentation-enabled, and cultivated meat and dairy. Animal agriculture accounts for roughly 16.5% of global greenhouse gas emissions and is a leading driver of land use change, biodiversity loss, and pandemic risk (Twine, 2021; Espinosa et al., 2020). Plant-based alternatives can substantially reduce this footprint (Poore & Nemecek, 2018), but adoption depends on matching the sensory and textural properties of animal-derived products closely enough for consumer acceptance. Formulating a plant-based product therefore amounts to solving an expensive black-box optimization problem, with each candidate recipe requiring physical preparation and human evaluation.

In the framework we introduce, Expert-Guided Bayesian Optimization (EGBO), a domain expert - human or LLM - proposes which variables to include, and BO operates within

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Submitted to the AI for Science workshop (ICML 2026). Do not distribute.

the resulting subspace; variables may be selected once or added adaptively. This decouples the problem into *support discovery*, in which the expert identifies variables appearing in a good sparse solution, and *low-dimensional optimization*, in which BO converges efficiently within the discovered subspace. The decoupling isolates expert reliability as two interpretable parameters: a per-variable recall p and a false-positive rate q. Our contributions are as follows:

1. **A general framework for expert-guided BO.** We formalize one-shot expert-guided variable selection as currently practiced and extend it to an adaptive variant in which the active set grows monotonically over time. We prove a simple one-shot bound that decomposes EGBO's error into an expert-dependent selection gap and a low-dimensional BO optimization gap, and analyze the coverage-dimension tradeoff governing adaptive expansion.

2. **A public benchmark for *in silico* prototyping.** To support practitioners in evaluating formulation methods before committing to expensive laboratory work, we introduce FORMULATEBENCH, a suite of 24 plant-based formulation tasks (14 meat, 10 dairy) grounded in real nutritional targets. LLM-guided EGBO outperforms all baselines, including vanilla BO, random search, REMBO, SAASBO, TuRBO, and SEBO, indicating that LLMs can be useful experts in this domain.

3. **Real-world deployment and human comparison.** We deploy LLM-guided EGBO to optimize two plant-based dairy products with a food company, improving sensory utility, as assessed by a trained human panel, by 29% and 26% in 10 iterations. Moreover, in a controlled comparison, EGBO outperformed a professional food scientist by 17%. These deployments demonstrate the practical efficacy of EGBO in a closed-loop experimental setting and highlight directions for future methodological work.

All code is released publicly at LINK REDACTED.

2. Related Work

High-dimensional Bayesian optimization. A number of approaches have been developed to scale BO to the high dimensional setting. Random embedding methods such as REMBO (Wang et al., 2016) project the search space into a low-dimensional subspace, but the projection is uninformed and may distort the objective. Trust-region methods like TuRBO (Eriksson et al., 2019) restrict optimization to local regions, improving scalability but not directly exploiting sparsity. Sparse axis-aligned subspace methods such as SAASBO (Eriksson & Jankowiak, 2021) learn which variables matter from data by placing sparsity-inducing priors on kernel lengthscales. Sparsity Exploring BO (SEBO) (Liu et al., 2023) targets sparse solutions via homotopy relaxation of the L_0 norm. These methods share a practical limitation: the relevant subspace must be inferred from evaluations, consuming budget that EGBO saves by obtaining the subspace from the expert upfront.

LLMs in optimization and experimental design. A growing body of work uses LLMs as components of optimization or experimental design loops. LLAMBO (Liu et al., 2024) uses LLMs as surrogate models and acquisition functions for BO, while OPRO (Yang et al., 2024) treats the LLM itself as the optimizer over natural-language solution descriptions. Closer to our setting, GPTuner (Lao et al., 2024) uses LLMs to select database configuration knobs and propose search ranges before optimization. EGBO is closest in spirit to GPTuner - both use LLMs for upfront variable and range selection - but additionally provides performance guarantees parameterized by expert quality, supports adaptive expert querying, and targets general optimization problems rather than database configuration specifically. We discuss additional related work in Appendix B.

3. Expert-Guided Bayesian Optimization

Though directly motivated by sustainable protein formulation, EGBO applies to any setting in which an expensive black-box function $f : \mathcal{X} \to \mathbb{R}$ must be maximized over a high-dimensional domain $\mathcal{X} = \prod_{i=1}^{N}[0, u_i^{\max}] \subseteq \mathbb{R}^N$, and where high-quality solutions are expected to depend on only a small subset of the N variables. In the sustainable protein setting, f could be the panel-assessed sensory score of a plant-based formulation parameterized by ingredients and processing steps.

EGBO maintains an active set of variables $S_r \subseteq [N]$ at each round r, and optimizes over the restricted domain $\mathcal{X}_{S_r} = \{x \in \mathcal{X} : x_i = 0 \text{ for all } i \notin S_r\}$. The algorithm has Q rounds. In the first round, the expert proposes an initial active set S_1. In later rounds, the expert may inspect the optimization history and add more variables. The active set grows monotonically: $S_1 \subseteq S_2 \subseteq \cdots \subseteq S_Q$. $Q = 1$ and $Q > 1$ correspond to one-shot and adaptive EGBO respectively. Let T be the total evaluation budget, including n_{init} initial Sobol evaluations. We allocate the remaining $T - n_{\text{init}}$ evaluations across the Q BO rounds as $T_1, \ldots, T_Q$, with $n_{\text{init}} + \sum_{r=1}^{Q} T_r = T$. Earlier evaluations remain valid when the active set expands, because $\mathcal{X}_{S_r} \subseteq \mathcal{X}_{S_{r+1}}$. Algorithm 1 describes the procedure.

3.1. Theoretical Analysis

To provide a simple analysis of the performance of Algorithm 1, we make three assumptions. A central feature of sustainable protein formulation problems is that high-quality

Algorithm 1 Expert-Guided Bayesian Optimization (EGBO)

Require: Domain $\mathcal{X}$, expert $\mathcal{E}$, total budget T, expert rounds Q, initial Sobol budget n_{init}, round budgets $T_1, \dots, T_Q$ satisfying $n_{\text{init}} + \sum_{r=1}^{Q} T_r = T$

Initialize dataset $\mathcal{D} \leftarrow \emptyset$
Query expert $\mathcal{E}$ for initial active set $S_1 \subseteq [N]$
Draw n_{init} Sobol points in $\mathcal{X}_{S_1}$, evaluate them, and add them to $\mathcal{D}$
Fit the Gaussian process surrogate using $\mathcal{D}$
for $r = 1, \dots, Q$ **do**
  for $t = 1, \dots, T_r$ **do**
    Choose $x \in \mathcal{X}_{S_r}$ using the BO acquisition function
    Evaluate $y = f(x) + \varepsilon$
    Add (x, y) to $\mathcal{D}$ and update the Gaussian process surrogate
  end for
  if $r < Q$ **then**
    Query expert $\mathcal{E}$ with the current history $\mathcal{D}$ for additional variables $A_{r+1} \subseteq [N] \setminus S_r$
    Set $S_{r+1} \leftarrow S_r \cup A_{r+1}$
  end if
end for
return $\hat{x}_T \in \arg\max_{(x,y)\in\mathcal{D}} y$

sparse solutions are rarely unique: a target sensory profile may be achievable via multiple disjoint ingredient sets.

Assumption 1 (Family of sparse approximate optima)**.** *There exist $k \in \mathbb{N}$ and $\eta \geq 0$ such that the family*

$$\mathcal{R}_\eta = \big\{R \subseteq [N] : |R| \leq k, \; \exists\, \tilde{x}^{(R)} \in \mathcal{X} \text{ with } \operatorname{supp}(\tilde{x}^{(R)}) \subseteq R \;\; (1)$$
$$\text{and } f(\tilde{x}^{(R)}) \geq f(x^*) - \eta\big\}$$

is non-empty. We refer to $\tilde{x}^{(R)}$ as a witness *for support R: a specific k-sparse near-optimum supported on R. We call η the* sparsity gap*, k the* target sparsity level*, and $M = |\mathcal{R}_\eta|$ the* multiplicity*. The* support union *$\mathcal{U}_\eta = \bigcup_{R\in\mathcal{R}_\eta} R$ collects all variables that appear in some sparse near-optimum, with $\kappa = |\mathcal{U}_\eta|$.*

Assumption 1 is strictly weaker than assuming a unique sparse global optimum: it requires only that *some* k-sparse point achieves near-optimal value.

We characterize expert quality relative to $\mathcal{U}_\eta$:

Assumption 2 (Expert quality)**.** *At each round q, the expert's additions to S_{q+1} are independent across coordinates conditional on S_q, with fresh draws each round. Every variable $i \in \mathcal{U}_\eta$ not yet in S_q is added with probability $p \in (0, 1]$ (*recall*); every $j \notin \mathcal{U}_\eta$ not yet in S_q is added with probability $q_{\text{fp}} \in [0, 1)$ (*false-positive rate*). In the one-shot setting ($Q = 1$) this reduces to: each $i \in \mathcal{U}_\eta$ is included in S with probability p and each $j \notin \mathcal{U}_\eta$ with probability q_{fp}.*

We additionally require a guarantee from the BO subroutine used within each restricted domain.

Assumption 3 (BO subroutine guarantee)**.** *For any active set $S \subseteq [N]$, when the BO subroutine is run on $\mathcal{X}_S$ with T_S evaluations whose acquisition functions are each maximized over $\mathcal{X}_S$, it returns $\hat{x}_{T_S}$ satisfying $f_S^\star - f(\hat{x}_{T_S}) \leq \varepsilon_{\text{BO}}(T_S, |S|)$ with probability at least $1 - \delta_{\text{BO}}$, where $\varepsilon_{\text{BO}}(T, d)$ is non-increasing in T and non-decreasing in d.*

We use this as an abstract finite-sample guarantee for the BO subroutine. Different BO algorithms satisfy different versions of such a guarantee under different smoothness, kernel, and noise assumptions. We state our results in terms of the generic ε_{BO}. Throughout, we assume the expert's variable selection randomness and the BO subroutine's randomness are independent.

EGBO helps only if two events occur. First, the expert must choose an active set that contains a good sparse solution. Second, BO must successfully optimize inside that active set. We capture this with the decomposition

$$f(x^*) - f(\hat{x}_T) = \underbrace{f(x^*) - f_S^\star}_{\text{selection gap}} + \underbrace{f_S^\star - f(\hat{x}_T)}_{\text{optimization gap}}, \quad (2)$$

where $f_S^\star = \max_{x\in\mathcal{X}_S} f(x)$. The selection gap measures the cost of restricting the search to the expert-selected subspace. The optimization gap measures BO's suboptimality within that subspace. If S contains the support of a sparse near-optimal witness $R \in \mathcal{R}_\eta$, then the restricted domain $\mathcal{X}_S$ contains a point whose value is within η of the global optimum. Therefore, $f(x^*) - f_S^\star \leq \eta$. The remaining error is then controlled by the BO subroutine, which operates in dimension $|S|$ rather than the ambient dimension N.

Theorem 1 (One-shot EGBO)**.** *Suppose Assumptions 1, 2, and 3 hold. Consider one-shot EGBO: the expert is queried once to obtain an active set S, and BO is then run on $\mathcal{X}_S$ for T evaluations. Fix any witness support $R \in \mathcal{R}_\eta$. With probability at least $p^{|R|}(1 - \delta_{\text{BO}}) \geq p^k(1 - \delta_{\text{BO}})$, the returned point $\hat{x}_T$ satisfies*

$$f(x^*) - f(\hat{x}_T) \leq \eta + \varepsilon_{\text{BO}}(T, |S|). \quad (3)$$

Moreover, under Assumption 2, $\mathbb{E}[|S|] = \kappa p + (N - \kappa)q_{\text{fp}}$, where $\kappa = |U_\eta|$.

The theorem says that one-shot EGBO succeeds when the expert-selected active set contains a sparse near-optimal support. In that case, the total error is at most the sparsity gap η plus the error from running BO in the lower-dimensional space $\mathcal{X}_S$. When there are many different sparse near-optimal supports, the probability of covering at least one of them can be much higher. If $\mathcal{R}_\eta$ contains M' pairwise disjoint supports of size at most k, then

$$\Pr(\exists R \in \mathcal{R}_\eta : R \subseteq S) \geq 1 - (1 - p^k)^{M'}.$$

Proposition 1 (Coverage under adaptive expansion)**.** *Suppose EGBO performs Q expert-query rounds with monotonic active-set growth, $S_1 \subseteq S_2 \subseteq \cdots \subseteq S_Q$. Under Assumption 2, define $\pi_Q = 1-(1-p)^Q$. For any fixed witness support $R \in \mathcal{R}_\eta$, $\Pr(R \subseteq S_Q) = \pi_Q^{|R|}$. In particular, $\Pr(R \subseteq S_Q) \geq \max\{0, 1-k(1-p)^Q\}$. The expected final active-set size is*

$$\mathbb{E}[|S_Q|] = \kappa\big(1-(1-p)^Q\big)+(N-\kappa)\big(1-(1-q_{\text{fp}})^Q\big). \quad (4)$$

Proposition 1 shows the main tradeoff in adaptive EGBO. More expert-query rounds increase the probability of covering a sparse near-optimal support, because each relevant variable has more chances to be added. However, more rounds also increase the expected number of false-positive variables. Thus adaptivity can improve the selection gap, but may make the BO problem harder by increasing $|S_Q|$.

Remark 1 (When one-shot EGBO improves over vanilla BO)**.** *The one-shot bound shows that EGBO improves over vanilla BO when the cost of using the expert-selected subspace is smaller than the benefit of optimizing in lower dimension. In particular, if S contains a witness support, then EGBO is better than vanilla BO whenever*

$$\eta + \varepsilon_{\text{BO}}(T, |S|) < \varepsilon_{\text{BO}}(T, N). \quad (5)$$

The left-hand side is the EGBO bound: a selection gap η plus a BO optimization gap in dimension $|S|$. The right-hand side is the corresponding BO optimization gap in the full ambient dimension N.

All proofs are in Appendix C.

4. Experimental Evaluation

In computational experiments preceding our real-world deployment, we evaluate EGBO at two levels. First, synthetic experiments on Hartmann6, a classic test function for optimization, in 200 dimensions (Section 4.1) evaluate the framework under controlled expert quality. Second, we replace the simulated expert with an LLM (Claude Opus 4.7) and measure performance on FORMULATEBENCH's 24 plant-based formulation tasks. We compare against vanilla BO, random search, REMBO, SEBO, TuRBO, SAASBO, and BO on k randomly chosen variables. All methods are run for $T = 20$ evaluations, chosen based on the practical constraints of the food science setting, with $n_{\text{init}} = 5$, and we report 95% confidence intervals over 10 random seeds. Gaussian process and SAASBO hyperparameters follow BoTorch defaults. Appendix D contains full experimental details.

4.1. Synthetic Experiments on Hartmann6

Methods. To Hartmann6, we append 194 dummy coordinates that do not affect the objective. The random subset + BO baseline is given the oracle sparsity level $k = 6$. We sweep two axes: (i) the number of expert query rounds Q, and (ii) the expert quality parameters.

Results. As shown in Figure 1, EGBO with a strong expert ($p = 0.9$, $q_{fp} = 0.01$) achieves the second best performance, after the oracle. However, EGBO with a weak expert ($p = 0.2$, $q_{fp} = 0.1$) is outperformed by random search at $Q = 1$, though adaptivity helps to partially close the gap. This failure mode is the empirical counterpart to the motivating concern in Section 1: when the expert's coverage probability is small, the optimizer spends its entire budget in a subspace that does not contain a good solution, and there is no mechanism to recover. Sweeping Q from 1 to $T = 20$ (Figure 9, Appendix E.4) reveals the two regimes predicted by Proposition 1. With a strong expert, regret at T is essentially flat across Q (gap to global optimum 0.94–1.19 over $Q \in [1, 20]$): one-shot querying already saturates the selection gap when recall is high, in line with the bound $\eta + \varepsilon_{BO}(T, |S|)$ of Theorem 1. With a weak expert, regret drops from 3.20 at $Q = 1$ to 2.68 at $Q = T$, with the largest improvement between $Q = 8$ and $Q = T$ as the expert is consulted at every BO step. Appendix Figure 10 plots the corresponding $|S_q|$ trajectory: under a strong expert $|S_q|$ saturates within one round at a near-optimal support, while under a weak expert $|S_q|$ grows slowly over many rounds before covering one. This matches Equation (4): low p makes round-by-round coverage of $\mathcal{U}_\eta$ unlikely, so weak experts require many rounds to overcome a poor initial subspace. The (p, q_{fp}) heatmap (Figure 8, Appendix E.3) empirically maps the selection-gap component of Theorem 1.

4.2. FORMULATEBENCH

Methods. To compare EGBO to baselines on sustainable protein formulation tasks, and evaluate whether LLMs are sufficiently strong experts in this domain, we construct FORMULATEBENCH: a public benchmark grounded in real-world consumer demand for sustainable proteins that are nutritionally equivalent or superior to the corresponding animal-based products (Ulhas et al., 2023). The task is, across 24 categories, to find a set of plant-based ingredients and concentrations that best matches the nutritional profile of the target animal product. Nutritional data is obtained from USDA FoodData Central (USDA, 2019). The ground set of ingredients is constructed as the union of ingredients in the NECTAR plant-based meat and dairy dataset (NECTAR, 2025) that have nutritional information in USDA FoodData Central. The LLM prompt is in Appendix D.3. FORMULATEBENCH deliberately uses nutritional distance as its primary objective rather than predicted sensory similarity. This is a conservative design choice: nutritional composition is fully computable from ingredient databases, requires no learned surrogate, and introduces no model-dependent

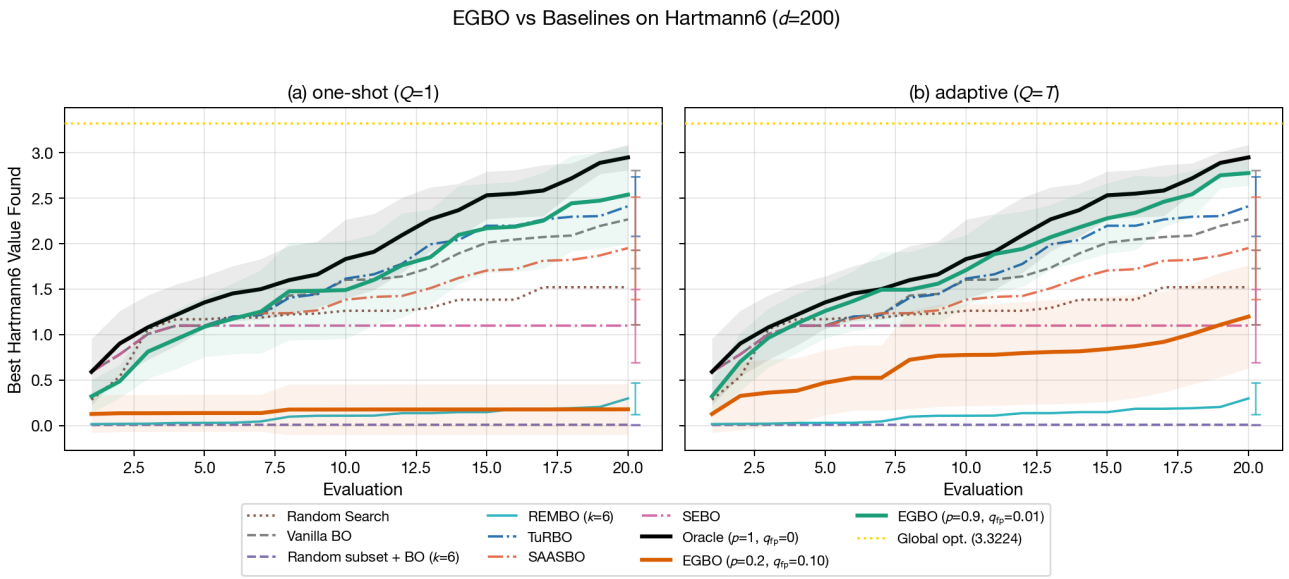

Figure 1. EGBO vs. baselines on Hartmann6 embedded in $d = 200$ dimensions, with 95% CIs. CIs for all methods are shown in Figure 7 of Appendix E. **(a)** One-shot expert query ($Q = 1$): the simulated expert proposes an active variable set once before BO begins, and BO runs in that fixed subspace for the remaining budget. **(b)** Fully adaptive ($Q = T$): the expert is queried before each evaluation, with monotonically growing active set. The strong-expert variant ($p = 0.9$, $q_{fp} = 0.01$) closely tracks the oracle in both regimes; the weak-expert variant ($p = 0.2$, $q_{fp} = 0.10$) underperforms random search under one-shot querying but improves under adaptive querying.

confound into the benchmark. A method that cannot match a target nutritional profile, a linear function of ingredient concentrations, is unlikely to succeed on the harder sensory objective.

The objective is the per-dimension z-score RMSE between a candidate formulation's nutrient profile and the target, over 111 and 91 candidate plant-based meat and dairy ingredients respectively, using $m = 8$ nutrient targets (calories, total fat, saturated fat, sodium, fiber, protein, carbs, sugar). Unlike the synthetic benchmark, the nutrition-matching objective depends on all ingredients, so sparsity is not built into the function. Instead, it is a consequence of the problem's geometry. Because the nutrition objective is an NNLS problem with m nutrient targets, there exists an optimal solution using at most m ingredients, or at most $m + 1$ under a simplex constraint (Proposition 2, Appendix C). In our setup $m = 8$, giving $k \leq 9$, independent of the ingredient database size. We additionally report two analytic floors per category: $\text{Oracle}_{\text{full}}$, the constrained NNLS optimum over all d ingredients, and $\text{Oracle}_{k=9}$, the same restricted to the LLM's chosen subset. These give an empirical decomposition of the suboptimality bound: $\text{Oracle}_{\text{full}} \to \text{Oracle}_{k=9}$ is the selection gap, and $\text{Oracle}_{k=9} \to f(\hat{x}_T)$ is the residual optimization gap.

Results. As shown in Figure 2, one-shot EGBO with Claude Opus 4.7 as the expert outperforms baselines on average across categories, and on most individual categories (Appendix F).[1] The two NNLS oracles let us decompose the residual. The optimization gap exceeds the selection gap on average, suggesting that for FORMULATEBENCH at this budget the primary bottleneck is BO subroutine convergence on the chosen subspace rather than the LLM's variable selection. Adaptive expert querying ($Q = 5$, with up to two ingredient additions per round) does not improve over one-shot ($Q = 1$) on this benchmark (Appendix F.3), motivating our choice of one-shot EGBO in Section 5. The LLM's $k = 9$ selection already spans a near-optimal NNLS support, so additional rounds expand the active set without improving achievable loss: the regime predicted by Proposition 1. We additionally evaluate one-shot EGBO under a non-NNLS sensory objective (Gemini 3.1 Pro's estimated similarity score, Appendix F.4). EGBO substantially outperforms all baselines on plant-based dairy and matches a random-subset baseline on plant-based meat, a difference we examine in Appendix F.4.

5. Real-World Deployment: Sustainable Dairy Formulation

Next, informed by the performance of LLM-guided EGBO on FORMULATEBENCH, we deploy EGBO in two real-world sustainable protein settings, in partnership with a

[1]For reproducibility, the 24 ingredient selections were sampled once from Claude Opus 4.7 (temperature 0) and cached as `LLM_SELECTIONS` in the released code.

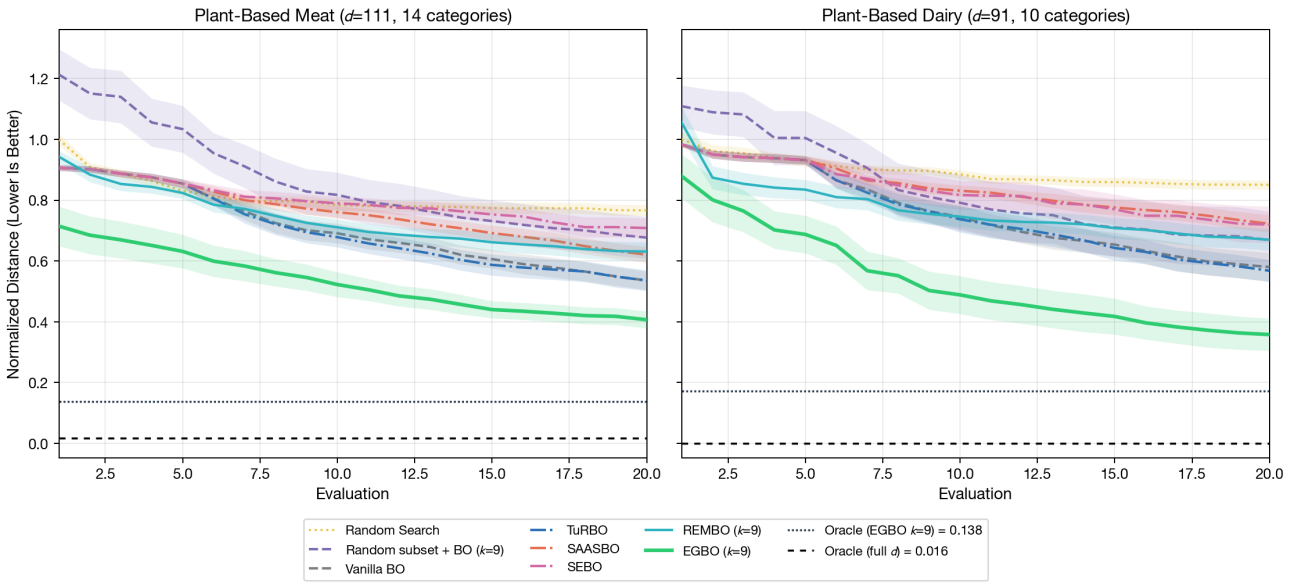

Figure 2. Nutrition matching results across the 24 plant-based meat and dairy categories of FORMULATEBENCH, with 95% CIs. One-shot EGBO (using Claude Opus 4.7) outperforms REMBO, SAASBO, TuRBO, SEBO, random search, and BO with a random subset of variables. $k = 9$ was chosen based on Proposition 2. Results per category are shown in Appendix F. The LLM prompt is in Appendix D.3.

food company. In each setting the objective is to optimize a plant-based product toward a target sensory profile under a limited budget. We study two product categories that pose distinct formulation challenges: yogurt and cottage cheese. Dairy products are particularly high-impact targets for plant-based reformulation due to the impact of the dairy industry on deforestation, climate change, land use, and animal welfare (Pendrill et al., 2022; Gerber et al., 2013; IPBES, 2019; von Keyserlingk et al., 2013). For yogurt, we conduct a controlled comparison against a professional human food scientist. The food company has developed a commercially available, LLM-based agentic framework, which we denote Food Scientist Agent (FSA) and use as the expert in EGBO. To ensure reproducibility, we release FSA's output, the optimization variables, and all code. We set $Q = 1$, as both FSA's strong initial recipes (utility 0.767, 0.776) and performance of LLM-guided EGBO on FORMULATEBENCH suggest one-shot selection is reasonable for this setting; we leave real-world evaluation of adaptive expansion to future work.

5.1. Methods.

The full candidate ingredient pool is the food company's database ($N \approx 88{,}000$). Prior to optimization, FSA performed one-shot variable selection: given the product category, the target dairy sensory profile, and the plant-based constraint, it selected a small set of active ingredients from the full pool. For yogurt, FSA selected $d = 7$ active ingredients. The sensory objective was defined over four attributes assessed on a 0 to 15 scale: Consistency (target $t_1 = 12$), Creaminess (target $t_2 = 5$), Tanginess (target $t_3 = 5$), and Uniformity (target $t_4 = 13$). The initial recipe had sensory scores $s^{(0)} = (5, 3, 10, 13)$, creating a nontrivial multi-objective problem in which one attribute already matches the target exactly while three require coordinated improvement. For cottage cheese, FSA selected $d = 9$ active ingredients. The sensory objective was defined over six attributes assessed on a 0 to 15 scale: Homogeneity (target $t_1 = 6$), Dispersion Phase (target $t_2 = 6$), Slipperiness (target $t_3 = 6$), Sour (target $t_4 = 6.5$), Dairy (target $t_5 = 7$), and Off-Notes (target $t_6 = 0$). The initial recipe had sensory scores $s^{(0)} = (4, 8, 9, 10, 3, 7)$, presenting a more complex optimization landscape: five of six attributes require substantial correction, with Off-Notes requiring near-complete elimination and Dairy requiring more than a doubling. In both cases, the total laboratory budget was $T = 10$ iterations of 3 recipes each (30 total recipes).

Scalar utility. In both campaigns, we aggregate the multi-objective sensory targets into a scalar utility score. For a candidate formulation x, let $s_j(x)$ denote the observed panel score for attribute j, t_j its target, m the number of sensory attributes, and $S_{\max}$ the maximum value of the sensory scale. We define the normalized closeness-to-target score

$$c_j(x) = 1 - \frac{|s_j(x) - t_j|}{S_{\max}}, \qquad j = 1, \ldots, m, \tag{6}$$

which lies in $[0, 1]$, with 1 corresponding to exact target matching. With uniform weights $w_j = 1/m$, the scalar

utility is

$$f(x) = \frac{1}{m} \sum_{j=1}^{m} \left(1 - \frac{|s_j(x) - t_j|}{S_{\max}}\right). \quad (7)$$

Each laboratory-evaluated formulation yields an observation of $f(x)$, and EGBO seeks a formulation $x^\star$ with high utility under the budget. At each iteration, EGBO proposed a batch of recipes within the currently active subspace; the proposed formulations were manufactured according to the respective standardized preparation protocols and scored by human sensory evaluation on the target attributes.

Real-world performance is evaluated along two axes. First, *final product quality*, measured by the best observed utility $f(\hat{x}_T)$ and by the final per-attribute distances to target. Second, *optimization efficiency*, measured by the progression of utility across iterations and by the number of iterations required to achieve a substantial reduction in target mismatch relative to the baseline.

5.2. Results

EGBO developed a plant-based yogurt formulation with a baseline utility of $f(x^{(0)}) = 0.767$ (recipe produced from FSA) then improved it to a best observed utility of $f(\hat{x}_T) = 0.992$ over 10 laboratory iterations (30 total recipes), a 29% improvement. For cottage cheese, EGBO developed a plant-based formulation from a baseline utility of $f(x^{(0)}) = 0.776$ then improved it to a best observed utility of $f(\hat{x}_T) = 0.975$ over 10 laboratory iterations (30 total recipes), a 26% improvement. Tables 1 and 7 summarize the per-attribute results for yogurt and cottage cheese respectively.

5.3. EGBO vs. Human Food Scientist

We compare EGBO with a professional human food scientist (HFS) on the yogurt task. HFS had prior experience in recipe formulation but no prior experience specifically with plant-based dairy, to avoid the situation of HFS simply reproducing a suitable recipe from memory. This mirrors the setting of a company entering a new product category rather than reproducing a known recipe. We report two complementary comparisons. First, we compare workflows under the same approximate laboratory time budget (40 hours): HFS develops a plant-based yogurt from scratch, while EGBO starts from a formulation selected by FSA and uses BO to refine it. Second, to isolate the refinement step, we compare BO and HFS optimization from the same agent-generated starting formulation. In the workflow comparison, HFS reached utility 0.850, while EGBO reached 0.992. In the same-start refinement comparison, HFS improved the agent-generated recipe from 0.767 to 0.867, whereas BO improved it from 0.767 to 0.992.

Attribute	Target	Baseline	Best	Error	
				Before	After
Consistency	12	5	12	7	0
Creaminess	5	3	5	2	0
Tanginess	5	10	5	5	0
Uniformity	13	13	12.5	0	0.5
Utility $f(x)$	1.000	0.767	0.992	0.233	0.008

Table 1. Overall utility of the plant-based yogurt, as assessed by the median scores of a trained panel of five participants, improved 29% relative to baseline.

Figure 3. Best plant-based yogurt formulation (iteration 7, recipe 2).

Experiment 1: full workflow comparison. HFS was asked to develop, from scratch, a plant-based yogurt recipe targeting the same four sensory objectives used in the EGBO campaign (Consistency 12, Creaminess 5, Tanginess 5, Uniformity 13). Despite having full freedom to perform batching, HFS chose to follow a conventional iterative workflow: prepare a candidate formulation, allow it to cool, taste, decide on the next modification, and repeat. A total of 29 trials were executed over approximately 40 hours. Of these, 19 were classified as failed trials (the operator abandoned the attempt and restarted from scratch), and 10 produced formulations deemed sufficiently promising to evaluate for utility. Figure 32 shows the utility progression across the 10 evaluated trials. The best utility achieved was 0.850 (trial 28), with a general upward trend as HFS progressively refined the plant base and texture.

Experiment 2: same-start refinement comparison. To isolate the refinement step, HFS was given FSA's recipe ($f(x^{(0)}) = 0.767$) and asked to optimize it toward the same four sensory targets. HFS was allocated 10 iterations, matching the EGBO campaign. Similarly, HFS chose to produce one recipe per iteration, and EGBO proposed batches of three recipes per iteration. Each HFS iteration required approximately 4 hours of laboratory time, for a total of approximately 40 hours.

Figure 33 shows the utility for both EGBO and HFS starting from the same baseline. EGBO reached a utility of 0.950

Comparison	Method	Start	Best	Relative Gain	Lab time
Overall workflow	HFS from scratch	–	0.850	–	∼40 h
Overall workflow	EGBO: FSA + BO	0.767	0.992	+17% vs. HFS	∼40 h
Same-start refinement	HFS refinement	0.767	0.867	–	∼40 h
Same-start refinement	BO refinement	0.767	0.992	+14% vs. HFS	∼40 h

Table 2. Human comparison on plant-based yogurt. We first compare the overall workflows under the same approximate laboratory time budget; the same-start comparison isolates the refinement step. EGBO (FSA variable selection + BO refinement) was run once and compared against a professional human food scientist (HFS) under two protocols, each with the same ∼40 h laboratory time budget. In the overall workflow comparison, HFS develops a recipe from scratch; EGBO uses its full pipeline (FSA-generated seed at $f(x^{(0)}) = 0.767$, then 10 iterations of BO refinement). In the same-start refinement comparison, HFS and BO both start from the FSA's initial recipe and refine for 10 iterations, isolating the refinement step. The single EGBO run yields the same 0.992 result in both rows; only the human comparator differs.

by iteration 4 and peaked at 0.992 by iteration 7. HFS improved more slowly, reaching 0.825 at iteration 3 and 0.867 at iteration 10. The gap widened monotonically from iteration 3 onward: by iteration 10, EGBO's best-so-far (0.992) exceeded the HFS' best-so-far (0.867) by 14.4%.

Time and utility comparison. Table 2 summarizes the comparison. The results highlight two distinct advantages of the EGBO workflow. First, *parallelism*: EGBO proposes a batch of 3 recipes per iteration, all of which can be prepared and evaluated in a single laboratory session, whereas HFS' sequential reasoning requires one-at-a-time execution. Second, *consistency*: HFS' trajectory shows regressions (iterations 4–5 and 8 in Experiment 2), reflecting the difficulty of mentally tracking coupled ingredient interactions across multiple sensory dimensions, whereas EGBO's GP surrogate maintains a global model that reduces such backsliding. We emphasize two caveats. First, this comparison involves a single human; different food scientists with different backgrounds may perform better or worse. Second, HFS had no prior experience with plant-based dairy specifically, whereas a specialist in this sub-domain might close the gap.

6. Discussion

The current pace of development in sustainable proteins is incompatible with the urgency needed for climate change mitigation (Zurek et al., 2022). Self-driving laboratories, which can run experiments continuously and at scale, offer a path to the throughput this challenge requires (Abolhasani & Kumacheva, 2023). EGBO advances the software layer of such a system, outperforming a professional food scientist on plant-based yogurt formulation. Our deployment surfaces open challenges for methodological development. Progress on computational simulation of taste and texture, perhaps building on recent progress in computational olfaction (Lee et al., 2023), could transform the throughput bottleneck, partially replacing trained human panels with learned surrogates. Multi-fidelity BO with weak low-fidelity models is thus a methodological area that could advance sustainable protein formulation (Mikkola et al., 2023; Sabanza-Gil et al., 2025). Combined with robotic preparation, this could close the remaining gaps toward fully autonomous formulation.

Limitations. Assumption 2 treats variable selection as independent across coordinates: a practitioner who identifies one protein source is likely to also identify related emulsifiers. The FORMULATEBENCH objective uses nutritional distance as a proxy for product quality, which may not capture sensory similarity. The human comparison involves a single scientist without plant-based dairy experience; further comparisons are needed. Finally, the expert quality parameters (p, q) are defined relative to the unknown support union $\mathcal{U}_\eta$ and cannot be estimated without knowledge of $\mathcal{U}_\eta$.

Societal impact. EGBO could be used to optimize formulations that prioritize cost or palatability over nutritional quality. To mitigate this, we have focused our benchmark and deployment exclusively on applications with positive environmental and public health motivations. Additionally, EGBO could reduce demand for human food scientists. To mitigate this, in our framework, humans remain responsible for objective specification and quality oversight, automating iterative refinement rather than replacing scientific judgment.

Impact Statement

This paper presents work whose goal is to advance the fields of machine learning and sustainable proteins. Sustainable proteins are critical for food security (Onwezen et al., 2024; Nirmal et al., 2025), and mitigation of pandemic risk (Hayek, 2022; Bartlett et al., 2022; Espinosa et al., 2020) and climate change (Poore & Nemecek, 2018; Rubio et al., 2020). More efficient discovery of sustainable proteins will also have substantial benefits for animal welfare (Food and Agriculture Organization of the United Nations, 2023).

EGBO is designed to accelerate the development of sustainable alternatives to animal-derived products, where each

experimental iteration is expensive and environmentally costly. More sample-efficient formulation campaigns could reduce the time and resources required to bring competitive plant-based products to market, contributing to reductions in the environmental footprint of food production. The framework extends beyond food science to any expensive black-box optimization problem where approximate sparsity is a reasonable assumption and domain expertise is available - including drug discovery, catalyst design, and materials composition. We flag one novel risk introduced by LLM-based experts: hallucinated or miscalibrated variable and range proposals can silently degrade performance, particularly in the high-dimensional regime we target, where the sparse support is unknown and cannot be directly verified. On structured benchmarks where sparse optimal supports can be characterized, (p, q) can be estimated and used to compare candidate experts; extending this to calibration for novel deployments is an open problem that practitioners should bear in mind before trusting LLM selections on unfamiliar tasks.

References

Abolhasani, M. and Kumacheva, E. The rise of self-driving labs in chemical and materials sciences. *Nature Synthesis*, 2(6):483–492, 2023.

Anonymous. TasteBench: benchmark for sensory prediction, from molecules to sustainable foods. *In submission*, 2026.

Bartlett, H., Holmes, M. A., Petrovan, S. O., Williams, D. R., Wood, J. L. N., and Balmford, A. Understanding the relative risks of zoonosis emergence under contrasting approaches to meeting livestock product demand. *Royal Society Open Science*, 9(6):211573, 2022. doi: 10.1098/rsos.211573.

Becker, D., Schmitt, C., Bovetto, L., Rauh, C., McHardy, C., and Hartmann, C. Optimization of complex food formulations using robotics and active learning. *Innovative Food Science and Emerging Technologies*, 83:103232, 2023.

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Bull, A. D. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.

Cao, L., Russo, D., Felton, K., Salley, D., Sharma, A., Keenan, G., Mauer, W., Gao, H., Cronin, L., and Lapkin, A. A. Optimization of formulations using robotic experiments driven by machine learning DoE. *Cell Reports Physical Science*, 2(1):100295, 2021.

Eriksson, D. and Jankowiak, M. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Uncertainty in Artificial Intelligence*, pp. 493–503, 2021.

Eriksson, D., Pearce, M., Gardner, J., Turner, R. D., and Poloczek, M. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, 2019.

Espinosa, R., Tago, D., and Treich, N. Infectious diseases and meat production. *Environmental and Resource Economics*, 76:1019–1044, 2020. doi: 10.1007/s10640-020-00484-3.

Food and Agriculture Organization of the United Nations. FAOSTAT: Crops and livestock products. https://www.fao.org/faostat/en/#data/QCL, 2023.

Frazier, P. I. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Gerber, P. J., Steinfeld, H., Henderson, B., Mottet, A., Opio, C., Dijkman, J., Falcucci, A., and Tempio, G. Tackling climate change through livestock: A global assessment of emissions and mitigation opportunities. Technical report, Food and Agriculture Organization of the United Nations (FAO), Rome, 2013.

Ghafarollahi, A. and Buehler, M. J. SciAgents: Automating scientific discovery through multi-agent intelligent graph reasoning. *arXiv preprint arXiv:2409.05556*, 2024.

Hayek, M. N. The infectious disease trap of animal agriculture. *Science Advances*, 8(44):eadd6681, 2022. doi: 10.1126/sciadv.add6681.

IPBES. Global assessment report on biodiversity and ecosystem services. Technical report, Intergovernmental Science-Policy Platform on Biodiversity and Ecosystem Services, Bonn, Germany, 2019.

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. ZINC: A free tool to discover chemistry for biology. *Journal of Chemical Information and Modeling*, 52(7):1757–1768, 2012. doi: 10.1021/ci3001277.

Lao, J., Wang, Y., Li, Y., Wang, J., Zhang, Y., Cheng, Z., Chen, W., Tang, M., and Wang, J. GPTuner: A manual-reading database tuning system via GPT-guided Bayesian optimization. *Proceedings of the VLDB Endowment*, 17 (8):1939–1952, 2024.

Lee, B. K., Mayhew, E. J., Sanchez-Lengeling, B., Wei, J. N., Qian, W. W., Little, K. A., Andres, M., Nguyen, B. B., Moloy, T., Yasonik, J., et al. A principal odor map unifies diverse tasks in olfactory perception. *Science*, 381 (6661):999–1006, 2023.

Liu, S., Feng, Q., Eriksson, D., Letham, B., and Bakshy, E. Sparse Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 3754–3774. PMLR, 2023.

Liu, T., Astorga, N., Seedat, N., and van der Schaar, M. Large language models to enhance Bayesian optimization. In *International Conference on Learning Representations*, 2024.

Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., and Ha, D. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

M. Bran, A., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6 (5):525–535, 2024. doi: 10.1038/s42256-024-00832-8.

Mikkola, P., Martinelli, J., Filstroff, L., and Kaski, S. Multi-fidelity Bayesian optimization with unreliable information sources. In Ruiz, F., Dy, J., and van de Meent, J.-W. (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7425–7454. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/mikkola23a.html.

NECTAR. Taste of the industry 2025. https://www.nectar.org/sensory-research/2025-taste-of-the-industry, 2025.

Nirmal, N. P. et al. Alternative protein sources: Addressing global food security and environmental sustainability. *Sustainable Development*, 2025. doi: 10.1002/sd.3338.

Onwezen, M. C. et al. Current challenges of alternative proteins as future foods. *npj Science of Food*, 8, 2024. doi: 10.1038/s41538-024-00291-w.

Pendrill, F., Gardner, T. A., Meyfroidt, P., Persson, U. M., Adams, J., Azevedo, T., Bastos Lima, M. G., Baumann, M., Curtis, P. G., De Sy, V., Garrett, R., Godar, J., Goldman, E. D., Hansen, M. C., Heilmayr, R., Herold, M., Kuemmerle, T., Lathuillière, M. J., Ribeiro, V., Tyukavina, A., Weisse, M. J., and West, C. Disentangling the numbers behind agriculture-driven tropical deforestation. *Science*, 377(6611):eabm9267, 2022.

Poore, J. and Nemecek, T. Reducing food's environmental impacts through producers and consumers. *Science*, 360 (6392):987–992, 2018.

Raccuglia, P., Elbert, K. C., Adler, P. D. F., Falk, C., Wenny, M. B., Mollo, A., Zeller, M., Friedler, S. A., Schrift, J., and Norquist, A. J. Machine-learning-assisted materials discovery using failed experiments. *Nature*, 533(7601): 73–76, 2016. doi: 10.1038/nature17439.

Roohani, Y. H., Lee, A. H., Huang, Q., Vora, J., Steinhart, Z., Huang, K., Marson, A., Liang, P., and Leskovec, J. Biodiscoveryagent: An AI agent for designing genetic perturbation experiments. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HAwZGLcye3.

Rubio, N. R., Xiang, N., and Kaplan, D. L. Plant-based and cell-based approaches to meat production. *Nature Communications*, 11:6276, 2020. doi: 10.1038/s41467-020-20061-y.

Sabanza-Gil, V., Barbano, R., Pacheco Gutiérrez, D., Luterbacher, J. S., Hernández-Lobato, J. M., Schwaller, P., and Roch, L. Best practices for multi-fidelity Bayesian optimization in materials and molecular research. *Nature Computational Science*, 5(7):572–581, 2025.

Snoek, J., Larochelle, H., and Adams, R. P. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2012.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. W. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. In *IEEE Transactions on Information Theory*, volume 58, pp. 3250–3265, 2012.

Thomas, A., Yee, A., Mayne, A., Mathur, M. B., Jurafsky, D., and Gligorić, K. What can large language models do for sustainable food? In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=f6SFHNfuMu.

Twine, R. Emissions from animal agriculture—16.5% is the new minimum figure. *Sustainability*, 13(11):6276, 2021.

Ulhas, R. S., Ravindran, R., Malaviya, A., Priyadarshini, A., Tiwari, B. K., and Rajauria, G. A review of alternative proteins for vegan diets: Sources, physico-chemical properties, nutritional equivalency, and consumer acceptance. *Food research international*, 173:113479, 2023.

USDA. Fooddata central, 2019. URL https://fdc.nal.usda.gov/. Accessed: 2025-10-04.

van den Bedem, S. D., Kuhl, E., and Cotto, C. Open-source benchmarking of plant-based and animal meats. *arXiv preprint arXiv:2603.03370*, 2026.

von Keyserlingk, M. A. G., Martin, N. P., Kebreab, E., Knowlton, K. F., Grant, R. J., Stephenson, M., Sniffen, C. J., Harner, J. P., Wright, A. D., and Smith, S. I. Invited review: Sustainability of the US dairy industry. *Journal of Dairy Science*, 96(9):5405–5425, 2013.

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., and de Freitas, N. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., and Chen, X. Large language models as optimizers. In *International Conference on Learning Representations (ICLR)*, 2024.

Zurek, M., Hebinck, A., and Selomane, O. Climate change and the urgency to transform food systems. *Science*, 376 (6600):1416–1421, 2022.

A. Symbols and Abbreviations

Table 3 contains a list of symbols and abbreviations introduced throughout this paper.

Symbol	Used For
Acronyms	
EGBO	Expert-Guided Bayesian Optimization
BO	Bayesian optimization
GP	Gaussian process
NNLS	nonnegative least squares
LLM	large language model
Problem parameters	
N	ambient input dimension
$\mathcal{X}$	input domain, $\mathcal{X} \subseteq \mathbb{R}^N$
f	black-box objective, $f : \mathcal{X} \to \mathbb{R}$
x^*	global maximizer of f on $\mathcal{X}$
$\|x\|_0$	number of nonzero (active) coordinates of x
$u_i^{\max}$	upper bound of the i-th coordinate
Sparsity structure	
k	target sparsity level (max size of any sparse near-optimal support)
η	sparsity gap: $f(\tilde{x}) \geq f(x^*) - \eta$ for some k-sparse $\tilde{x}$
$\tilde{x}^{(R)}$	witness for support R: a k-sparse near-optimum supported in R
R	a sparse near-optimal support, $R \subseteq [N]$, $\|R\| \leq k$
$\mathcal{R}_\eta$	family of all k-sparse η-near-optimal supports
M	multiplicity, $M = \|\mathcal{R}_\eta\|$
M'	number of pairwise disjoint supports in $\mathcal{R}_\eta$ used in bounds
U_η	support union, $\bigcup_{R \in \mathcal{R}_\eta} R$
κ	size of the support union, $\kappa = \|U_\eta\|$
Expert quality	
E	domain expert (human, LLM, or other knowledge source)
p	per-variable expert recall on U_η
q_{fp}	per-variable expert false-positive rate outside U_η
Algorithm and active set	
T	total BO budget (function evaluations)
Q	number of expert query rounds
τ	BO iterations per round, $\tau = \lfloor T/Q \rfloor$
S, S_q	active variable set (at round q)
d, d_Q	active set size, $d = \|S\|$ (resp. $d_Q = \|S_Q\|$)
$\mathcal{X}_S$	restricted domain $\{x \in \mathcal{X} : x_i = 0 \text{ for } i \notin S\}$
Convergence quantities	
$\hat{x}_T$	best point observed by EGBO after T evaluations
f_S^*	best achievable value in $\mathcal{X}_S$, $\max_{x \in \mathcal{X}_S} f(x)$
Δ_S	selection gap, $f(x^*) - f_S^*$
r_T	optimization gap, $f_S^* - f(\hat{x}_T)$
$\varepsilon_{\mathrm{BO}}(T, d)$	BO subroutine convergence rate as a function of budget T and dimension d
δ_{BO}	failure probability of the BO subroutine
NNLS objective (Section 4.2)	
V	ingredient-to-nutrient matrix, $V \in \mathbb{R}_{\geq 0}^{m \times N}$
v^*	target nutrient profile, $v^* \in \mathbb{R}^m$
m	number of nutrient targets

Table 3. Symbols and abbreviations used in this paper.

B. Extended Related Work

ML-driven formulation optimization. Within food and formulated-product development, Cao et al. (2021) coupled Thompson Sampling Efficient Multi-objective Optimization (TSEMO) with a naive Bayes stability classifier to optimize a five-ingredient liquid formulation, finding nine acceptable recipes in 15 working days, and Becker et al. (2023) applied

TSEMO to a three-ingredient whey protein formulation using a fully automated milli-fluidic platform. Neither approach exploits expert priors to reduce the search dimension. Thomas et al. (2025) found that LLMs can exceed the performance of human food scientists in experimental design for plant-based meats, as judged by other expert food scientists.

C. Theory and Proofs

Throughout this appendix, we use the independence between the expert's variable-selection randomness and the BO subroutine's randomness assumed in Section 3.

C.1. Suboptimality Decomposition

Proof of Equation (2). By definition,

$$f_S^\star = \max_{x \in \mathcal{X}_S} f(x).$$

Adding and subtracting $f_S^\star$ gives

$$f(x^*) - f(\hat{x}_T) = \big(f(x^*) - f_S^\star\big) + \big(f_S^\star - f(\hat{x}_T)\big).$$

The first term is the selection gap and the second term is the optimization gap.

Now suppose $R \subseteq S$ for some $R \in \mathcal{R}_\eta$. By Assumption 1, there exists a witness $\tilde{x}^{(R)}$ with

$$\mathrm{supp}(\tilde{x}^{(R)}) \subseteq R \qquad \text{and} \qquad f(\tilde{x}^{(R)}) \geq f(x^*) - \eta.$$

Since $R \subseteq S$, we have $\tilde{x}^{(R)} \in \mathcal{X}_S$. Therefore,

$$f_S^\star = \max_{x \in \mathcal{X}_S} f(x) \geq f(\tilde{x}^{(R)}) \geq f(x^*) - \eta.$$

Thus $f(x^*) - f_S^\star \leq \eta$. □

C.2. One-Shot Coverage (Theorem 1)

Proof. Fix a witness support $R \in \mathcal{R}_\eta$. In the one-shot setting, each variable $i \in R$ is included in the active set S independently with probability p. Hence $\Pr(R \subseteq S) = p^{|R|}$. Since $|R| \leq k$ and $p \in (0, 1]$, $p^{|R|} \geq p^k$.

On the event $R \subseteq S$, the subspace $\mathcal{X}_S$ contains the witness $\tilde{x}^{(R)}$. By the decomposition above, the selection gap is therefore at most η: $f(x^*) - f_S^\star \leq \eta$.

By Assumption 3, when BO is run on $\mathcal{X}_S$ for T evaluations, it returns $\hat{x}_T$ satisfying $f_S^\star - f(\hat{x}_T) \leq \varepsilon_{\mathrm{BO}}(T, |S|)$ with probability at least $1 - \delta_{\mathrm{BO}}$.

Combining the selection and optimization gaps gives

$$f(x^*) - f(\hat{x}_T) \leq \eta + \varepsilon_{\mathrm{BO}}(T, |S|).$$

The coverage event $R \subseteq S$ and the BO success event are independent by assumption, so the joint event occurs with probability at least $p^{|R|}(1 - \delta_{\mathrm{BO}}) \geq p^k(1 - \delta_{\mathrm{BO}})$.

It remains to compute the expected active-set size. Under Assumption 2,

$$\mathbb{E}[|S|] = \kappa p + (N - \kappa) q_{\mathrm{fp}}.$$

□

C.3. Coverage of Multiple Disjoint Supports

Theorem 1 fixes one sparse near-optimal support R. If there are many such supports, the probability that the expert covers at least one of them can be larger.

Lemma 1 (Coverage of multiple disjoint supports)**.** *Let $\mathcal{R}' \subseteq \mathcal{R}_\eta$ be a collection of pairwise disjoint supports. Suppose each variable in U_η is included independently with probability π. Then*

$$\Pr(\exists R \in \mathcal{R}' : R \subseteq S) = 1 - \prod_{R \in \mathcal{R}'} \left(1 - \pi^{|R|}\right).$$

If additionally $|R| \leq k$ for every $R \in \mathcal{R}'$, then

$$\Pr(\exists R \in \mathcal{R}' : R \subseteq S) \geq 1 - (1 - \pi^k)^{|\mathcal{R}'|}.$$

Proof. For a fixed support R, $\Pr(R \subseteq S) = \pi^{|R|}$. Because the supports in $\mathcal{R}'$ are pairwise disjoint, the events $\{R \subseteq S\}$ depend on disjoint sets of independent Bernoulli variables, so they are independent. Hence

$$\Pr(\text{no support in } \mathcal{R}' \text{ is covered}) = \prod_{R \in \mathcal{R}'} \left(1 - \pi^{|R|}\right).$$

If $|R| \leq k$ and $\pi \in [0, 1]$, $\pi^{|R|} \geq \pi^k$, so $1 - \pi^{|R|} \leq 1 - \pi^k$ and the bound follows. □

In the one-shot setting, $\pi = p$. Therefore, if $\mathcal{R}_\eta$ contains M' pairwise disjoint supports of size at most k, then

$$\Pr(\exists R \in \mathcal{R}_\eta : R \subseteq S) \geq 1 - (1 - p^k)^{M'}.$$

C.4. Adaptive Coverage (Proposition 1)

Proof. Consider any variable $i \in U_\eta$. In each expert-query round, if i has not already been included, it is added with probability p. The probability that i is not added in any of the Q rounds is $(1 - p)^Q$, so $\pi_Q = 1 - (1 - p)^Q$.

Fix $R \in \mathcal{R}_\eta$. By Assumption 2, inclusion events are independent across coordinates, so $\Pr(R \subseteq S_Q) = \pi_Q^{|R|}$.

A simple union-bound gives $\Pr(R \not\subseteq S_Q) \leq |R|(1 - p)^Q \leq k(1 - p)^Q$, hence $\Pr(R \subseteq S_Q) \geq \max\{0,\, 1 - k(1 - p)^Q\}$.

Finally, by linearity of expectation,

$$\mathbb{E}[|S_Q|] = \kappa\big(1 - (1 - p)^Q\big) + (N - \kappa)\big(1 - (1 - q_{\text{fp}})^Q\big).$$

□

D. Shared Experimental Details

This appendix documents resources, GP/acquisition configuration, and LLM prompts shared by the synthetic Hartmann6 study (Appendix E), FORMULATEBENCH (Appendix F), and the real-world dairy deployment (Appendix G). Library versions are pinned in each release directory's `requirements.txt`: `torch` $\geq$ 2.0, `botorch` $\geq$ 0.10, `gpytorch` $\geq$ 1.11, `numpy` $\geq$ 1.24, plus `pandas` $\geq$ 2.0, `scipy` $\geq$ 1.10, `matplotlib` $\geq$ 3.7 where used.

D.1. Compute resources

All experiments in this paper run on CPU; no GPU is required at any point. We executed runs on a 14-inch MacBook Pro with an Apple M5 Pro chip (18-core CPU, 20-core GPU, 16-core Neural Engine), 48 GB unified memory, and OS Tahoe 26.4.1 with the library versions pinned above. Per-experiment compute requirements are summarized in Table 4 and detailed below.

Hartmann6 ($N = 200$). The full sweep covers 9 methods $\times$ 10 seeds $\times$ 20 evaluations per run = 1,800 function evaluations per heatmap cell. Per-seed wall time is dominated by NUTS sampling in the SAAS-based methods (SAASBO, SEBO) at roughly 5–10 minutes; all other methods complete in under one minute per seed. Peak resident memory stays below 4 GB. The full Hartmann6 sweep (one-shot + adaptive, all expert regimes) is approximately 50 CPU-hours on the machine above.

Table 4. Compute resources per experimental setting. Lab time refers to physical-laboratory wall-clock time (recipe preparation, cooling, sensory evaluation), which dominates the deployment budget.

Setting	Hardware	Per-run wall time	Total CPU time	Lab time
Hartmann6 ($N = 200$)	1 CPU core	5–10 min (SAAS); <1 min (others)	∼50 CPU-h	—
FORMULATEBENCH	1 CPU core	2–5 min (SAAS); <1 min (others)	$\sim 10^2$ CPU-h	—
Yogurt deployment	1 CPU core	<1 min per acquisition step	∼10 CPU-min	∼40 h
Cottage cheese deployment	1 CPU core	<1 min per acquisition step	∼10 CPU-min	∼40 h

FORMULATEBENCH. The full sweep is 24 categories × 8 methods × 10 seeds × 20 evaluations = 38,400 objective evaluations. Per-(seed, category, method) wall time is dominated by the SAAS-based methods at 2–5 minutes; non-SAAS methods complete in well under a minute. Peak resident memory remains under 4 GB throughout. The full FORMULATEBENCH sweep is approximately 10^2 CPU-hours.

Real-world deployment. The computational footprint of each EGBO iteration is negligible: acquisition-function optimization with `num_restarts` = 20, `raw_samples` = 1024, `sample_shape` = [512], `sequential` = True completes in under one minute on a single CPU core, and per-campaign cumulative acquisition compute is on the order of 10 minutes. The binding constraint is laboratory wall time: each batch of $q = 3$ recipes requires approximately 4 hours of physical preparation, cooling, and trained-panel sensory evaluation, giving a total of ~ 40 hours of lab time per 10-iteration campaign. Both the yogurt and cottage cheese campaigns ran on the same compute and panel infrastructure.

Total compute footprint. Summing across all reported experiments, the full reproduction of every numeric result and figure in this paper requires approximately $\mathbf{1.5 \times 10^2}$ CPU-hours of single-machine compute, plus $\sim$ **80** hours of physical laboratory time across the two product campaigns and the HFS comparison.

D.2. Shared GP and acquisition configuration

Across all three settings, the BO subroutine uses BoTorch's `SingleTaskGP` with the default Matérn-5/2 ARD kernel, a zero prior mean, and `Standardize`($m = 1$) as the outcome transform. Hyperparameters (lengthscales, signal variance, noise) are fit by maximum marginal likelihood via `fit_gpytorch_mll` on `ExactMarginalLogLikelihood`; we use BoTorch defaults for priors. The acquisition function is qLogNoisyExpectedImprovement (qLogNEI) sampled with a `SobolQMCNormalSampler`; we use `sample_shape` = [256] for the synthetic and FORMULATEBENCH experiments and [512] in the deployment, where larger batches ($q = 3$) and expensive evaluations justify a sharper acquisition estimate. We optimize the acquisition function with `optimize_acqf` (L-BFGS-B from multiple restarts). Inputs are normalized to $[0, 1]^d$ before being passed to the GP. The fully Bayesian baselines (SAASBO, SEBO) instead use `SaasFullyBayesianSingleTaskGP` with NUTS (warmup steps = 512, post-warmup samples = 256); on NUTS-fit failure we fall back to `SingleTaskGP`. The NUTS budget is slightly larger than the warmup = 256, samples = 256 recommended by Eriksson & Jankowiak (2021), chosen to be conservative under the high ambient dimensions in our setting.

D.3. LLM expert: prompts and caching

FORMULATEBENCH uses three distinct LLM calls: a nutrition-aware ingredient-selection expert, a sensory-aware ingredient-selection expert, and a sensory simulator that acts as the objective for FORMULATEBENCH-SENSORY. All calls are issued through OpenRouter; the model identifiers, decoding settings, and prompts are reproduced below. To make the released runs reproducible without API access, the nutrition-expert selections are sampled once and cached as the `LLM_SELECTIONS` dict in `run_formulatebench.py`; the sensory expert and simulator calls are cached on disk per (model, prompt) by `openrouter_cache/` and `sensory_cache.sqlite`, both included in the release alongside the per-seed result JSONs. The Food Scientist Agent (FSA) used in the real-world deployment (Section 5) is a commercially-available agentic system; we use its produced recipes as one-shot inputs and do not invoke it during BO.

Nutrition expert (FORMULATEBENCH-NUTRITION). Active-set selection on the nutrition objective uses Claude Opus 4.7 (`anthropic/claude-opus-4.7` via OpenRouter), temperature 0, with system message "You are an expert plant-based food scientist.". The active-set cap $k = 9$ follows from the NNLS sparsity bound (Proposition 2, Appendix F). The user prompt is reproduced in Figure 4; the response is parsed as a JSON object {`"ingredients": [...]`}. Selections are sampled once across the 24 categories and cached.

User prompt

You have available to you the following set of ingredients: <*ingredients and nutritional properties*>.
Your goal is to match the nutritional properties of animal-based <*category*>: <*nutritional properties*>.
Choose a subset of at most 9 ingredients, on which we will run Bayesian optimization to iteratively determine ingredient concentrations.
Respond with a JSON object: {`"ingredients": ["name1", "name2", ...]`}.

Figure 4. Prompt for the nutrition-aware expert in FORMULATEBENCH-NUTRITION; the system message is "You are an expert plant-based food scientist." The cap $k = 9$ is derived from Proposition 2.

Sensory expert (FORMULATEBENCH-SENSORY). Active-set selection on the sensory objective uses Claude Opus 4.6[2] (`anthropic/claude-opus-4.6` via OpenRouter), temperature 0.2[3], `max_tokens`=2500, with system message "You are an expert plant-based food scientist. Return strict JSON only.". The user message is a JSON payload whose natural-language `prompt` field is reproduced in Figure 5; the payload also carries the per-category nutrition target, a per-nutrient constraint policy (protein, fiber, potassium, calcium $\geq$ target; saturated fat, sodium, sugars $\leq$ target; energy within $\pm 15\%$; fat and carbs within $\pm 20\%$), and a required output schema with `selected_indices`, `selected_ingredients`, `reasoning`, and `nutrition_strategy` fields.

Expert prompt

You are an expert plant-based food scientist. Your goal is to replicate the sensory experience of <*category*> with plant-based ingredients, while matching or improving on nutrition.
Nutritional properties of the animal-based <*category*>: <*nutrition vector*>.
Here is the set of possible ingredients, with nutritional properties: <*ground set table*>.
Select an initial set of ingredients; Bayesian optimization will be used to determine their concentrations.

Figure 5. Prompt for the Claude Opus 4.6 ingredient-selection expert in FORMULATEBENCH-SENSORY. The full Claude API call additionally includes the per-nutrient constraint policy and a required JSON output schema.

Sensory simulator (FORMULATEBENCH-SENSORY objective). The sensory objective is computed by Gemini 3.1 Pro Preview (`google/gemini-3.1-pro-preview` via OpenRouter), temperature 0.1, `max_tokens`=1200, validated independently as a panel-anchored sensory simulator on the NECTAR dataset (NECTAR, 2025; Anonymous, 2026) with accuracy competitive with the median individual human panelist. The prompt is reproduced in Figure 6; the JSON payload also carries the target product's nutrition vector, the candidate recipe's nutrition profile, and a required output schema with `score` (1–10) and `reason` fields. The integer score is parsed and mapped to $[0, 1]$ via $(x - 1)/9$.

Sensory simulator prompt

You are an American omnivore. Here is a plant-based <*category*> formulation: <*ingredients + g per 100 g*>.
How similar do you anticipate the sensory experience will be to animal-based <*category*>?
Output a score from 1-10, with 10 being identical to the animal-based product.

Figure 6. Prompt for the Gemini 3.1 Pro Preview sensory simulator used as the objective in FORMULATEBENCH-SENSORY.

[2]Claude Opus 4.6 was the latest Anthropic model available at the time of the sensory benchmark run; the nutrition benchmark, finalized later, uses 4.7. The cached selections in the released code make this version difference inert for reproduction.

[3]We use temperature 0.2 for the sensory expert because its required output schema includes natural-language `reasoning` and `nutrition_strategy` fields alongside the ingredient indices. A small amount of stochasticity yields more substantive prose in those fields without meaningfully changing the ingredient selection (which is strongly anchored by the structured prompt and the per-nutrient constraint policy). The nutrition expert's output schema contains only the ingredient list, so we use temperature 0 there for full determinism.

E. Synthetic Benchmark: Hartmann6 ($N = 200$)

E.1. Setup

Problem. We embed the standard 6-D Hartmann function into $N = 200$ ambient dimensions; the embedded objective is $f(x) = \text{Hartmann6}(x_{1:6})$, so coordinates $7, \ldots, 200$ are inert and never enter the function. The search domain is $[0, 1]^{200}$. For *subspace methods* (oracle, random subset, REMBO, EGBO), BO optimizes only over the active set S and inactive coordinates are pinned at 0 (matching the support convention $\text{supp}(x) \subseteq S$ used in Assumption 1). Full-dimensional methods (vanilla BO, TuRBO, SAASBO, SEBO) optimize over all 200 coordinates; because f ignores dimensions 7–200, no pinning is required, but TuRBO initializes its trust-region center at $\mathbf{0.5} \in [0, 1]^{200}$ following BoTorch defaults.

Budget. Total budget $T = 20$, of which $n_{\text{init}} = 5$ are Sobol points (BoTorch `SobolEngine`, scrambled, with the i-th Sobol draw seeded as `seed` $+ i$ so that EGBO and Oracle share init points across active sets) and the remaining 15 are acquisition steps with batch size $q = 1$. We run 10 random seeds per (method, regime); shaded bands are 95% confidence intervals from the standard error of the mean.

Acquisition optimization. qLogNEI is optimized with `num_restarts` $= 10$, `raw_samples` $= 512$.

Simulated expert. The expert is implemented in `expert_step()` as independent per-coordinate Bernoulli draws: each of the 6 relevant coordinates is added to the active set with probability p, and each of the 194 irrelevant coordinates with probability q_{fp}. The convergence-curve experiments report two regimes as representatives of the "EGBO wins" and "EGBO loses" regions: a strong expert $(p, q_{fp}) = (0.9, 0.01)$ and a weak expert $(p, q_{fp}) = (0.2, 0.10)$, plus the oracle $(p, q_{fp}) = (1, 0)$. In adaptive runs the active set grows monotonically across rounds.

Baselines.

- **Random search:** uniform $[0, 1]^{200}$.
- **Vanilla BO:** `SingleTaskGP` + qLogNEI on the full 200-D box.
- **Random subset + BO:** fixed random $k = 6$ subset (seed offset $+10\,000$); BO is run inside it with the rest pinned at 0. We intentionally give this baseline the oracle sparsity k so that the comparison isolates the value of *which* variables are chosen, not *how many*.
- **Oracle:** BO restricted to the true relevant 6-D subspace.
- **REMBO** (Wang et al., 2016): Gaussian random projection $A \in \mathbb{R}^{200\times 6}$ with $A_{ij} \sim \mathcal{N}(0, 1/\sqrt{6})$. Latent bounds are computed from 4096 Sobol probes back-projected through A rather than the canonical $[-\sqrt{6}, \sqrt{6}]$ box; this gave consistently better REMBO performance in our setting and we report results under the favorable bounds.
- **TuRBO** (Eriksson et al., 2019): trust-region length $L_0 = 0.8$, $L_{\min} = 2^{-7}$, $L_{\max} = 1.6$, success tolerance 3, failure tolerance $\max(4, d)$ following the original paper's dimension-dependent rule (with batch size $q = 1$).
- **SAASBO** (Eriksson & Jankowiak, 2021): `SaasFullyBayesianSingleTaskGP` with NUTS (warmup 512, samples 256), with the fallback noted in Appendix D.2.
- **SEBO** (Liu et al., 2023): SAAS GP with smooth-ℓ_0 penalty, $\lambda_{\max} = 1.0$, smoothing $a = 0.05$, linear homotopy $\lambda_t = \lambda_{\max}\, t/T$, penalty centered at $x_{\text{ref}} = 0$. These values follow the BoTorch reference implementation rather than the original SEBO settings; we adopted the BoTorch values because they were robust across our $N = 200$ regime without per-task tuning.

Run counts and compute. With 10 seeds, $T = 20$ evaluations, and 9 methods, the Hartmann6 sweep is $9 \times 10 \times 20 = 1{,}800$ function evaluations per heatmap cell; full per-seed traces are written to `results/<NAME>/{oneshot,adaptive}/raw.json`. Runs were executed on a single CPU node; per-seed wall time is dominated by NUTS in SAASBO/SEBO at roughly 5–10 minutes.

Entry points. `run_hartmann6.py` (per-method runs); `run_hartmann6.sh` (fan-out across seeds); `merge_hartmann6.py` and `plot_hartmann6.py` (aggregation and figures).

E.2. Best-so-far convergence (95% CIs)

Figure 7 shows Figure 1 with 95% CIs for all methods.

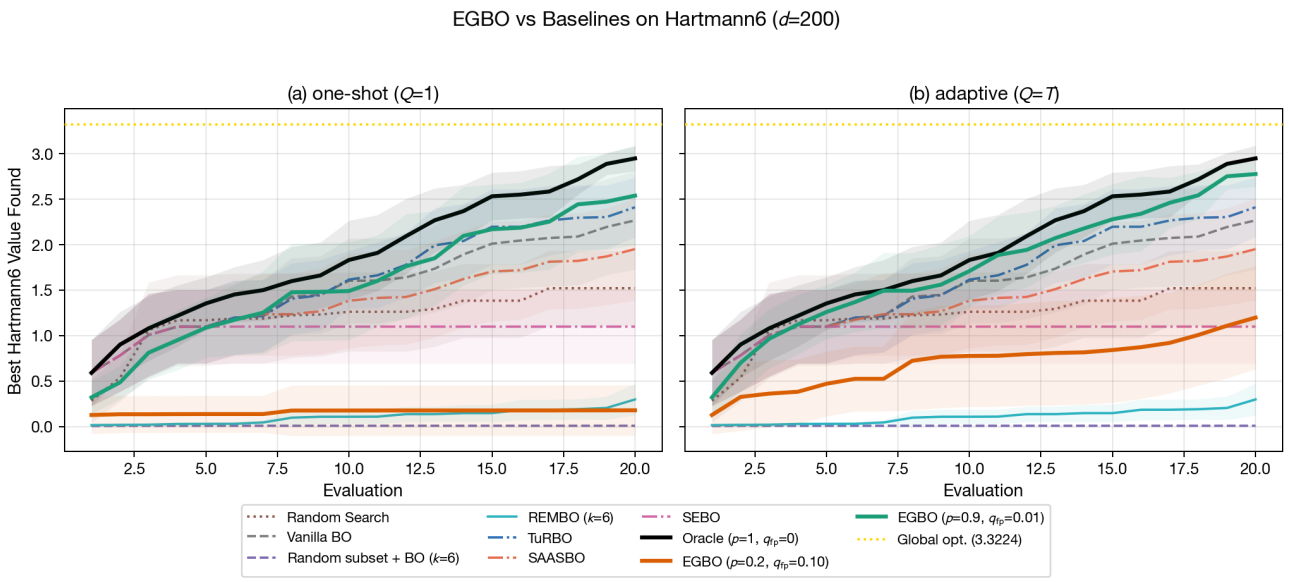

Figure 7. Hartmann6: best-so-far convergence with 95% CIs over 10 seeds.

E.3. Expert-quality heatmap

This experiment maps the (p, q_{fp}) plane empirically to identify the regimes where EGBO improves over vanilla BO. Recall is swept over $p \in \{0.1, 0.2, 0.4, 0.6, 0.8, 0.95\}$ and false-positive rate over $q_{fp} \in \{0, 0.02, 0.05, 0.10, 0.20\}$, all at adaptive $Q = T$ to give the expert the maximum number of opportunities to recover from misses. For each cell we run 10 seeds at $T = 20$ evaluations and report the gap-closed fraction $(f_{\text{EGBO}} - f_{\text{vanilla}})/(f_{\text{oracle}} - f_{\text{vanilla}})$, where 1.0 means EGBO matches the oracle, 0 means it matches vanilla BO, and negative values indicate EGBO underperforms vanilla BO.

Recall is the dominant axis. EGBO closes the majority of the oracle gap whenever $p \geq 0.4$ with relatively mild sensitivity to q_{fp} in this regime; below $p = 0.2$ the gap-closed fraction collapses, and at $p = 0.1$ EGBO is comparable to or worse than vanilla BO regardless of q_{fp}. This is consistent with Theorem 1: false positives enter the bound only through $\varepsilon_{\text{BO}}(T, |S|)$, and at moderate q_{fp} the active set inflates but remains small enough that the BO subroutine still beats search in the full $N = 200$ ambient dimension. The two presets used in Section 4.1, `egbo_strong` at $(0.9, 0.01)$ and `egbo_weak` at $(0.2, 0.10)$, sit in the green corner and on the failure boundary respectively, picked deliberately to illustrate the two regimes.

E.4. Intermediate-Q sweep

This experiment tests Proposition 1's prediction that adaptive expert-query rounds help only when the initial prior is incomplete. We sweep $Q \in \{1, 2, 4, 8, 16, 20\}$ for the two representative presets (`egbo_strong`, `egbo_weak`) at 20 seeds per cell, with all other settings matching Appendix E.1. Note that with bo_left $= T - n_{\text{init}} = 15$ BO iterations available, the maximum number of distinct expert-query rounds is 16; the $Q = 20$ and $Q = 16$ cells therefore produce identical results.

The two regimes split along expert quality, as Proposition 1 predicts. With a strong expert, regret is essentially flat across Q (gap-to-optimum 0.98 at $Q = 1$, 1.19 at $Q = 20$): recall is high enough that one round suffices to cover a near-optimal support, and additional rounds only inflate $|S_q|$ with false positives. The trend is within the 95% CIs and we do not claim statistical significance at $\alpha = 0.05$. With a weak expert, regret drops monotonically from 3.20 at $Q = 1$ to 2.68 at $Q = T$, with the largest improvement between $Q = 8$ and $Q = T$ as the expert is consulted at every BO step. The mechanism is made concrete by the $|S_q|$ trajectories in Appendix E.5: weak experts only assemble a usable support after many rounds, while strong experts have nothing to add.

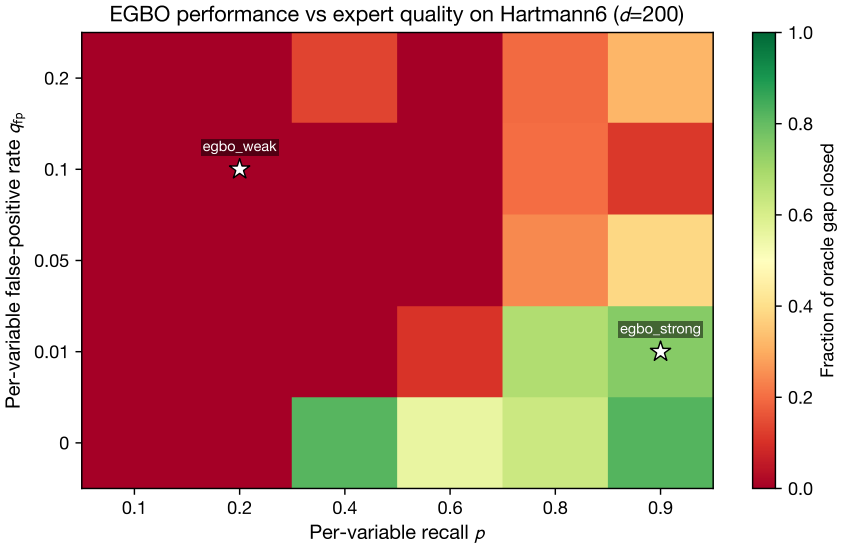

Figure 8. Fraction of the oracle–vanilla gap closed by adaptive EGBO ($Q = T$) on Hartmann6 in $N = 200$ ambient dimensions, as a function of the per-variable recall p (rows) and false-positive rate q_{fp} (columns). Each cell is the mean over 10 random seeds, $T = 20$ evaluations. Stars mark the two presets used in the main text: `egbo_strong` at $(p = 0.9, q_{fp} = 0.01)$ and `egbo_weak` at $(p = 0.2, q_{fp} = 0.10)$. The empirical map of Theorem 1's selection gap: EGBO closes the majority of the oracle gap whenever recall $p \geq 0.4$, with relatively mild sensitivity to q_{fp} in this regime; below $p = 0.2$ the regret rises sharply, and at $p = 0.1$ EGBO is comparable to or worse than vanilla BO over the full 200-dimensional space regardless of q_{fp}.

E.5. Active-set trajectory

Figure 10 visualizes the active-set dynamics predicted by Equation (4) directly on the Hartmann6 task. For each (Q, expert) configuration we record $|S_q|$ (total active-set size) and $|S_q \cap \mathcal{U}_\eta|$ (number of relevant variables in the active set, where $|\mathcal{U}_\eta| = 6$ since coordinates 7–200 are inert) at every BO iteration, averaged over 20 seeds. We plot three representative Q values: $Q = 1$ (no expansion after initialization), $Q = 4$ (expand every four BO iterations), and $Q = 20$ (expand every BO step), under both expert presets.

The two panels expose the mechanism behind the Q-sweep regret pattern in Appendix E.4. Under a strong expert (left), $|S_q|$ jumps near the true support size in the first round and the dashed line ($|S_q \cap \mathcal{U}_\eta|$) tracks the solid one closely throughout: the relevant support is essentially covered after one query, so larger Q only adds the slow accumulation of false positives at rate $q_{fp} = 0.01$ per round per inert coordinate. Equation (4) predicts $\mathbb{E}[|S_Q|]$ rises from ≈ 7.3 at $Q = 1$ to ≈ 41 at $Q = 20$ in this regime; the resulting BO dimension cost is what produces the slight upward regret trend in Appendix E.4. Under a weak expert (right), $|S_q|$ grows slowly across rounds for small Q and aggressively for large Q, with the dashed line approaching the $|\mathcal{U}_\eta| = 6$ ceiling only by mid-trajectory at $Q = 20$. This is the regime in which adaptivity pays off: each additional round provides a fresh chance to recruit a missing relevant variable, at the cost of a large absolute $|S_q|$ dominated by false positives ($\mathbb{E}[|S_{20}|] \approx 176$ by Equation (4)).

Together, this active-set view, the Q-sweep regret in Appendix E.4, and the (p, q_{fp}) heatmap in Appendix E.3 align: the green band of the heatmap corresponds to fast $|S_q \cap \mathcal{U}_\eta|$ saturation with controlled $|S_q|$, while the red band corresponds to either incomplete coverage (low p) or excessive $|S_q|$ (high q_{fp}).

F. FORMULATEBENCH

F.1. Setup

Problem. The ground set is the union of plant-based ingredients in NECTAR with nutrition data in USDA FoodData Central ($d = 111$ for meat, $d = 91$ for dairy). The objective is the per-dimension z-score RMSE against an animal-product target on $m = 8$ nutrients (energy, protein, fat, saturated fat, carbohydrates, fiber, sugars, sodium), with a per-category measurement mask $M \subseteq \{1, \ldots, 8\}$ for missing USDA values; ingredient-set z-score statistics (μ_j, σ_j) are computed once over the candidate ground set.

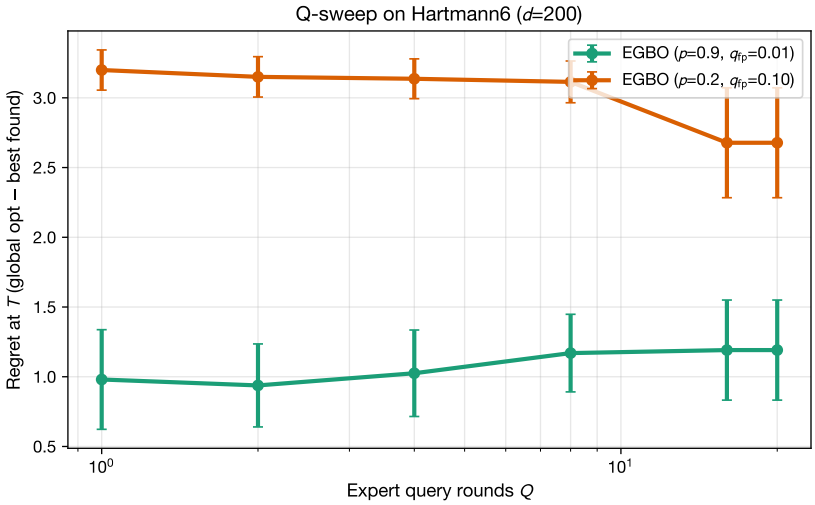

Figure 9. Final regret (gap to the global optimum, lower is better) as a function of the number of expert query rounds Q on Hartmann6 in $N = 200$ ambient dimensions, $T = 20$ evaluations, 20 random seeds per cell. Error bars are 95% confidence intervals. With a strong expert ($p = 0.9$, $q_{fp} = 0.01$, green), regret is essentially flat across Q and trends slightly upward from 0.98 at $Q = 1$ to 1.19 at $Q = T$. Adaptive querying does not help, and can mildly hurt, a strong expert because per-iteration false positives accumulate and grow $|S_q|$ beyond the true 6-dimensional support, making the BO problem harder under a fixed evaluation budget (the active-set trajectory in Appendix E.5 makes this growth concrete). The trend is within the 95% CIs and we do not claim statistical significance at $\alpha = 0.05$. With a weak expert ($p = 0.2$, $q_{fp} = 0.10$, orange), regret drops from 3.20 at $Q = 1$ to 2.68 at $Q = T$, with the largest improvement between $Q = 8$ and $Q = T$ once the expert is consulted at every BO step. $Q = 20$ coincides with $Q = 16$ because bo_left $= T - n_{\text{init}} = 15$ is the maximum number of distinct expert-query iterations for this budget. The two regimes match the prediction of Proposition 1: adaptivity helps only when the prior is incomplete.

Simplex parameterization. Compositions live on Δ^{d-1} and are parameterized as logits $z \in [-12.0, 0.0]^d$ mapped through a softmax; the -12 floor ($e^{-12} \approx 1.5 \times 10^{-6}$) keeps gradients stable while leaving the simplex effectively open. The first initialization point is the uniform mixture and the remaining $n_{\text{init}} - 1$ are Dirichlet draws.

Sparsity bound justifying $k = 9$. When the objective is nonnegative least squares with m nutrient targets, a classical result gives a bound on the sparsity of optimal solutions; we use this to set the LLM expert's selection cap to $k = m + 1 = 9$.

Proposition 2 (Sparsity of NNLS nutrition matching)**.** *Let $V \in \mathbb{R}^{m \times N}_{\geq 0}$ be a nonnegative ingredient-to-nutrient matrix and $v^* \in \mathbb{R}^m$ a target nutrient profile. Consider the nonnegative least-squares problem*

$$\min_{x \geq 0} \|Vx - v^*\|_2^2. \tag{8}$$

There exists an optimal solution x^* with $\|x^*\|_0 \leq m$. If the problem additionally imposes a simplex constraint $\sum_{i=1}^N x_i = 1$, there exists an optimal solution with $\|x^*\|_0 \leq m + 1$.

Proof. Among all optimal solutions, pick one for which $|I| = \|x^*\|_0$ is minimal, and let $I = \text{supp}(x^*)$. Suppose for contradiction that $|I| > m$. Then the columns $\{V_i : i \in I\}$ are linearly dependent in $\mathbb{R}^m$, so there exists $d \neq 0$ with $V_I d = 0$. Extending d by zeros outside I, the perturbation $x^* + td$ preserves the objective for all t. Choose t with largest absolute value such that $x^* + td \geq 0$; at least one coordinate becomes zero, contradicting minimality of $|I|$. Hence $|I| \leq m$.

The simplex-constrained version follows the same argument with d restricted to $\ker V_I \cap \{d : \sum_i d_i = 0\}$, the null space of $\begin{pmatrix} V_I \\ \mathbf{1}^\top \end{pmatrix} \in \mathbb{R}^{(m+1) \times |I|}$, which has dimension at least $|I| - (m + 1)$. If $|I| > m + 1$ this is non-trivial, and the rest of the argument gives $|I| \leq m + 1$. When no box constraint binds, the argument produces a k-sparse optimum for any k at least the stated bound, so $\eta(k) = 0$. □

Budget. $T = 20$, $n_{\text{init}} = 5$, batch size $q = 1$, 10 random seeds per (method, category). With 24 categories and 8 methods, this is $24 \times 10 \times 8 = 1{,}920$ BO runs and $1{,}920 \times 20 = 38{,}400$ objective evaluations per full sweep.

Acquisition optimization. Same as Hartmann6: qLogNEI, `num_restarts` $= 10$, `raw_samples` $= 512$.

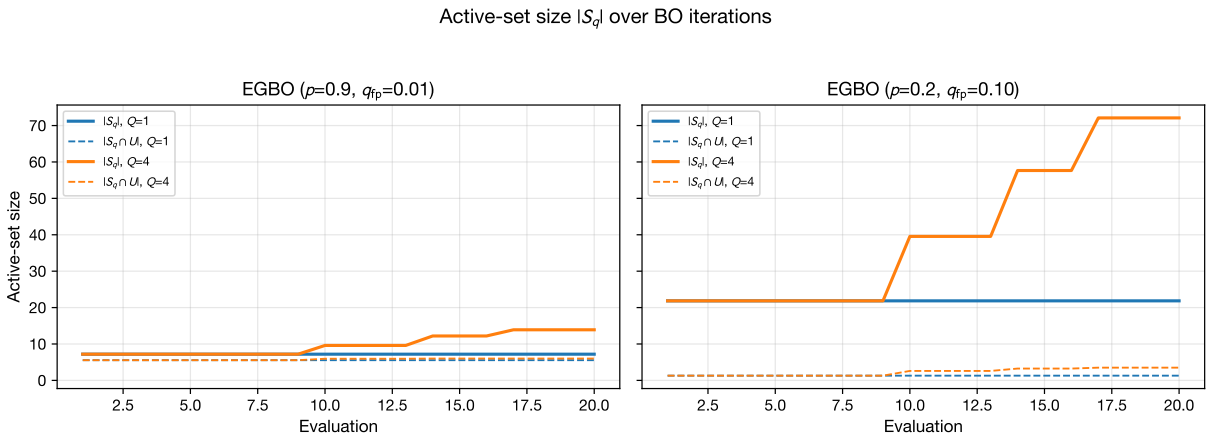

Figure 10. Mean active-set size $|S_q|$ over BO iterations for three representative Q values $\{1, 4, 20\}$ on Hartmann6 in $N = 200$ ambient dimensions, 20 random seeds. *Left:* strong expert ($p = 0.9, q_{fp} = 0.01$). *Right:* weak expert ($p = 0.2, q_{fp} = 0.10$). Solid lines plot $|S_q|$ (total active-set size). Dashed lines plot $|S_q \cap U_\eta|$ (number of relevant variables in the active set; $|U_\eta| = 6$ for embedded Hartmann6). Under a strong expert $|S_q|$ saturates within one round near the true support size; under a weak expert, $|S_q|$ grows slowly across many rounds before it covers a near-optimal support, and false positives accumulate when Q is large. This visualization makes Equation (4) of Proposition 1 concrete: low recall p makes round-by-round coverage of $\mathcal{U}_\eta$ unlikely, so weak experts require many rounds before the active set is useful.

Baselines. Same library settings as Hartmann6 with two configuration differences: (i) all methods optimize over the simplex logits described above, with inequality constraints handled by `optimize_acqf`; (ii) the Random-subset-plus-BO baseline uses a fixed random $k = 9$ subset (seed offset $+10\,000$), matching EGBO's active-set size for a fair head-to-head. REMBO uses a Gaussian projection $A \in \mathbb{R}^{d\times 9}$ with latent radius $\sqrt{9} = 3$. TuRBO uses failure tolerance $\max(4, d)$ as in Hartmann6.

LLM expert. Active-set selection uses Claude Opus 4.7 at temperature 0 as documented in Appendix D.3; the prompt is reproduced in Figure 4.

Compute. FORMULATEBENCH runs were executed on CPU. The SAAS-based methods (SAASBO, SEBO) dominate wall time at ∼2–5 minutes per (seed, category, method); the full sweep took on the order of 10^2 CPU hours.

Entry point. Run `python run_formulatebench.py --domain {meat,dairy} --budget 20 --seeds 10 --n-init 5`. Per-seed traces are written atomically as JSON.

F.2. Per-category nutrition results

Figures 11 and 12 show results in each category of meat and dairy.

F.3. Adaptive vs. one-shot ablation

Setup. We re-run FORMULATEBENCH-NUTRITION with adaptive EGBO at Q = 5, using identical settings to the one-shot benchmark above except: the expert is queried 4 additional times during BO (so 5 expert rounds total), and may suggest up to 2 new ingredients per round (`additions_per_round`=2). The active set grows from $|S_1| = 9$ to $|S_5| \leq 17$. All other settings are identical: $T = 20$, $n_{\text{init}} = 5$, 10 seeds per category, Claude Opus 4.7 (temperature 0) as the expert via OpenRouter. Past observations remain valid as the active set grows: a previous evaluation $x \in \Delta^{|S_q|}$ is interpreted as $\tilde{x} \in \Delta^{|S_{q+1}|}$ with zero weight on the new ingredients (per the *monotonic active-set growth* convention from Algorithm 1).

Adaptivity does not improve over one-shot. Figure 13 shows pooled best-so-far z-RMSD across all 24 categories × 10 seeds for Q = 1 (one-shot, green) and Q = 5 (adaptive, purple). Across both domains, the mean final z-RMSD at Q = 5 is statistically indistinguishable from Q = 1:

- Meat (14 categories): $\Delta(\text{Q} = 5 - \text{Q} = 1) = +0.006$, mean relative change -3.1%, win-rate $6/14$.

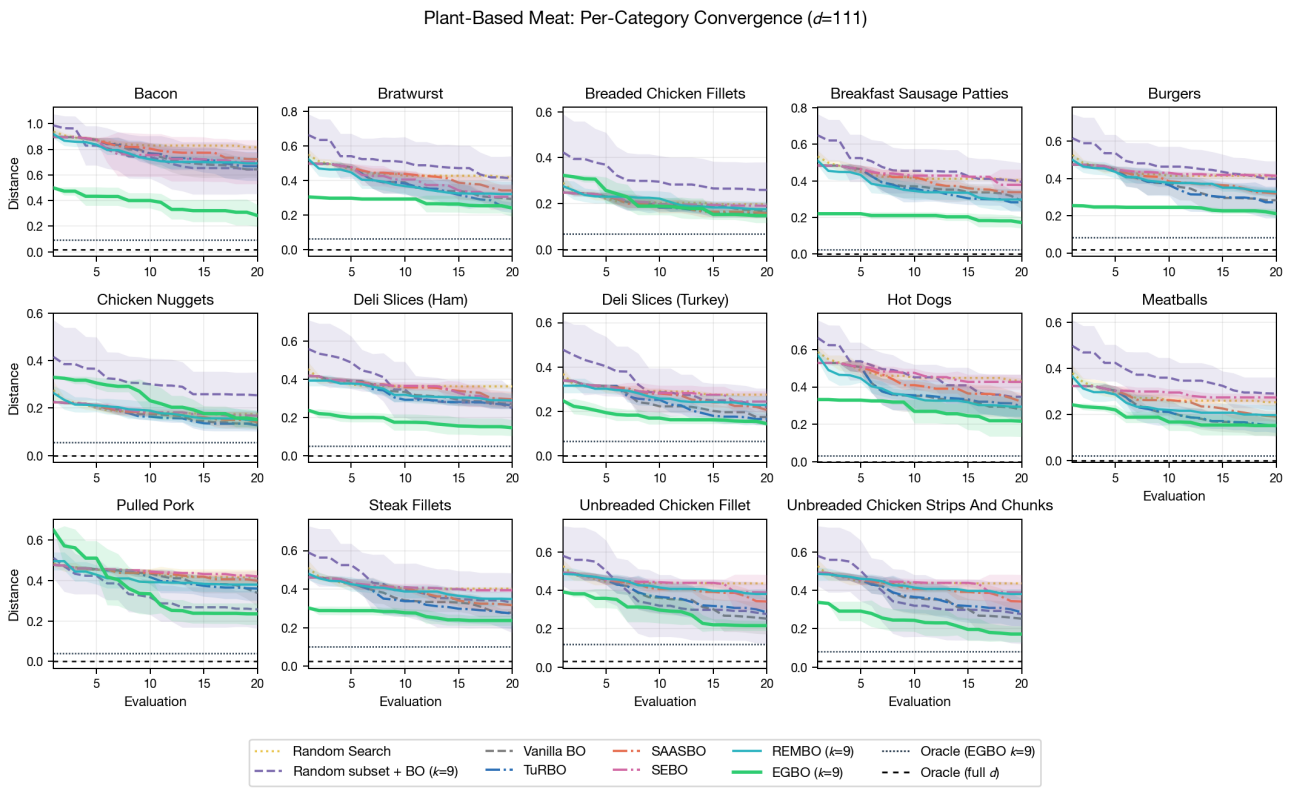

Figure 11. FORMULATEBENCH results for meat categories.

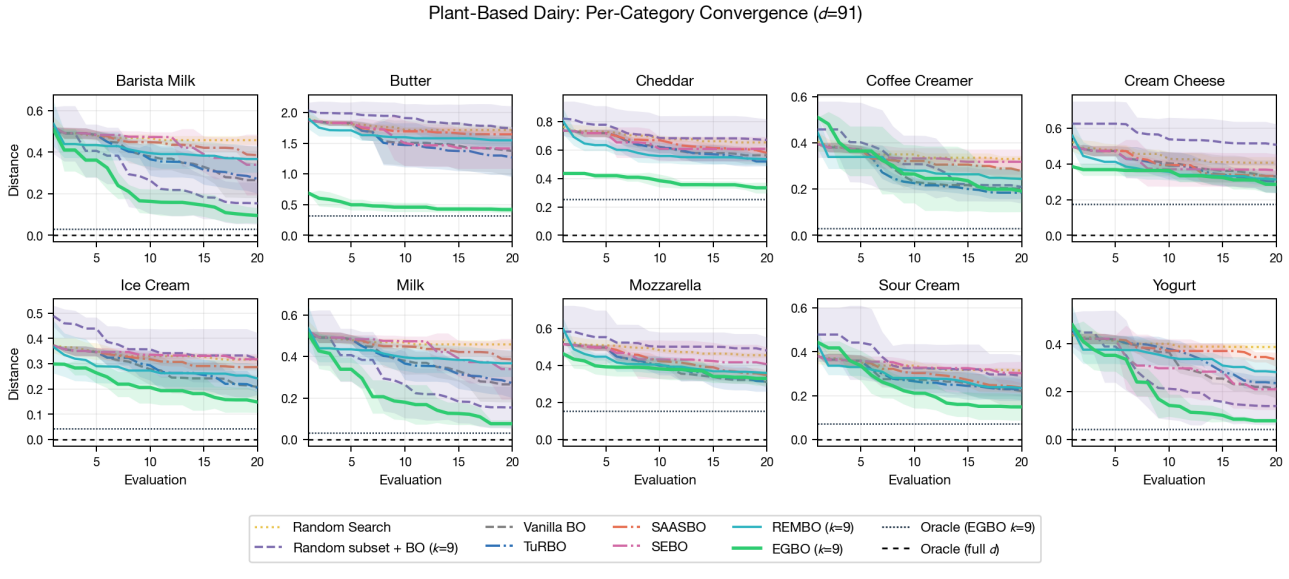

Figure 12. FORMULATEBENCH results for dairy categories.

- Dairy (10 categories): $\Delta = +0.002$, mean relative change -2.7%, win-rate $2/10$.

Per-category Δ values are within seed SEM. The categories where adaptive querying does help (e.g., butter $+8.2\%$, coffee creamer $+12.0\%$, deli ham $+10.1\%$) have the largest selection gap ($\text{Oracle}_{k=9} - \text{Oracle}_{\text{full}}$), matching Proposition 1: adaptive querying offers measurable gain only when the one-shot prior leaves selection-gap headroom. For the median FORMULATEBENCH category the LLM's $k = 9$ already spans a near-optimal NNLS support, so additional rounds expand the active set without improving achievable loss; the larger active set makes the BO problem harder under fixed budget. This negative result motivates the deployment design choice in 5, which uses one-shot EGBO.

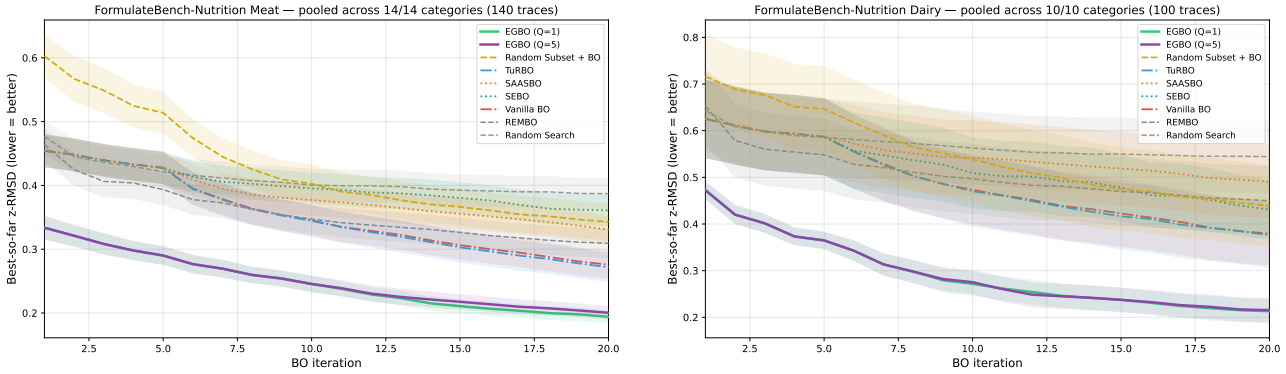

Figure 13. Adaptive (Q $= 5$, purple) vs. one-shot (Q $= 1$, green) EGBO on FORMULATEBENCH-NUTRITION: pooled best-so-far z-RMSD (↓) across all categories $\times$ 10 seeds, with 95% CI bands. Adaptive querying does not improve over one-shot.

F.4. Sensory variant

Setup. The sensory variant uses Gemini 3.1 Pro Preview as a panel-anchored sensory simulator (full prompt in Appendix D.3); it was independently validated on the NECTAR dataset (NECTAR, 2025; Anonymous, 2026), with accuracy competitive with the median individual human panelist. The LLM expert is Claude Opus 4.7 prompted to select a $k = 9$ ingredient subset that "replicates the sensory experience of [CATEGORY] with plant-based ingredients, while matching or improving on nutrition" (full prompt in Appendix D.3). Other settings: $T = 20$, $n_{\text{init}} = 10$, 5 seeds per category, $k = 9$ for EGBO and the random-subset+BO baseline, identical method implementations to the nutrition benchmark.

Results. Figure 14 shows Gemini-3.1-Pro animal-similarity per domain. On plant-based dairy, one-shot EGBO is the clear winner: mean similarity 0.22, against 0.12 for the next-best baseline (random subset + BO at $k = 9$, an 84% relative gain) and ≤ 0.05 for dense BO baselines. On plant-based meat, EGBO and random-subset+BO are within noise of each other (0.28 vs 0.29); both substantially outperform dense BO baselines.

We hypothesize that the LLM's plant-based food prior is more developed for dairy than for meat: plant-based dairy ingredients have been documented in detail in training corpora since the 1990s, while plant-based meat is more recent and proprietary; plant-dairy similarity is bounded by texture and basic flavor that plant ingredients can directly substitute, while plant-meat similarity is bounded by structural-protein networks that no plant ingredient fully mimics. We do not attempt to disentangle these mechanisms, but these results support our choice of dairy categories in our the deployment in Section 5.

F.5. Diagnostic: is the LLM doing greedy NNLS variable selection?

A natural concern about FORMULATEBENCH-NUTRITION is that the LLM expert is shown per-ingredient nutrient vectors and is essentially solving NNLS variable selection by greedy reasoning ("which 9 columns of the $d \times m$ nutrient matrix best span the target?"). If true, EGBO's performance would reduce to the LLM's NNLS-solving ability rather than food-science expertise.

We probe this directly by comparing the LLM's $k = 9$ selections to a *greedy-NNLS oracle* that iteratively adds the ingredient minimizing the residual NNLS loss restricted to the selected set + new candidate (under the simplex constraint). We compute three quantities per category: $\text{Oracle}_{\text{LLM}}$, $\text{Oracle}_{\text{greedy}}$, and a random-subset baseline $\text{Oracle}_{\text{random}}$ averaged over 5 seeds. We also report set overlap $|S_{\text{LLM}} \cap S_{\text{greedy}}|$.

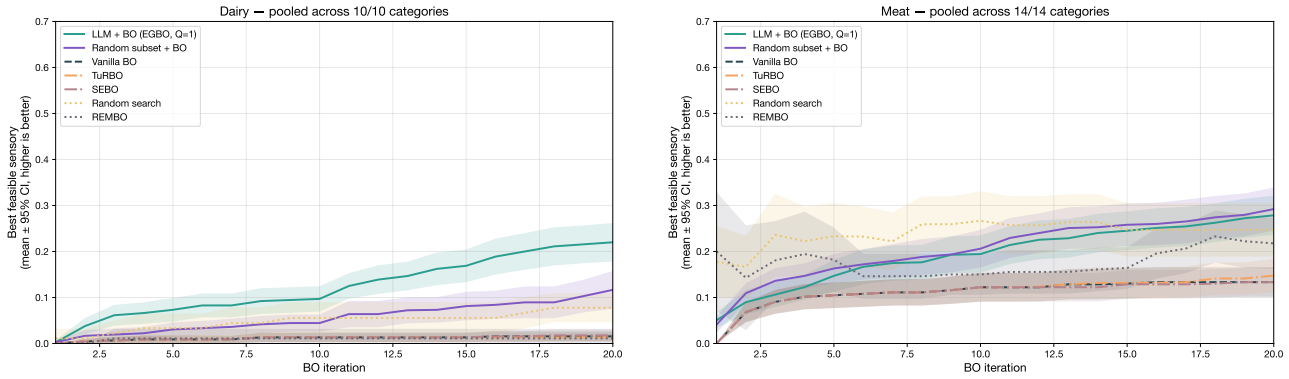

Figure 14. Best Gemini 3.1 Pro animal-similarity score (higher is better) among recipes that satisfy per-category nutrition constraints, over $T = 20$ BO iterations, 95% CI bands. *Left:* 10 plant-based dairy categories $\times$ 5 seeds. *Right:* 14 plant-based meat categories $\times$ 5 seeds. One-shot EGBO with a Claude Opus 4.6 expert ($k = 9$) substantially outperforms all baselines on dairy; on meat, EGBO ties random-subset+BO ($k = 15$) and both outperform dense BO baselines.

Across all 24 categories, the LLM's selection is *systematically worse* than greedy-NNLS at the NNLS objective:

	$\text{Oracle}_{\text{full}}$	$\text{Oracle}_{\text{LLM}}$	$\text{Oracle}_{\text{greedy}}$	$\lvert S_{\text{LLM}} \cap S_{\text{greedy}} \rvert$
Meat (14)	0.009	0.065	0.009	0.86/9
Dairy (10)	0.000	0.116	0.001	1.50/9

The LLM beats greedy-NNLS on the NNLS objective in $0/24$ categories. The two selections share on average ≈ 1 ingredient out of 9. Inspecting the picks shows the divergence is qualitative: greedy-NNLS picks mathematically-optimal but practically-implausible ingredients (e.g., *rowanberry extract* for cheddar, *potassium citrate* for yogurt, *macadamia milk* for butter), while the LLM picks recognizable plant-based food-grade ingredients (carrageenan and gellan gum as gelling agents, oat / soy / pea milk as dairy bases, cocoa butter and cashews as plant-cheese fats). We conclude that the LLM is *not* solving the NNLS variable-selection problem: its choices are systematically different from and worse than the NNLS-greedy choice. Instead, the LLM appears to be applying a food-science prior, preferring ingredients that are realistic for plant-based products and commonly used in commercial reformulation, which is suboptimal for NNLS but remains useful for downstream BO that runs in the chosen subspace. This reframes the EGBO success on FORMULATEBENCH-NUTRITION not as "LLMs are good at NNLS" but as "LLMs apply a food-science variable-selection prior that, while NNLS-suboptimal, supports better downstream BO than dense baselines on the original $d \approx 100$ space."

G. Real-World Deployment

G.1. Shared Setup

Problem. The candidate ingredient pool is a food-company catalog of $N \approx 88{,}000$ entries. Prior to optimization, the company's Food Scientist Agent (FSA, an LLM-based agentic framework) performs one-shot variable selection ($Q = 1$) yielding $d = 7$ active ingredients for yogurt and $d = 9$ for cottage cheese; all other $N - d$ ingredients are fixed at zero. Variables are encoded as continuous quantities with per-ingredient $[\min, \max]$ bounds taken from the company catalog; property and quantity constraints are enforced via linear inequality constraints passed to `optimize_acqf`.

Budget and batching. Each campaign runs $T = 10$ iterations with a laboratory batch of $q = 3$ recipes per iteration, for 30 total recipes. The first $n_{\text{init}} = 5$ evaluations are scrambled Sobol points drawn from a pool of 2,048 feasible candidates (Sobol seed equals the iteration count for reproducibility); subsequent batches are produced by qLogNEI with `num_restarts` $= 20$, `raw_samples` $= 1024$, `sample_shape` $= [512]$, and `sequential` = True so each of the three batch members is selected greedily conditional on the previously chosen ones. The Streamlit application optionally enables an Input Warping kernel (`robust` = True) for non-stationarity; the campaigns reported here use the default unwarped configuration.

FSA seed. The FSA's initial recipe is loaded as observation $(x^{(0)}, y^{(0)})$ with $y^{(0)} = f(x^{(0)})$ obtained by panel scoring of the FSA recipe (0.767 for yogurt, 0.776 for cottage cheese). It counts toward the budget and seeds the GP before any BO step.

Utility. Sensory scores are entered through the Streamlit UI (`app.py`); `food_bo.py:_compute_utility` normalizes each attribute to $[0, 1]$ using the configured panel scale and converts it to a per-attribute utility according to the goal (max/min/target); the scalar objective is the weighted sum $\sum_i w_i u_i$ with uniform weights $w_i = 1/m$, matching the main-text utility definition.

Reproducibility. Sobol scrambling and a `torch.manual_seed` call both use `len(self.X_history)` so reruns at any iteration deterministically reproduce the proposed batch given the recorded panel scores. Campaign state is persisted to pickle and exported as JSON. We release the full optimization variables, ingredient lists, panel protocols, and per-iteration recipes alongside the code.

Compute. Each iteration's acquisition optimization runs in under a minute on a single CPU; the binding constraint is laboratory time ($\sim$4 hours per recipe batch), giving $\sim$40 hours per campaign.

G.2. Recipe Formulations

This appendix documents the output of FSA, final ingredient compositions and processing protocols for two plant-based products developed under the EGBO framework: the plant-based yogurt corresponding to the closed-loop deployment, and an analogous plant-based cottage cheese formulation. Quantities are reported as supplied to the laboratory. Ingredients held fixed during optimization are flagged as constants; all others were varied by EGBO.

G.2.1. Plant-Based Yogurt

The best yogurt formulation identified at iteration 7 of the campaign is reported in Table 5. The total batch mass is 630.00 g. Soy protein isolate and tricalcium phosphate were held constant across all iterations, serving as a functional protein backbone and a calcium fortificant respectively; the remaining seven ingredients constitute the active EGBO design space.

Table 5. Plant-based yogurt: ingredient composition of the best recipe (iteration 7, recipe 2). Total batch mass 630.00 g.

Ingredient	Mass (g)	% w/w	Status
Coconut cream, canned, sweetened	340.00	53.97	varied
Tap water	237.75	37.74	varied
Lemon juice, raw	20.00	3.17	varied
Nutritional yeast	10.00	1.59	varied
Salt	7.00	1.11	varied
Soy protein isolate	6.30	1.00	constant
Tapioca starch	6.00	0.95	varied
Tricalcium phosphate	1.95	0.31	constant
Xanthan gum	1.00	0.16	varied
Total	630.00	100.00	

The yogurt was prepared according to the following three-step protocol.

Step 1: Powder dispersion. All ingredients listed in Table 5, including the calcium fortificant, were combined and dissolved using a high-shear mixer at 3000 RPM, 25 °C, for 5 min.

Step 2: Cooking. The dispersed yogurt base was cooked using a moist-heat boiling protocol in a covered pot (cover closure 90%) at 90 °C for 8 min.

Step 3: Chilling. The cooked product was transferred to a refrigerator and chilled prior to sensory evaluation.

G.2.2. PLANT-BASED COTTAGE CHEESE

Table 6 reports the ingredient composition of the plant-based cottage cheese formulation. The first two ingredients form the curd-forming substream, which is subsequently centrifuged to remove whey; the remaining seven ingredients form the cooked dressing, which is combined with the rinsed curds. Weight fractions are reported as supplied to the process and therefore total slightly above 100%, because the soy whey separated in Step 3 is discarded as a byproduct and is not present in the final product.

Table 6. Plant-based cottage cheese: ingredient composition. Curd-substream ingredients (top) are coagulated and centrifuged before being recombined with the cooked dressing (bottom).

Ingredient	% w/w	Substream
Soy milk, unsweetened, plain, shelf stable	64.0000	curd
Distilled vinegar	4.0000	curd
Tap water	17.0667	dressing
Coconut cream, canned, sweetened	13.8667	dressing
Lemon juice, raw	1.0667	dressing
Nutritional yeast	0.6400	dressing
Salt	0.4267	dressing
Tapioca starch	0.4267	dressing
Xanthan gum	0.1067	dressing

The cottage cheese was prepared according to the following seven-step protocol.

Step 1: Pasteurization. Soy milk was heat-treated for microbial-load reduction using a batch pasteurizer at $90\ ^\circ\mathrm{C}$ for $5\ \mathrm{min}$ with full cover closure (100%). Output: pasteurized soy milk.

Step 2: Acid coagulation. Pasteurized soy milk was combined with distilled vinegar in a mixing vessel operated at $200\ \mathrm{RPM}$, $75\ ^\circ\mathrm{C}$, for $2\ \mathrm{min}$ with 80% cover closure. Output: coagulated soy mixture.

Step 3: Curd separation. The coagulated mixture was separated by basket centrifugation at $800\ \mathrm{RPM}$ ($50\ \mathrm{Hz}$ drive frequency), $25\ ^\circ\mathrm{C}$, for $3\ \mathrm{min}$ in a single pass. Outputs: soy curds; soy whey (discarded).

Step 4: Curd rinsing. Soy curds were washed with tap water in a washing machine at $10\ ^\circ\mathrm{C}$ for $1\ \mathrm{min}$ (single pass). Outputs: rinsed soy curds; spent rinse water.

Step 5: Dressing cooking. Coconut cream, tap water, lemon juice, salt, nutritional yeast, tapioca starch, and xanthan gum were combined and cooked using a moist-heat boiling protocol in a covered pot (cover closure 90%) at $90\ ^\circ\mathrm{C}$ for $8\ \mathrm{min}$. Output: cooked plant-based dressing.

Step 6: Combination. Rinsed soy curds and cooked dressing were combined in a planetary mixer at $60\ \mathrm{RPM}$, $20\ ^\circ\mathrm{C}$, for $2\ \mathrm{min}$. Output: unchilled plant-based cottage cheese.

Step 7: Chilling. The product was cooled to $4\ ^\circ\mathrm{C}$ in a refrigerator. Output: plant-based cottage cheese (final).

G.3. Sensory evaluation protocol

The utility scores reported in Section 5 were obtained via a structured descriptive sensory evaluation conducted by a trained panel of five participants. The procedure was held fixed across all $T = 10$ EGBO iterations to ensure that observed utility differences reflect genuine recipe-level differences. The sensory panel is treated as a noisy black-box oracle, where for a formulation $x \in \mathbb{R}^N$, the observed score is $y = f(x) + \varepsilon$, with ε capturing inter-panelist and session variability.

Panel composition and preparation. All panelists were members of a trained sensory panel with prior experience in structured attribute-based evaluation, and each session was preceded by a calibration briefing aligning the panel on the target attributes and on the 0–15 intensity scale. To ensure a neutral palate at the time of evaluation, panelists abstained from

Figure 15. A photo of the environment in which the sensory panels were conducted.

food consumption for at least two hours before the session, and from strong-flavored stimuli (spicy or acidic foods, alcohol, coffee, smoking) for at least six hours before the session.

Photo of sensory panel environment Figure 15 shows the location of the sensory panels.

Evaluation environment. Sessions were conducted in a dedicated sensory evaluation room designed to minimize external bias. The space was isolated from ambient odors and distractions, and environmental conditions (temperature, noise level, airflow) were held stable across sessions. Controlled lighting was applied to suppress visual bias on attributes such as color and surface appearance. Panelists worked in individual booths, did not communicate during the session, and did not discuss or compare samples. Each panelist evaluated samples in a predefined sequence determined prior to the session.

Calibration system. A Golden Standard reference sample was made available throughout each session as a continuous calibration anchor. Panelists were instructed to revisit the reference between samples to maintain a stable internal reference point, and printed 0–15 attribute scales were provided before the start of evaluation. This combination of a physical reference and an explicit scale was used to control for within-session and between-session drift in panelist perception.

Evaluation procedure. The protocol followed a comparative descriptive analysis approach: attribute-based, comparative against the Golden Standard, and non-hedonic (preference and liking are not recorded). Each sample was assessed via a fixed sequence of stages, each applied where relevant to the product class:

- **Visual assessment.** Density, surface characteristics, structural integrity, and overall first impression.
- **Aroma.** Identification of intensity and key aromatic notes under controlled exposure to avoid sensory fatigue.
- **Texture and mouthfeel.** Initial perception on the first spoon or bite, creaminess and slipperiness, graininess, and cohesion or structural behavior under deformation.
- **Flavor.** Basic taste profile (sweet, sour, salty, bitter, umami), flavor identity (e.g., dairy, fermented, clean, off-notes), balance, and aftertaste.
- **Overall assessment.** Holistic evaluation and identification of key strengths and weaknesses.

For the plant-based yogurt campaign, the four target attributes (Consistency, Creaminess, Tanginess, Uniformity) were drawn from the Texture and Flavor stages above and recorded on the 0–15 scale; for plant-based cottage the same was applied for the six target attributes (Homogeneity, Dispersion Phase, Slipperiness, Sourness, Dairy and Off-Notes).

Between-sample protocol. To preserve sensory clarity across samples, panelists cleansed their palate with water between evaluations, with neutral carriers (bread and crackers) available as an option, and observed a short reset period before proceeding to the next sample. The Golden Standard was re-tasted as needed during this interval to realign perception.

Objectivity, fatigue management, and performance principle. The panel operated under strict objectivity principles: evaluation was based on the defined sensory attributes rather than personal preference, no response was treated as right or wrong, and consistency across samples was prioritized over individual bias. When sensory fatigue or attribute confusion arose, panelists followed a four-step recovery protocol: pausing evaluation, cleansing the palate, re-evaluating the Golden Standard, and then resuming testing. The overall protocol prioritized accuracy, consistency, and repeatability of the recorded scores; speed of evaluation was treated as secondary to data quality.

Data aggregation. Panelist scores were aggregated immediately after evaluation into a single set of attribute-level scores. Individual panelist responses were not retained, and no individual-level data were used in subsequent analysis. The utility values reported in the main text were computed from these aggregated scores using the scalar utility function defined in Section 5.

G.4. Yogurt campaign

Sensory convergence. Figure 16 shows the trajectory of all four sensory attributes. Consistency rose from 5 at baseline to 12 (exact target) by iteration 4, and remained within ± 1 of target for all subsequent iterations. Creaminess increased from 3 to 5 (exact target), first reaching it at iteration 4 and stabilizing from iteration 7 onward. Tanginess, which required a decrease from 10 to 5, proved the most volatile attribute, dropping sharply in early iterations but oscillating between 3 and 7 through the middle of the campaign before converging to 5 at iteration 7. Uniformity, which started at 13 (on target), drifted to values between 10 and 13 during the exploration-heavy early iterations but recovered to 12.5–13 in the final batches. The best single recipe (iteration 7, recipe 2) achieved scores of $(12,\ 5,\ 5,\ 12.5)$, matching the target exactly on three of four attributes and falling only 0.5 points short on Uniformity.

Utility progression. Figure 17 reports the maximum utility per iteration and the cumulative best-so-far. The utility climbed steeply in the first four iterations, reaching 0.950 by iteration 4. A second jump to 0.983 occurred at iteration 6, and the campaign peak of 0.992 was reached at iteration 7. The best-so-far remained at or above 0.983 from iteration 6 onward, indicating that the optimizer had largely converged by the sixth batch and subsequent iterations refined rather than explored. The residual gap to perfect utility is $1 - 0.992 = 0.008$.

Ingredient concentration trajectories. Figure 18 shows how the seven active ingredients evolve over the campaign.

Final formulation. The best yogurt formulation identified at iteration 7 is reported in Table 5. Total batch mass is 630.00 g. Soy protein isolate and tricalcium phosphate were held constant across all iterations, serving as a functional protein backbone and a calcium fortificant respectively; the remaining seven ingredients constitute the active EGBO design space.

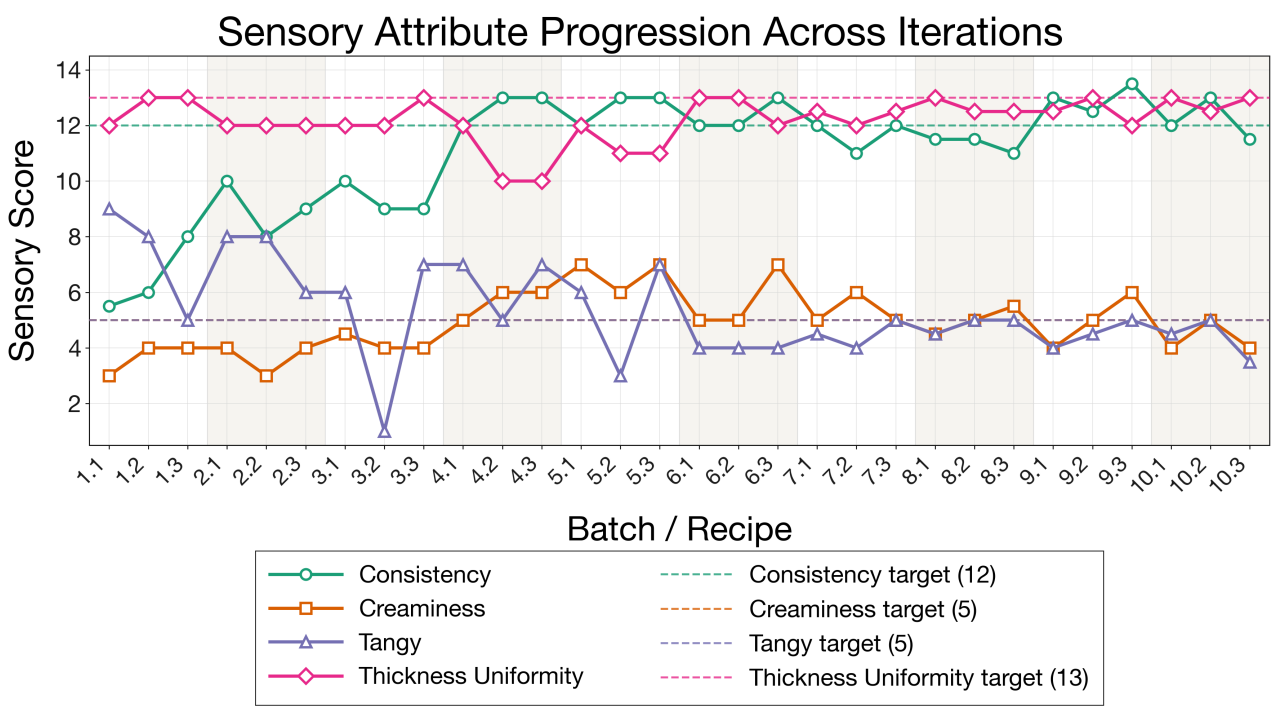

Figure 16. Sensory attribute trajectories across the 10 EGBO iterations (3 recipes per iteration) for the plant-based yogurt campaign. Solid lines show observed panel scores; dashed lines indicate the target value for each attribute. Consistency and Creaminess converge to their targets by iteration 7; Tanginess exhibits the highest volatility before settling at target; Uniformity remains near target throughout with minor drift.

The yogurt was prepared in three steps: **(1) Powder dispersion** — all ingredients combined and dissolved with a high-shear mixer at 3000 RPM, 25 °C, for 5 min. **(2) Cooking** — moist-heat boiling protocol in a covered pot (cover closure 90%) at 90 °C for 8 min. **(3) Chilling** — refrigerated prior to sensory evaluation.

Iteration photos. Figures 19–28 show the visual evolution of the yogurt across iterations.

G.5. Cottage cheese campaign

Sensory convergence. Figure 29 shows the trajectory of all six sensory attributes. Off-Notes, which required a decrease from 7 to 0, showed the most dramatic improvement, dropping steadily from 7 to 1.5 by the final iterations. Sour decreased from 10 toward its target of 6.5, stabilizing in the range 6–7 from iteration 7 onward. Dairy, which started at 3 against a target of 7, proved the most difficult attribute to improve, rising gradually to 5 by the final iterations but not fully reaching target (which is expected since this is a plant-based dairy recipe). Homogeneity improved from 4 to values consistently near 6 (target) in the second half of the campaign. Dispersion Phase and Slipperiness, both starting well above their targets of 6, converged toward target by iteration 8, though with residual volatility. The best single recipe (iteration 10, recipe 1) achieved a utility of 0.975, with the closest match on Sour (6.5, exact target) and the largest residual gap on Dairy (5.0 vs. target 7).

Utility progression. Figure 30 reports the maximum utility per iteration and the cumulative best-so-far. The utility rose steadily from 0.776 at baseline, crossing 0.900 at iteration 4 and 0.940 at iteration 5. The best-so-far reached 0.957 at iteration 7 and continued climbing to the campaign peak of 0.975 at iteration 10, indicating that the optimizer had not fully plateaued and additional iterations could yield further gains.

Final formulation. Table 7 reports the per-attribute errors, and Table 6 the ingredient composition. The first two ingredients form the curd-forming substream, which is centrifuged to remove whey; the remaining seven form the cooked dressing, combined with the rinsed curds.

The cottage cheese was prepared in seven steps: **(1) Pasteurization** — soy milk batch-pasteurized at 90 °C for 5 min (100% cover closure). **(2) Acid coagulation** — combined with distilled vinegar at 200 RPM, 75 °C, for 2 min (80% cover

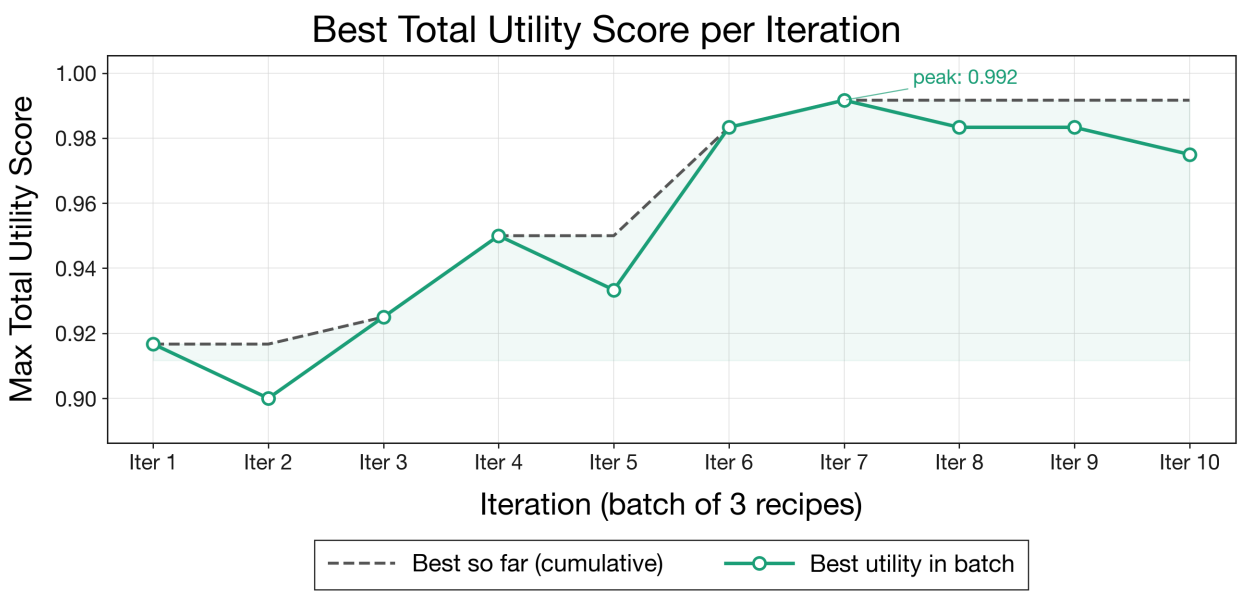

Figure 17. Best total utility score per iteration (solid) and cumulative best-so-far (dashed) across the 10 EGBO iterations. Utility rises from $f(x^{(0)}) = 0.767$ at baseline to a peak of 0.992 at iteration 7, with the best-so-far remaining at or above 0.983 from iteration 6 onward.

Table 7. Plant-based cottage cheese real-world validation. The best recipe (iteration 10, recipe 1) shows the largest gains on Off-Notes and Sour, with Dairy remaining the hardest attribute to reach. Overall utility improved 26% relative to baseline.

Attribute	Target	Baseline	Best	Error	
				Before	After
Homogeneity	6	4	6	2	0
Dispersion Phase	6	8	6	2	0
Slipperiness	6	9	6.5	3	0.5
Sour	6.5	10	6.5	3.5	0
Dairy	7	3	5	4	2
Off-Notes	0	7	1.5	7	1.5
Utility $f(x)$	1.000	0.776	0.975	0.224	0.025

closure). **(3) Curd separation** — basket centrifugation at 800 RPM (50 Hz), 25 °C, for 3 min, single pass; soy whey discarded. **(4) Curd rinsing** — washing machine, tap water, 10 °C for 1 min. **(5) Dressing cooking** — coconut cream, tap water, lemon juice, salt, nutritional yeast, tapioca starch, xanthan gum boiled in covered pot (90% cover) at 90 °C for 8 min. **(6) Combination** — rinsed curds + cooked dressing in planetary mixer at 60 RPM, 20 °C, for 2 min. **(7) Chilling** — cooled to 4 °C.

G.6. Cross-campaign comparison

Figure 31 overlays the best-so-far utility curves for both campaigns. Both start from similar baselines generated by the FSA (0.767 for yogurt, 0.776 for cottage cheese) and follow a characteristic steep-then-plateau trajectory. Yogurt converges faster, reaching 0.983 by iteration 6 and peaking at 0.992 at iteration 7, while cottage cheese reaches 0.957 by iteration 7 and peaks at 0.975 at iteration 10. The slower convergence of cottage cheese is consistent with its higher-dimensional sensory objective ($m = 6$ vs. $m = 4$) and larger active ingredient set ($d = 9$ vs. $d = 7$), both of which increase the effective optimization difficulty. Nevertheless, both campaigns achieved substantial improvements — 29% and 26% respectively — within the same 10-iteration budget, demonstrating that EGBO generalizes across product categories with different sensory structures and ingredient spaces, considering also that it started from a > 0.75 utility.

G.7. EGBO vs. Human Food Scientist

G.7.1. Protocol

The EGBO vs. HFS comparison (Section 5.3) was conducted under the following instructions and constraints, designed to ensure a controlled comparison while respecting the inherently different workflows of algorithmic and manual formulation.

HFS profile. HFS is a professional food scientist and chef with prior experience in recipe formulation but no prior experience specifically with plant-based dairy products. The operator was not given access to the EGBO system, its suggestions, or its results at any point during the experiment. HFS participated voluntarily as a professional collaborator; the comparison was designed in mutual agreement and HFS was not a research subject.

Shared constraints.

1. **Sensory targets.** HFS was given the same four sensory objectives used in the EGBO campaign, assessed on a 0 to 15 scale: Consistency (target 12), Creaminess (target 5), Tanginess (target 5), and Uniformity (target 13).
2. **Utility metric.** The scalar utility score (Equation 7) was computed for each evaluated formulation and reported to the operator after each evaluation.
3. **Plant-based constraint.** All ingredients must be plant-based; no animal-derived ingredients were permitted.
4. **Time tracking.** HFS documented the time spent on each trial or iteration, including preparation, cooling, tasting, and decision-making time.
5. **Documentation.** All ingredient quantities, preparation steps, and sensory observations were recorded for each trial.

Experiment 1: full workflow comparison.

1. **Objective.** Develop, from scratch, a plant-based yogurt recipe that matches the target sensory profile as closely as possible.
2. **No starting recipe.** The operator was free to choose any plant-based ingredients and any preparation method.
3. **No budget constraint.** There was no limit on the number of trials; the operator was instructed to continue until satisfied with the recipe or until further progress seemed unlikely.
4. **Trial definition.** A trial is defined as a complete attempt starting from raw ingredients. If the operator modifies a formulation mid-preparation, this is considered part of the same trial. If the operator stops and starts over from the beginning, a new trial begins. Trials abandoned before producing a tasteable result were recorded as failed trials.
5. **Evaluation.** Formulations that the operator deemed sufficiently promising were evaluated for sensory scores on the four target attributes and the corresponding utility was computed and shared with the operator.
6. **Workflow.** The scientist followed a conventional iterative workflow: prepare a candidate formulation, allow it to cool, taste, evaluate, decide on the next modification or restart, and repeat.

Experiment 2: same-start refinement comparison.

1. **Objective.** Optimize the FSA's starting recipe ($f(x^{(0)}) = 0.767$) toward the same four sensory targets.
2. **Fixed starting point.** The operator was given the exact recipe (ingredient list and concentrations) produced by the FSA, the same starting recipe used by EGBO.
3. **Budget.** Maximum of 10 iterations, matching the EGBO budget.
4. **One recipe per iteration.** The operator produces exactly 1 recipe per iteration; designing a batch of 3 independent variants in parallel was judged infeasible because each formulation decision depends sequentially on the sensory evaluation of the previous one.

5. **Sensory feedback.** After each iteration, the formulation was evaluated by the sensory panel on the four target attributes, and the scores and resulting utility were reported to the operator before the next iteration began.

6. **Permitted modifications.** The operator was free to adjust any ingredient concentration, add new plant-based ingredients, or remove ingredients, subject to the plant-based constraint.

7. **Time per iteration.** Each iteration consisted of recipe design, preparation, cooling, and sensory evaluation, requiring approximately 4 hours.

Key differences between EGBO and HFS workflows. Table 8 summarizes the structural differences between the two workflows.

Table 8. Structural differences between EGBO and HFS workflows in Experiment 2.

	EGBO	HFS
Recipes per iteration	3 (batch)	1 (sequential)
Total recipes in 10 iter	30	10
Recipe design time	Seconds	Hours
Decision basis	GP surrogate	Expert judgment
Regression possible	Unlikely	Observed

The batch vs. sequential distinction is not an artificial handicap imposed on HFS but a genuine consequence of the two workflows. EGBO's acquisition function can propose multiple diverse points in a single optimization step because it reasons over a global surrogate model. HFS, by contrast, reasons sequentially: each modification is informed by the sensory outcome of the previous one, and designing three independent variants in parallel would require the operator to mentally maintain three separate hypotheses about ingredient interactions without feedback.

G.7.2. Results

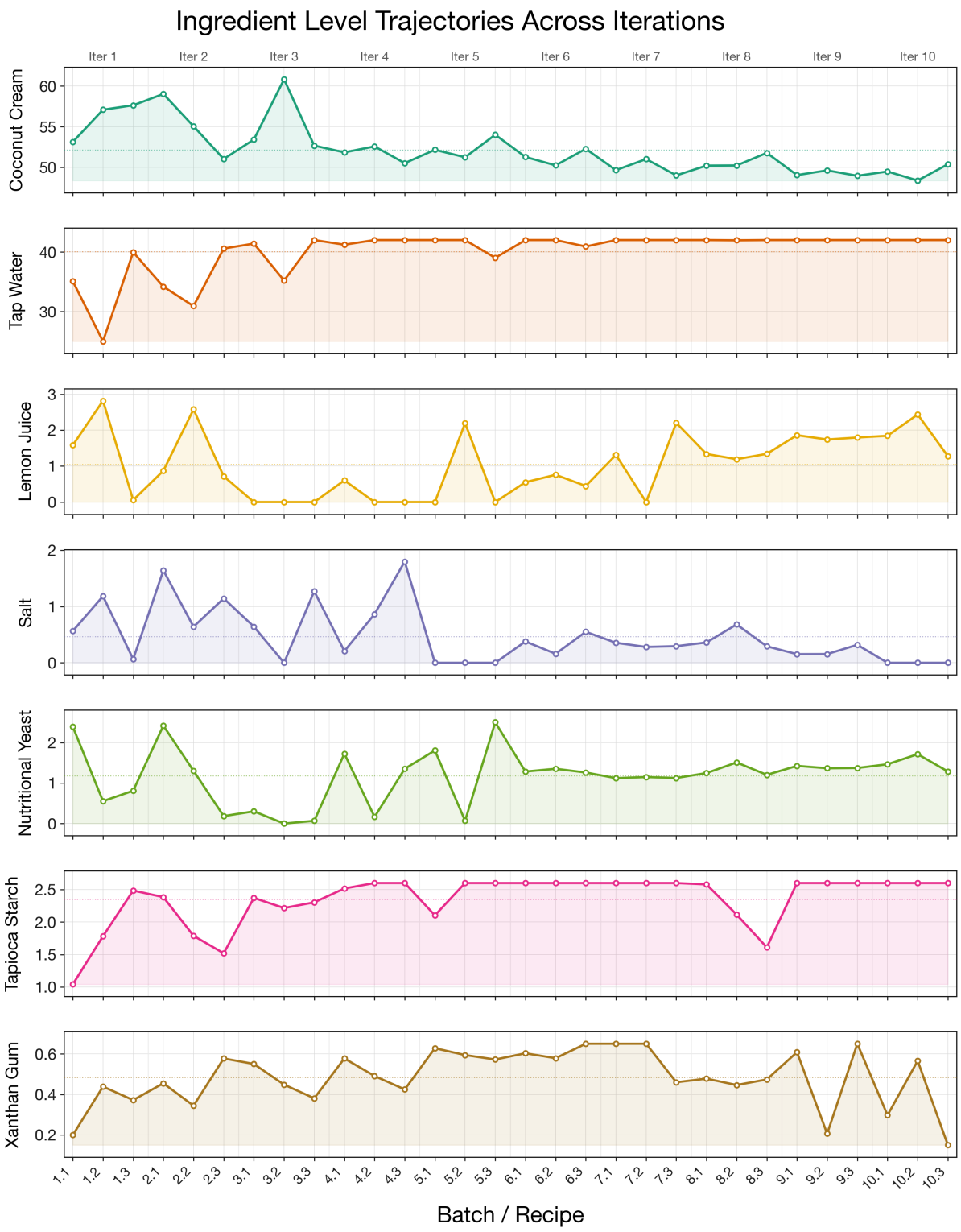

Figure 18. Ingredient concentration trajectories across the 10 EGBO iterations for the seven varying formulation dimensions. Each panel is independently scaled to reveal fine-grained adjustments: the dominant Coconut cream and Tap water rebalancing is visible alongside the smaller but consequential shifts in Lemon juice, Salt, Nutritional yeast, Tapioca starch, and Xanthan gum.

Iteration 1

Figure 19. Plant-based yogurt: iteration 1.

Iteration 2

Figure 20. Plant-based yogurt: iteration 2.

Iteration 3

Figure 21. Plant-based yogurt: iteration 3.

Iteration 4

Figure 22. Plant-based yogurt: iteration 4.

Iteration 5

Figure 23. Plant-based yogurt: iteration 5.

Iteration 6

Figure 24. Plant-based yogurt: iteration 6.

Iteration 7

Figure 25. Plant-based yogurt: iteration 7.

Iteration 8

Figure 26. Plant-based yogurt: iteration 8.

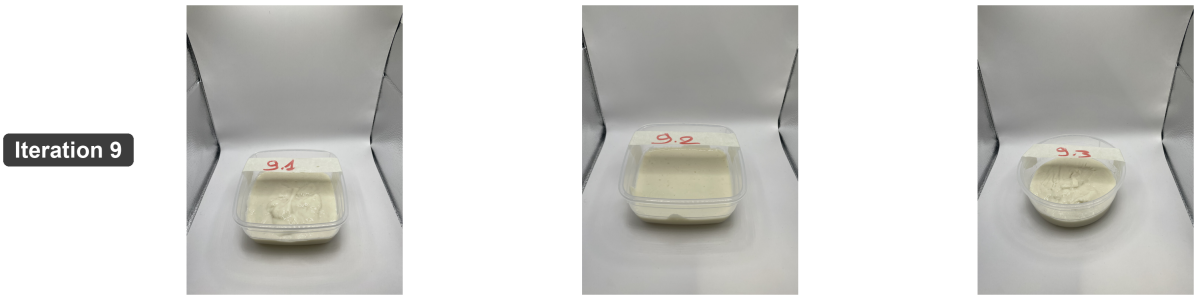

Figure 27. Plant-based yogurt: iteration 9.

Figure 28. Plant-based yogurt: iteration 10.

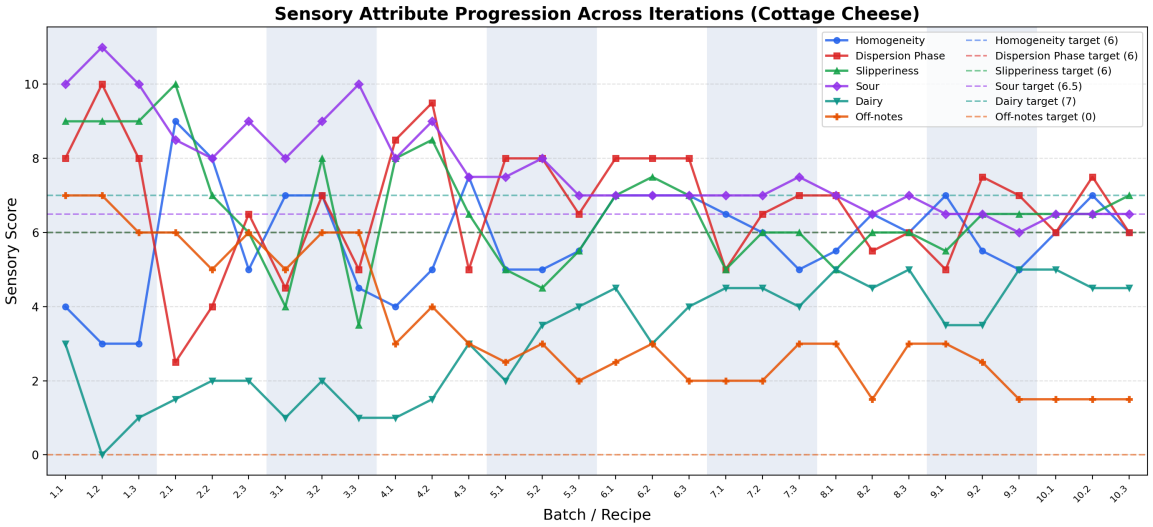

Figure 29. Sensory attribute trajectories across the 10 EGBO iterations (3 recipes per iteration, 30 total) for the plant-based cottage cheese campaign. Solid lines show observed panel scores; dashed lines indicate the target value for each attribute.

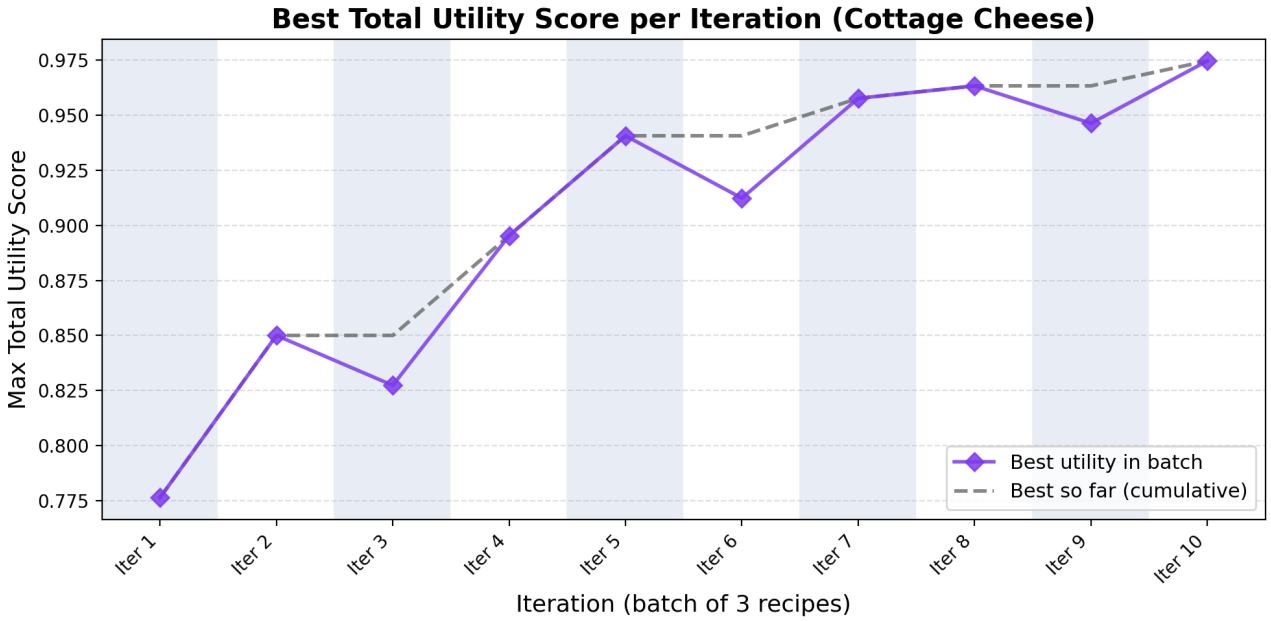

Figure 30. Best total utility score per iteration (solid) and cumulative best-so-far (dashed) across the 10 EGBO iterations for the cottage cheese campaign. Utility rises from $f(x^{(0)}) = 0.776$ at baseline to a peak of 0.975 at iteration 10.

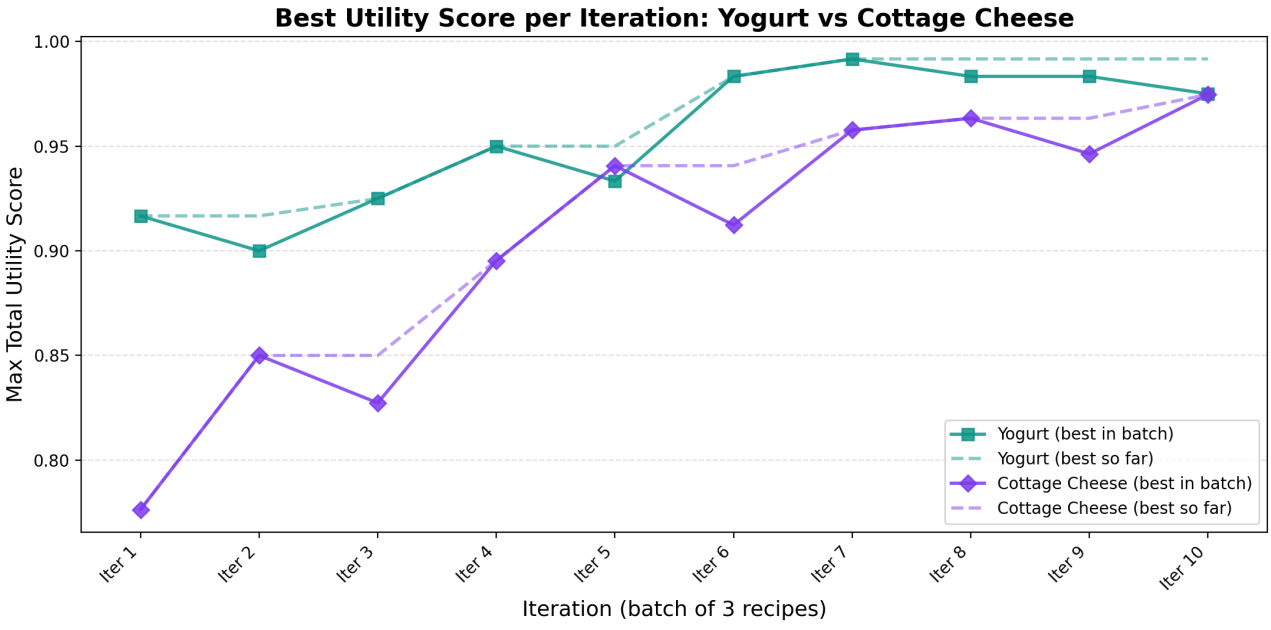

Figure 31. Comparison of best-so-far utility across both real-world campaigns. Yogurt converges faster (peaking at 0.992 by iteration 7), consistent with its lower-dimensional sensory objective (m=4, d=7). Cottage cheese converges more slowly (peaking at 0.975 at iteration 10), reflecting its higher-dimensional objective (m=6, d=9).

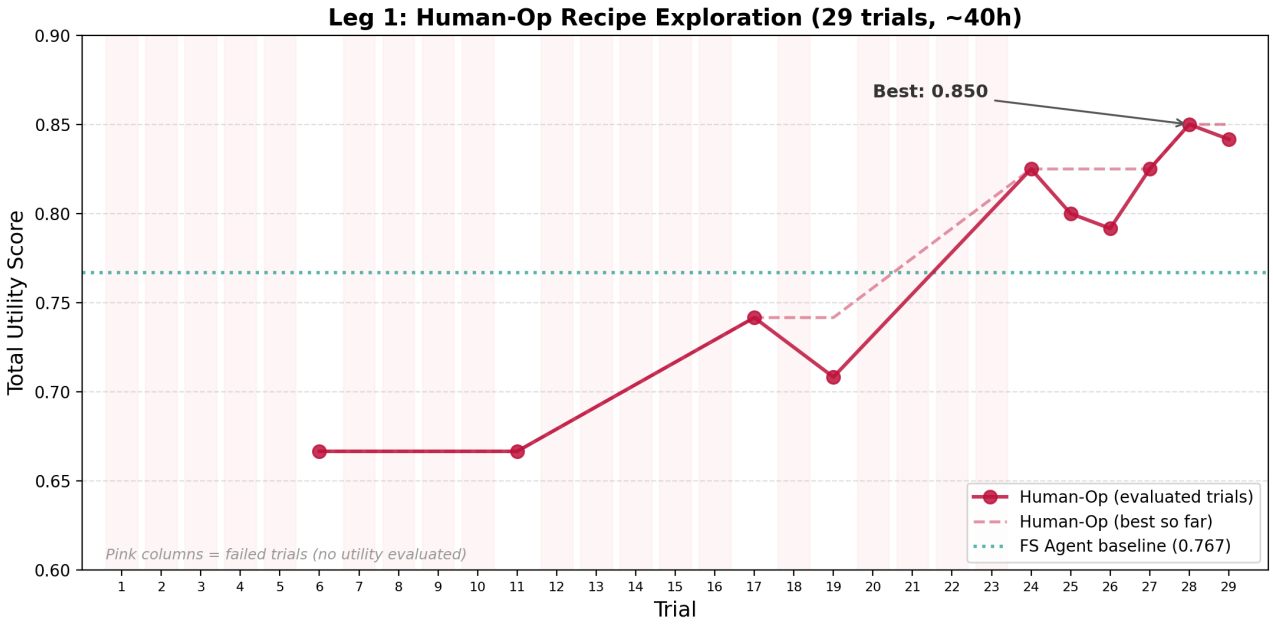

Figure 32. Experiment 1: HFS recipe exploration. The operator executed 29 trials over ∼40 h, of which 10 produced formulations evaluated for utility. The dotted line indicates the utility of the FSA's starting recipe (0.767), produced in negligible time. HFS's best result (0.850, trial 28) required the full 40-hour exploration budget.

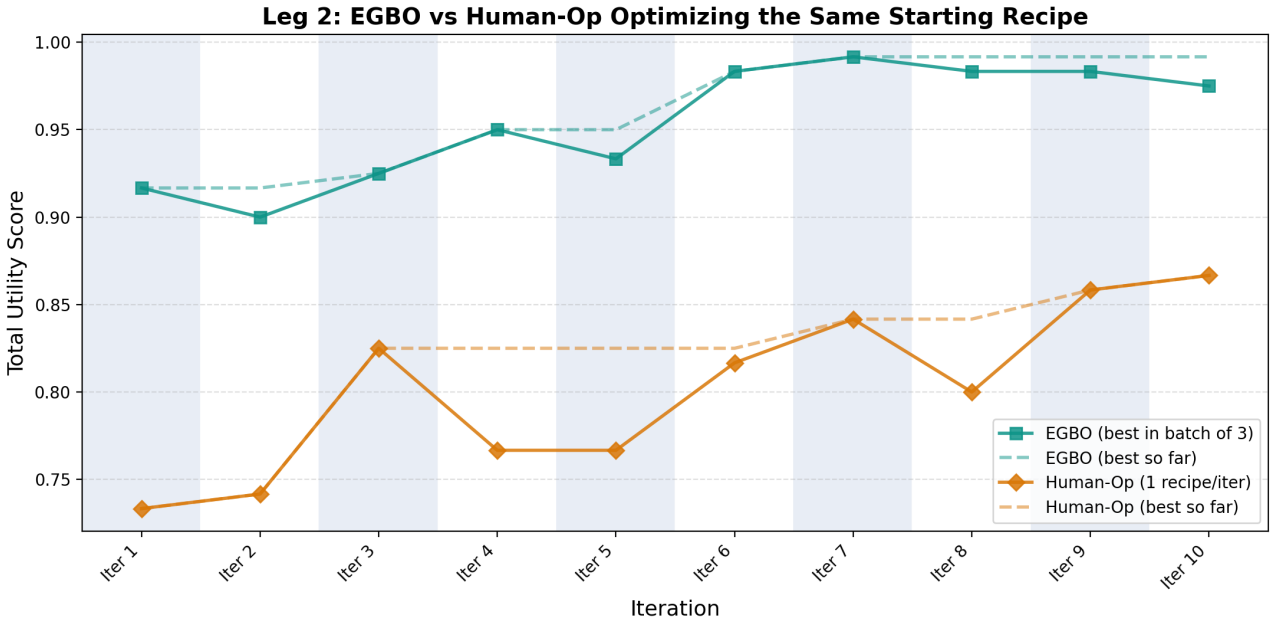

Figure 33. Experiment 2: EGBO vs. HFS, both optimizing the same starting recipe ($f(x^{(0)}) = 0.767$) over 10 iterations. EGBO proposes 3 recipes per iteration; HFS produces 1. EGBO reaches 0.992 by iteration 7, while HFS reaches 0.867 by iteration 10, a 14.4% gap.