Practical Bayesian Optimization for Scientific Discovery

Hamza T. Chaudhry [*1,2] **Sean H. Murphy** [*2,3] **Umesh Padia** [*2,3] **Cengiz Pehlevan** [1,4] **James Harrison** [5] **George Church** [2,6] **Jasper Snoek** [5]

Abstract

Bayesian optimization (BO) is a standard tool for experimental scientific discovery, where evaluations are costly and candidate spaces are vast. Classical formulations often rely on methods that scale poorly with data size and are ill-suited to discrete sequences and molecules. Scientific foundation models now provide rich, transferable representations for these domains. However, it remains unclear how to best leverage them with BO in lab-in-the-loop campaigns, or how this approach compares with LLMs and generative modeling, two leading paradigms in AI-for-Science. We investigate these questions across proteins, DNA, RNA, and small molecules through 61 regression tasks drawn from established experimental benchmarks. To our knowledge, this constitutes the most extensive cross domain study of foundation-model driven Bayesian optimization to date. We systematically ablate foundation models, surrogates, acquisition functions, and fine-tuning regimes under both sequential and batched selection. We find that Gaussian process surrogates are consistently outperformed by neural alternatives paired with Thompson sampling, particularly MLP ensembles and variational Bayesian last layers. We further find that smaller batch sizes reach peak performance faster and recover more elite candidates under the same total experimental budget, a trend that is consistent across tasks and surrogate choices. Finally, we show that foundation-model driven BO outperforms both an in-context LLM surrogate and a guided discrete diffusion model.

Accepted at ICML 2026 Workshop on Decision-Making from Offline Datasets to Online Adaptation: Black-Box Optimization to Reinforcement Learning. [*]Equal contribution [1]Harvard SEAS, Cambridge, MA, USA [2]Wyss Institute, Boston, MA, USA [3]MIT EECS, Cambridge, MA, USA [4]Kempner Institute, Cambridge, MA, USA [5]Google DeepMind [6]Harvard Medical School, Boston, MA, USA. Correspondence to: Hamza T. Chaudhry <hchaudhry@g.harvard.edu>, Sean H. Murphy <seanmurphy@csail.mit.edu>, Umesh Padia <umesh_padia@hms.harvard.edu>.

Preprint. June 30, 2026.

1. Introduction

In experimental discovery campaigns, candidates are cheap to propose and often expensive to measure (Tom et al., 2024; Shields et al., 2021). Protein engineering, regulatory genomics, RNA design, and molecular optimization therefore share a common decision problem: given prior measurements and a limited budget, which candidates should be tested next? Bayesian optimization (BO) addresses expensive black-box optimization in settings like these by explicitly balancing exploration of uncertain regions against exploitation of regions predicted to perform well. This balance is typically mediated by a probabilistic surrogate model whose posterior uncertainty quantifies the potential value of each candidate evaluation (Srinivas et al., 2010; Russo & Van Roy, 2014). Gaussian process (GP) surrogates remain a principled default because they provide closed-form posterior predictions and support provable regret guarantees (Rasmussen & Williams, 2006).

Scientific discovery, however, has characteristics that distinguish it from the continuous low-dimensional benchmarks on which classical BO is often applied. Proteins, RNA, DNA, and small molecules each require their own specialized representations, and a hand-designed kernel does not transfer between them (Romero et al., 2013; González-Duque et al., 2024). The GP also scales cubically in the number of observations and requires elaborate batch approximations (Desautels et al., 2014; González et al., 2016; Wilson et al., 2017), which becomes prohibitive at the batch sizes typical of experimental campaigns. Coupling FM representations with BO is appealing, but existing approaches are fragmented. For instance, prior work has used FM embeddings as kernel features for a GP (Kristiadi et al., 2024; Ranković et al., 2025), as inputs to a Random Forest (Jiang et al., 2024), or as in-context tokens for an LLM-driven acquisition step (Liu et al., 2024), with limited head-to-head comparison of which choices matter in practice.

In the broader literature on AI-for-Science, two alternative paradigms are the subject of active investigation. The first treats LLMs as in-context surrogates, prompting the model with the history of measurements and asking it to score the remaining candidate pool directly (Liu et al., 2024; Chang et al., 2025; Ramos et al., 2026). This avoids explicit sur-

rogate training, but it asks the acquisition step to act on outputs that are not calibrated posterior uncertainties. Several lines of evidence reinforce this concern: miscalibration introduced by instruction tuning and RLHF (Xiong et al., 2024; Tian et al., 2023; Kalai et al., 2025), insensitivity to experimental feedback in genetic perturbation and molecular property tasks (Gupta et al., 2025), and weak in-context performance on continuous-valued regression (Nafar et al., 2025). The second treats experimental design as conditional generation, training or steering a generative model so that its samples are themselves the proposed candidates (Stanton et al., 2022; Yang et al., 2025; Wang et al., 2024; Rector-Brooks et al., 2026; Hartman et al., 2025). However, this is complicated by the fact that guidance toward high-reward regions can collapse sample diversity (Barceló et al., 2024) and push samples off the data manifold (Klarner et al., 2024), and when the steering signal is a learned reward, optimization pressure can exploit reward-model errors rather than producing genuinely high-fitness candidates (Yoshizawa et al., 2025).

In this study, we aim to clarify how foundation models should be used within Bayesian optimization for scientific discovery. We study frozen and adapted foundation-model embeddings as task representations, sample-efficient surrogate heads as uncertainty-aware predictors, and batch-aware acquisition as the mechanism for selecting experiments under realistic laboratory scenarios. This framing allows us to ask which components of foundation-model-augmented BO matter most, how these choices interact under experimentally relevant constraints, and how the resulting approach compares with the leading paradigms of in-context LLM surrogates and steered generative design. We make the following contributions:

1. *A practical Bayesian optimization framework* (SciOpt) that unifies proteins, RNA, DNA, and small molecules in a single modular pipeline of frozen foundation-model embeddings and a sample-efficient surrogate head, with backbone, head, acquisition, and fine-tuning regime (frozen, supervised fine-tuning, self-supervised fine-tuning) as independent design axes.
2. *A systematic empirical study* on 61 real-data regression tasks across protein, chemistry, RNA, DNA benchmarks, ablating nine surrogates, two FM backbones per modality, three acquisition functions, three fine-tuning regimes, and across five batch sizes. We find that VBLL with Thompson sampling is the unique upper-right Pareto point of optimal performance and coverage on chemistry, RNA, and DNA, while MLP Ensemble leads on protein.
3. *Head-to-head comparison against two leading AI-for-Science paradigms.* SciOpt beats an in-context LLM surrogate (Qwen3.6-35B) on Top-5% coverage for a representative dataset in every modality, and beats a steered masked-diffusion generative model (SGPO) on the combinatorially complete GB1, TrpB, and PhoQ protein datasets.

2. Technical Overview

SciOpt (Figure 1) functions as a closed-loop selector over a finite candidate pool. A scientific foundation model maps each candidate to an embedding z; a surrogate trained on accumulated measurements $\mathcal{D}_t$ predicts a mean and uncertainty; and an acquisition rule selects the next candidate or batch for experimental evaluation. The new measurements are added to the labeled set, the surrogate is retrained, and the loop repeats.

We organize this loop around four design axes: the foundation-model backbone, the surrogate head, the acquisition rule, and the fine-tuning regime. Each axis is exposed as a separate component in the configuration, so any compatible $(\text{FM}, \text{head}, \text{acquisition}, \text{regime})$ tuple can be run without changing the rest of the pipeline. This factorization lets us compare one choice at a time while holding the other axes fixed.

Foundation-model backbones. For the main sweep, we use one primary FM per modality: ESM-C 600M (EvolutionaryScale Team, 2024) for proteins, RNA-FM (Chen et al., 2022) for RNA, NTv3-650M (Boshar et al., 2025) for DNA, and ChemFM-1B (Cai et al., 2026) for chemistry. In the frozen setting, the FM block in Figure 1 acts only as a feature extractor: embeddings are precomputed once per (FM, task) pair, and the BO loop trains only the surrogate head. Alternative backbones, pooling choices, and pseudo-likelihood calculations for the zero-shot baseline are described in Appendix A and Appendix E.

Surrogate heads. The surrogate heads, which consume the FM representations to predict the property of interest (e.g., binding affinity, promoter expression), fall into three groups. *Non-adaptive baselines* do not learn from experimental feedback: `Random` samples uniformly and `Zero_Shot` ranks candidates by the FM prior alone. *Greedy methods* train a deterministic head on the labeled set each round and select by predicted mean, with no per-candidate uncertainty: this group includes the random-forest EvolvePro configuration (Jiang et al., 2024), a linear head, and an MLP. *Uncertainty-aware methods* expose a per-candidate uncertainty estimate to the acquisition step and we include classical algorithms: a random forest, an exact GP, an MLP ensemble. We also include a variational Bayesian last layer, which has been shown to be competitive with exact GPs (Harrison et al., 2024; Brunzema et al., 2025). Implementation details and hyperparameters are in Appendix B.

Acquisition functions. Three acquisition rules cover the methods in Table 1. *Greedy* selects the candidate with the largest predicted value, or the largest pseudo-likelihood

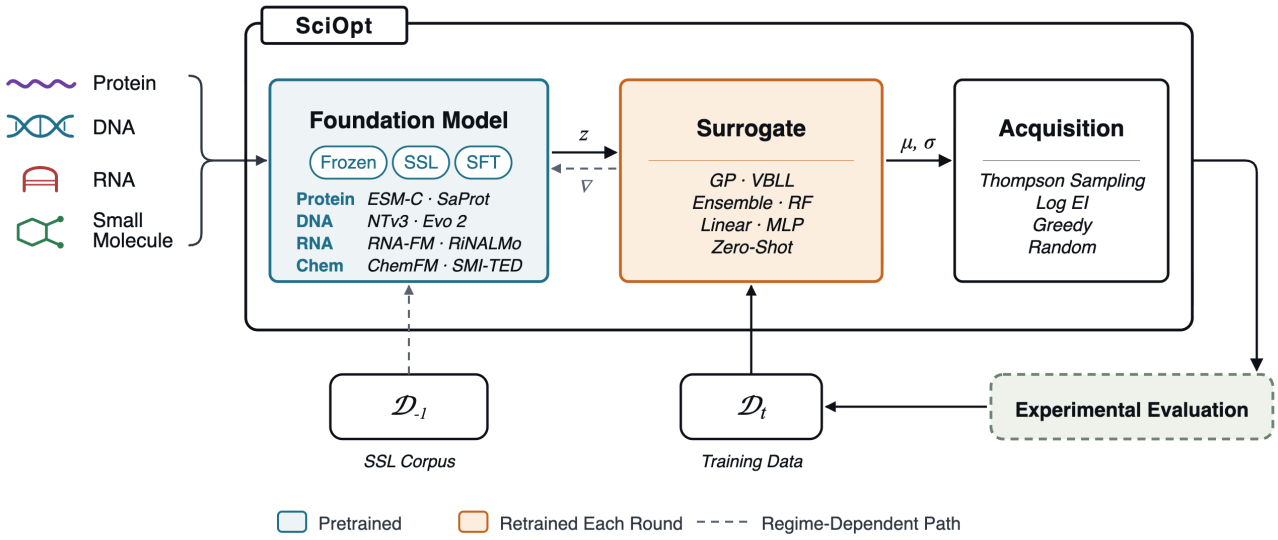

Figure 1. **SciOpt.** A modular Bayesian optimization framework that couples a frozen scientific foundation model to a learned surrogate head and a batch-aware acquisition. Each node lists the choices studied across four modalities (protein, DNA, RNA, chemistry). Solid arrows indicate the BO loop, in which the surrogate is retrained on accumulated data D_t and proposes the next batch. Dashed arrows denote optional configurations such as self-supervised pretraining on a domain corpus D_1 and supervised fine-tuning via surrogate gradients (∇).

for `Zero_Shot`. *LogEI* scores candidates by expected improvement using the surrogate's predictive uncertainty (Ament et al., 2023). *Thompson sampling* samples a possible score function and selects its maximizer over the heldout pool (Russo & Van Roy, 2014). For batch construction at size k, non-Thompson methods take the top-k candidates under the round-start acquisition score. Thompson methods draw one posterior sample per batch member; the additional VBLL diversity rule is given in Appendix C.3.

Fine-tuning regimes. The regime-dependent paths in Figure 1 specify which FM parameters, if any, are updated during the BO loop. In the *Frozen* regime, embeddings are cached and only the surrogate head is trained. This is the regime used for the full 61-task surrogate sweep, as it is the most computationally tractable. The *Supervised Fine-Tuning* regime lets the FM adapt to measured labels through LoRA adapters (Hu et al., 2022); at each round, the adapters and head are reinitialized and then trained on the current labeled set. The *Self-Supervised Fine-Tuning* regime continues pretraining the FM prior to running experiments, by using the pretraining objective on related data. This is applied in protein tasks as shown in ESM-1v and ProteinTTT (Meier et al., 2021; Bushuiev et al., 2026), where there is a wild-type sequence you wish to mutate and you can enhance your FM embeddings by continuing pretraining with a masked-language-model objective on related protein sequences according to multi-sequence alignment (MSA). Details and results for the different fine-tuning regimes are in Appendix F.

3. Results

Problem setup. Let $\mathcal{X}$ be a finite candidate pool of size N and $f : \mathcal{X} \to \mathbb{R}$ an unknown black-box objective. At round t the learner has noisy measurements $\mathcal{D}_{t-1} = \{(x_s, y_s)\}_{s<t}$. Sequential BO selects one candidate per round while batch BO selects k candidates without intermediate observations. Each campaign is initialized with a round 0 of 96 candidates drawn uniformly from below the median of the marginal label distribution, mimicking a low-fitness starting library, and runs for $T = 100$ rounds in the sequential setting or batch rounds of $T \in \{32, 16, 8, 4, 2\}$ at batch sizes $k \in \{24, 48, 96, 192, 384\}$ respectively for a total of $N = 768$ candidates each. Every (task, method) tuple is run with 5 BO seeds, with round 0 candidates shared within a (task, seed) pair for fair comparison between methods.

Tasks and metrics. Our study contains 61 real-data scientific regression tasks across proteins (n_P=18), chemistry (n_C=12), RNA (n_R=20), and DNA (n_D=11). Tasks are included when they have a deduplicated candidate pool of at least 2,000 sequences or molecules and fit within the primary FM's context window. Large datasets are deterministically subsampled when needed to $10,000$ candidates, with the same runtime pool shared across all surrogate baselines and BO seeds, so aggregate metrics are more comparable. Per-task data sources, splits, and preprocessing are in Appendix D.

Table 1. **Summary of methods compared in the main results.** All methods consume the same per-task FM embeddings. The table lists the surrogate/acquisition pair used in the main sweep; implementation details and hyperparameters are in Appendix B.

Method	Acquisition	Role in comparison
Non-adaptive baselines		
Random	Random	Uniform random sampling without feedback.
Zero-Shot	Likelihood	Ranks candidates by the FM's pseudo-likelihood without fitting to labels.
Greedy heads		
Random Forest	Greedy	EvolvePro-style random forest on frozen FM embeddings (Jiang et al., 2024).
Linear	Greedy	Linear probe on FM embeddings; tests whether a shallow head is sufficient.
MLP	Greedy	Nonlinear deterministic head trained on labeled data each round.
Uncertainty-aware heads		
Random Forest	LogEI	Adds exploration to Random Forest using variance across trees.
Gaussian Process	LogEI	Classical Bayesian surrogate baseline on FM embeddings (Rasmussen & Williams, 2006).
MLP Ensemble	Thompson	Uses variation across independently trained MLP heads as a function posterior (Lakshminarayanan et al., 2017; Riquelme et al., 2018).
VBLL	Thompson	Variational Bayesian last-layer surrogate with analytic posterior sampling (Harrison et al., 2024).

We track two primary metrics:

$$\begin{aligned} \mathrm{NBSF}(t) &= \max_{s \le t} \left(\frac{y_s - y_{\min}}{y_{\max} - y_{\min}} \right), \\ \text{Top-5\%}(t) &= \sum_{s \le t} \mathbf{1}[\, y_s \ge y_{0.95} \,] \end{aligned}$$

where $y_{\min}$, $y_{\max}$, and $y_{0.95}$ denote the minimum, maximum, and 95th percentile of the labels over the full candidate pool $\mathcal{X}$. Normalized-Best-So-Far (NBSF) measures how close the campaign has come to the global maximum. Top-5% Count measures how many distinct elite candidates have been recovered, so it captures coverage rather than only peak performance. For uncertainty-aware heads, we also compute cumulative expected calibration error (ECE) on heldout candidates to evaluate whether predictive uncertainty is calibrated over the course of the campaign.

3.1. Sequential Bayesian Optimization

Sequential BO is the cleanest setting for comparing surrogate and acquisition choices. After each selected candidate is measured, the surrogate is retrained on the updated labeled set before the next decision. This removes batch-construction effects and tests how well each method turns frozen FM embeddings into one-step acquisition scores.

Figure 2 shows the final NBSF and Top-5% count by modality. The upper frontier is dominated by neural heads. `VBLL` is the strongest overall uncertainty-aware method, sitting at or near the upper-right frontier in every modality and leading clearly on chemistry and RNA. Protein is the main exception on coverage: `MLP_Ensemble` recovers the most Top-5% candidates, with `MLP_Greedy` also competitive and `VBLL` trailing on that axis despite comparable NBSF. DNA shows a compact upper tier with `VBLL`, `MLP_Greedy`, `MLP_Ensemble`, and `Linear` all close within error bars.

The middle tier is formed mostly by `Linear`, `GP`, and the random-forest variants, although the exact ordering varies by modality. `Linear` and `GP` are often close, which suggests that much of the predictive signal is already present in the FM embedding. The exact GP remains a useful calibrated baseline, but it does not translate that calibration into the best optimization performance. `RF_Greedy`, the EvolvePro-style recipe, generally falls below the neural heads; adding LogEI to the same random-forest posterior generally boosts performance in most settings but does not move it to the frontier. `Zero_Shot` improves over Random but remains well below learned surrogates, showing that the FM prior alone is not enough for reliable adaptive optimization.

The strong performance of `MLP_Greedy` is worth noting. Although its acquisition rule is greedy, the head is reinitialized and retrained each round, so stochastic optimization can introduce a form of implicit exploration. This can be viewed intuitively as a form of implicit Thompson sampling: the trained weights at each acquisition step are approximately a draw from an implicit posterior over MLP weights induced by the initial distribution plus the optimizer trajectory on a finite dataset (Maclaurin et al., 2015). We therefore view `MLP_Greedy` as a strong simple baseline rather than as a purely exploitative control. Still, the most consistent frontier methods are the ones with explicit uncertainty or sampling, especially `VBLL` and `MLP_Ensemble`. Calibration results are given in Appendix G.2.

We evaluate fine-tuning on selected representative protein tasks, as it is computationally intractable to do across all datasets. Supervised LoRA fine-tuning can improve sequential performance for some heads, while MSA-based self-supervised fine-tuning has mixed, task- and head-dependent

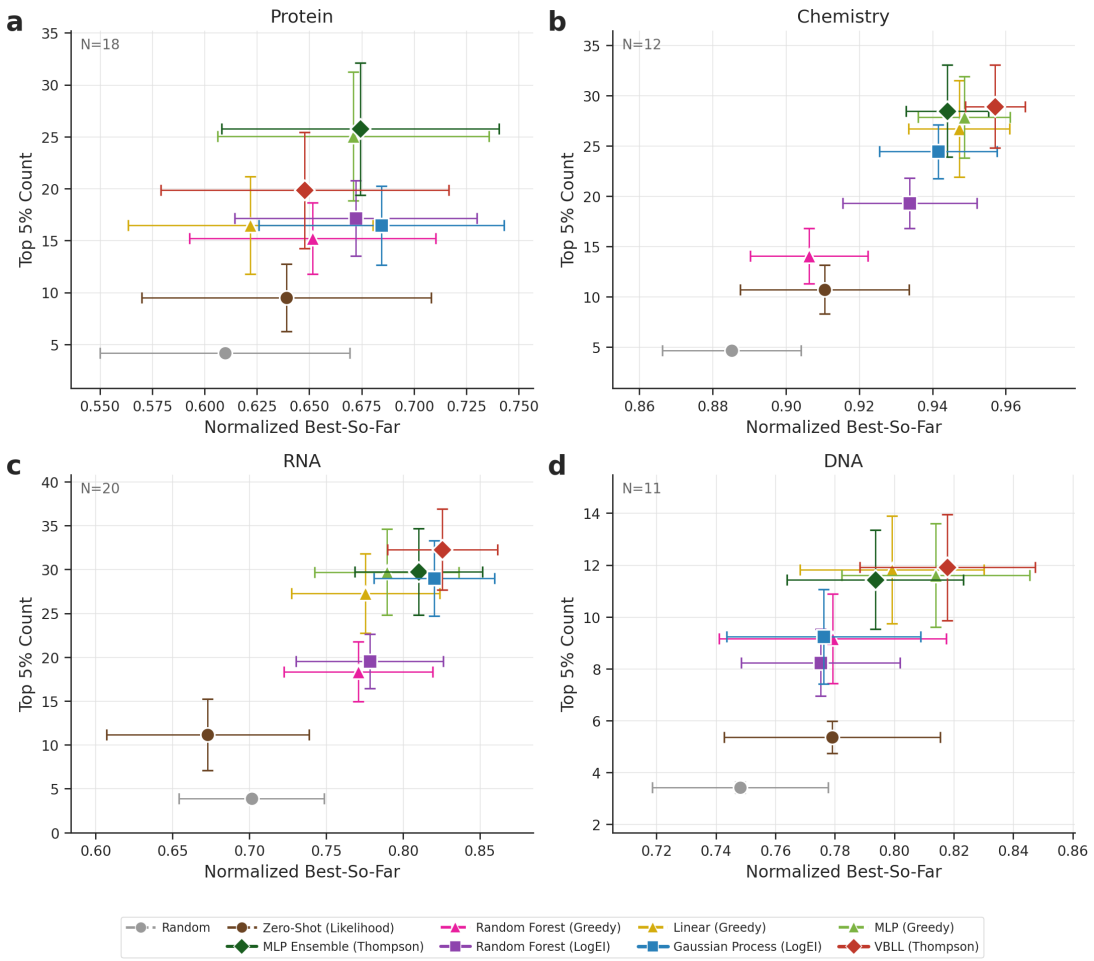

Figure 2. **Per-modality Pareto frontier of NBSF against Top-5% count, sequential BO (T=100, k=1).** Each marker is one method aggregated over tasks and seeds; error bars are ±1 SEM. `VBLL` (red diamond) is the unique upper-right frontier point on chemistry, RNA, and DNA. On protein tasks, `MLP_Ensemble` (dark green diamond) leads with `MLP_Greedy` (light green triangle) following close behind. The middle cluster of `Linear` (yellow triangle), the simple greedy baseline, and `GP` (blue square), the classical BO surrogate, are one tier below. `RF_Greedy` (pink triangle), also known as EvolvePro, is generally outperformed by the other methods, including `RF_LogEI` (purple square) which adds in some exploration to the search. `Zero_Shot` (brown circle) slightly outperforms `Random`, which performs the worst of all methods.

effects: aggregate NBSF changes are small, but Top-5% coverage can improve for some neural heads and degrade for others. Detailed fine-tuning results are given in Appendix F.

3.2. Batch Bayesian Optimization

Experimental campaigns rarely get to choose candidates one at a time. Protein engineering, MPRA screens, RNA assays, and small-molecule screens are usually organized around batches of candidates, each tested over a period of weeks to months. Running larger batches improves throughput, amortizes setup cost, and can be necessary when reagents, cell lines, instruments, or personnel are available only within a limited window. The statistical cost is that BO receives less feedback: the surrogate must choose all k candidates in a batch before observing any of their labels.

We quantify this tradeoff with a fixed total budget of candidates. After the shared 96-candidate initialization, we evaluate batch sizes $k \in \{24, 48, 96, 192, 384\}$ for $T \in \{32, 16, 8, 4, 2\}$ rounds, respectively, giving the same acquisition budget of 768 new candidates in every setting. Each surrogate uses its native batch rule, and curves are aggregated over all 61 tasks, 9 methods, and 5 seeds.

Figure 3 shows a consistent batch-size effect. Smaller batches reach high NBSF earlier because labels are incorporated into the surrogate more often. The final NBSF gap narrows by the end of the budget, which means peak best-so-far performance is relatively forgiving of moderate batching. Top-5% count behaves differently: smaller batches recover elite candidates at a higher rate throughout the campaign, and the gap remains visible even at the final budget. This

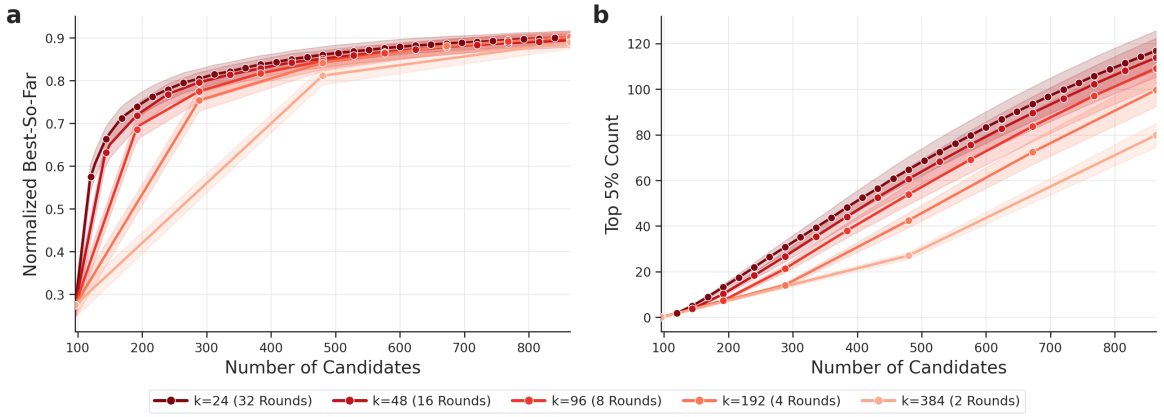

Figure 3. **Batch-size scaling at fixed budget.** NBSF (left) and Top-5% Count (right) vs. cumulative candidates measured across multiple batch sizes with a common total budget. Curves aggregate across 61 tasks, 9 surrogates, and 5 seeds. Bands are ±1 SEM over the 61 tasks and quite narrow, indicating that the batch-size effect is highly conserved across tasks and methods. Smaller batches dominate at low cumulative data and curves converge as the surrogate posterior tightens, albeit at a significantly slower rate for Top-5% Count.

suggests that coverage of the high-fitness set depends more strongly on the number of feedback rounds than on the total number of candidates measured. We include the full per-modality Pareto frontier of NBSF against Top-5% Count for the practical k=96 setting in Figure 13 in Appendix H.

3.3. Comparison to In-Context Learning with LLMs

Experimental setup. In this paradigm, we compare SciOpt to the use of LLMs as in-context surrogates for experimental design. In this setting, each BO round is a single forward pass over a prompt containing the running history of measured candidates and the remaining candidate pool; the model returns one candidate ID, with no fitted surrogate, no embedding head, and no explicit acquisition function (Liu et al., 2024; Chang et al., 2025; Ngo et al., 2026). We use Qwen3.6-35B-A3B-FP8 (Qwen Team, 2026) as the representative LLM baseline. Full prompt format, inference settings, parsing rules, and contamination controls are given in Appendix I.

We evaluate one representative task per modality from the 61-task benchmark. The full protein and DNA pools are too large to provide in context, so for `gb1` and `gosai_sknsh` we use deterministic subsamples of size 1,500 and 2,100, respectively. We rerun the SciOpt methods on these subsampled datasets for a fair comparison. This setup gives the LLM the full visible candidate pool whenever context permits, and uses identical starting data across methods.

Results on representative tasks from each modality. Figure 4 shows that the LLM baseline can find high-scoring candidates on some tasks, but it is much less reliable than SciOpt as an optimizer. On NBSF, the LLM trails the stronger SciOpt surrogate on every task, but the identity of that surrogate is task-dependent. The gap is modest on `pparg` (LLM 0.864, MLP 0.861, VBLL 0.956) and `gb1` (LLM 0.722, MLP 0.853, VBLL 0.582; VBLL's drop here likely reflects the smaller candidate pool), and widens substantially on `ke_mrna` (LLM 0.720 vs both surrogates $\geq$ 0.92, essentially at random 0.711) and on `gosai_sknsh` (LLM 0.558 vs MLP 0.946 and VBLL 0.742, below random 0.635).

The Top-5% count separates the methods more clearly. On the two matched-pool tasks, the LLM recovers fewer elite candidates than SciOpt: 19.8 versus 22.8 on `pparg`, and 10.6 versus 63.8 and 65.2 on `ke_mrna`. On `gb1` and `gosai_sknsh`, absolute Top-5% counts are affected by the smaller LLM-visible pools, so we treat those comparisons as conservative diagnostics rather than matched-pool estimates. The qualitative pattern is still consistent across tasks: the LLM occasionally identifies a good region, but it does not cover the elite set at the same rate as a learned surrogate.

Together, the two metrics suggest that the LLM is not failing simply because it cannot understand the task; reasoning traces (Appendix I) show coherent, domain-grounded selection rationales on every task. It often improves NBSF, especially when an early high-fitness mode is enough to make progress. The failure mode is coverage: the model tends to return candidates near regions it has already identified rather than spreading measurements across distinct high-fitness regions. This behavior is consistent with prior reports that instruction-tuned LLMs can behave like strong priors but poor calibrated surrogates under small experimental budgets (Rychert et al., 2025; Xiong et al., 2024; Tian et al., 2023). In this role, a learned surrogate provides a more reliable decision step than in-context selection alone, though the strongest surrogate is task-dependent.

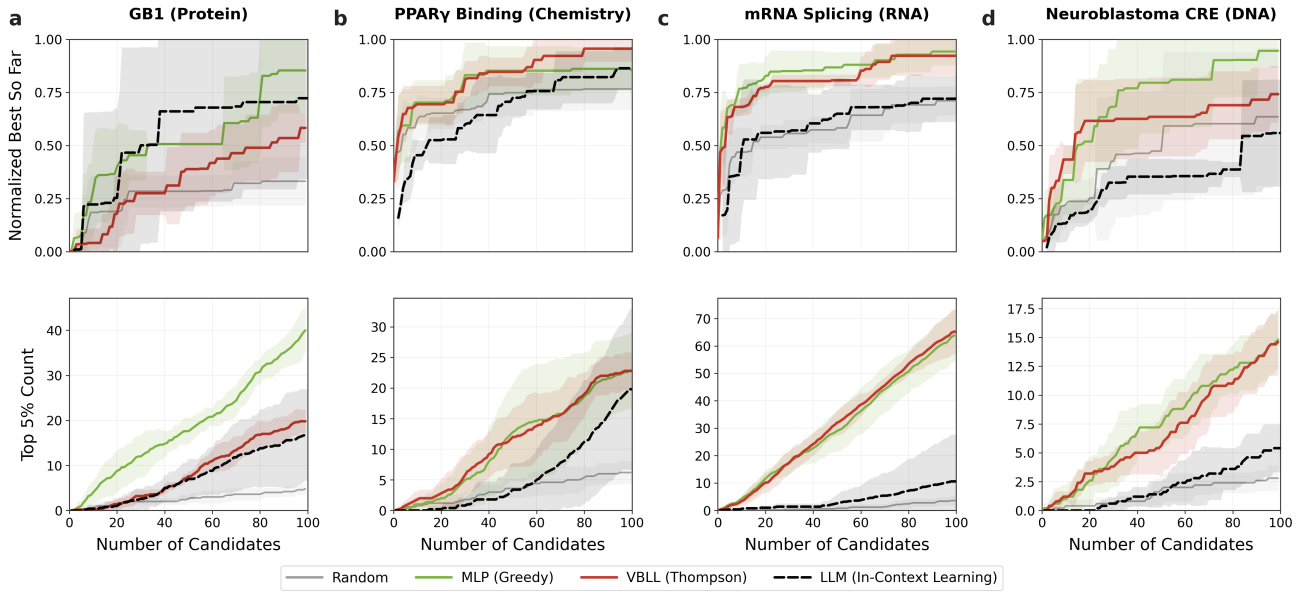

Figure 4. **SciOpt vs. LLM in-context selection.** Columns show one task per modality: `gb1` (protein), `pparg` (chemistry), `ke_mrna` (RNA), and `gosai_sknsh` (DNA). Top row: NBSF. Bottom row: Top-5% count. All methods run sequential selection for 100 rounds after a shared 96-candidate below-median initialization; shaded regions show ± 1 s.d. over 5 seeds. LLM (In-Context Learning) selects one candidate per round from a prompt containing the visible pool and the full measurement history. MLP and VBLL use the modality-default FM embeddings.

3.4. Comparison to Steered Generation

Experimental setup. In this paradigm, we elect to compare against Steered Generation for Protein Optimization (SGPO). The authors conduct a broad sweep over generative priors and steering strategies, benchmarked on protein fitness optimization, from which we select the strongest reported configuration (Yang et al., 2025). The results are summarized in Figure 5.

SGPO samples from a masked diffusion language model (MDLM), $p_\theta(x)$, trained on the target-family MSA, and uses decoupled annealing posterior sampling (DAPS) with an ensemble of one-hot MLP value functions $g_{\phi,t}$ trained on $\mathcal{D}_{t-1}$. Informally, this steers generation toward a reward-tilted distribution of the form $p_t(x) \propto p_\theta(x)\exp(\beta g_{\phi,t}(x))$, with each generated sample guided by a Thompson-style draw from the ensemble surrogate.

We benchmark on three combinatorially complete protein landscapes with full experimental fitness maps: GB1 IgG-Fc binding, TrpB enzymatic activity, and PhoQ-PhoP interface signaling (Wu et al., 2016; Johnston et al., 2024; Podgornaia & Laub, 2015). This removes the use of the surrogate-oracle used in SGPO, which can guide sampling toward sequences that have not been experimentally measured. Thus, every proposal from either method is scored against the same measured ground truth. More details about the set up can be found in J.

Results on combinatorially complete protein landscapes. SciOpt, evaluated with `MLP_Greedy` and `VBLL` on ESMC embeddings, matches or exceeds SGPO on Top-5% Count and NBSF on GB1 and PhoQ. On TrpB, SGPO retains a Top-5% lead throughout its active phase, and SciOpt `MLP_Greedy` draws roughly even by the budget cap. SciOpt `VBLL` shows wider seed-level variance than the `MLP_Greedy` variant due to a subset of seeds failing to escape early local optima.

Diversity collapse is universal across the three protein tasks, but most acutely present in the GB1 task (Barceló et al., 2024). At the four assayed positions, the natural-homolog distribution for GB1 concentrates $\sim 65\%$ of its mass on the wild-type combination VDGV, compared with $\sim 2\%$ (AVST) for PhoQ and $\sim 10\%$ (VFVS) for TrpB. This constitutes a failure mode of an otherwise strong evolutionary prior for protein design. The GB1 assayed positions coincide with the prior's mode, so the majority of proposals from $p_\theta(x)$ duplicate sequences labeled in rounds 0 and 1, leading to mode collapse.

Finally, the PhoQ MSA is the shallowest of the three targets, which limits the SGPO prior coverage of high-fitness regions. This leads it to perform marginally better than random on NBSF as well as Top-5% Count. On the other hand, TrpB has the deepest MSA of the three targets. As expected, SGPO performs the best overall on this task and the benefit of the rich prior is apparent by the steep early rise in the NBSF curve.

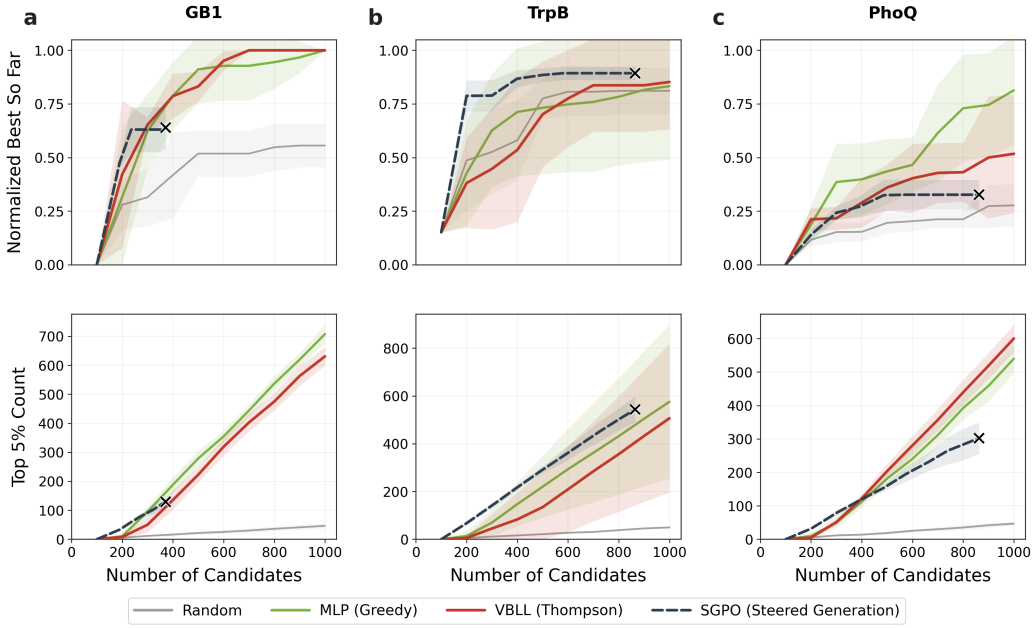

Figure 5. **SciOpt vs. Steered Generative Modeling.** Columns show three combinatorially complete protein landscapes `GB1`, `TrpB`, and `PhoQ`. Top row: NBSF. Bottom row: Top-5% count. SciOpt is evaluated with MLP and VBLL heads on ESM-C embeddings. SGPO samples high fitness candidates via steered generation, as discussed in 3.4. **X** marks the point after which SGPO can no longer propose unique candidate sequences, resulting in diversity collapse

4. Discussion

In summary, SciOpt unifies Bayesian optimization across proteins, DNA, RNA, and small molecules. In a systematic evaluation across dozens of tasks, we demonstrate that frozen scientific representations, when paired with lightweight neural surrogates and posterior-sampling acquisitions, provide a practical approach to sequential experimental design that efficiently identifies promising candidates and is highly competitive with alternative paradigms in AI-for-Science.

Limitations. The main limitation is that our evaluation is retrospective. All campaigns considered optimize over fixed candidate pools with known experimental labels. This design enables controlled comparison across methods, but it abstracts away experimental challenges. SciOpt also assumes that a candidate pool already exists. It selects from this pool rather than generating new proteins, molecules, RNA sequences, or regulatory elements, so its performance depends on the quality, diversity, and coverage of the upstream candidate generator. Several modeling choices further bound the scope of the study. We use a below-median initialization protocol, which makes the starting library deliberately challenging but does not fully separate the effect of initialization quality from surrogate quality. We use one primary foundation model per modality in the main sweep, with broader backbone comparisons left to the Appendix E. Finally, the benchmark is single-objective, while real campaigns often balance potency, selectivity, stability, synthesis, delivery, cost, and other constraints (Häse et al., 2018).

The comparisons to adjacent AI-for-Science paradigms should also be interpreted narrowly. The LLM experiment evaluates direct in-context candidate selection from a visible pool, not frontier agentic systems with retrieval, tool use, multi-step planning, or external simulators (Boiko et al., 2023; M. Bran et al., 2024; Swanson et al., 2025; Gottweis et al., 2025; Mitchener et al., 2025). The SGPO comparison evaluates a representative steered generative workflow on complete protein landscapes, but does not rule out stronger priors or improved diversity control. Furthermore, mutations on combinatorial protein landscapes were limited to 4 positions, which does not reflect protein engineering settings where biochemical knowledge of the target is limited and no tractable set of plausible positions (active-site residues, contact maps, or prior deep mutational scans) is available to constrain the search.

Practical takeaways. Despite these limitations, SciOpt provides a simple recipe for running Bayesian optimization over scientific candidate pools, requiring only three common components across domains. Neural surrogates paired with posterior sampling, especially VBLL and MLP ensembles, occupied the leading Pareto frontier. However, notably, a reinitialized greedy MLP emerged as a strong simple baseline, suggesting that stochastic training can provide useful implicit exploration even without an explicit uncer-

tainty model. We also find that calibration quality and optimization performance are largely decoupled in this regime (Appendix G.2): the best-calibrated surrogate (the exact GP) does not win on acquisition, while a badly-calibrated MLP ensemble sits on the Pareto frontier. What matters for fixed-pool BO is whether the posterior preserves the ranking of the upper tail and supplies enough disagreement to keep Thompson sampling exploratory, not whether it faithfully reproduces global coverage.

The batch size scaling experiments showed, at fixed total budget, smaller batches incorporate feedback more frequently, reaching strong NBSF values earlier and recovering more Top-5% candidates. This suggests that for sequential experimental design, there exists a tradeoff between model learnability and per-round scale. However, fortunately, the default well-plate size of 96 showed strong convergence in both peak candidate identification and elite coverage.

Future directions. The most important next step is validation in closed wet-lab loops (Burger et al., 2020; Shields et al., 2021; Tom et al., 2024). Live experimental campaigns would expose the operational constraints that fixed-pool benchmarks cannot capture, and would test whether the batch-size scaling trends remain useful as practical planning curves. A related direction is multi-objective, constrained, and multi-fidelity BO, since practical discovery campaigns rarely optimize a single scalar endpoint. Extending SciOpt to reason over multiple assay readouts, heterogeneous measurement costs, and proxy-to-validation assay hierarchies would make the framework closer to real campaign decisions.

More broadly, our findings point toward a complementary view of AI-for-Science. The SGPO generative baseline can propose strong candidates early, but our complete-landscape evaluation also shows that its supply of unique proposals can mode collapse under optimization pressure or a misaligned prior. This motivates closed-loop systems in which generative models expand the candidate set, while BO surrogates and acquisition functions can promote diverse sampling over shared foundation-model latent space. LLMs may be better positioned at a higher level of abstraction: retrieving and synthesizing biochemical knowledge from the literature, imposing constraints on search spaces, and automating the workflow that connects design, measurement, and model updating.

Acknowledgements

H.C. was supported by the GFSD Fellowship, Harvard GSAS Prize Fellowship, and Harvard James Mills Peirce Fellowship. U.P. and G.M.C. were supported by the US Department of Energy Grant DE-FG02-02ER63445. C.P. was supported by an NSF CAREER Award (IIS-2239780), DARPA grants DIAL-FP-038 and AIQ-HR00112520041, the Simons Collaboration on the Physics of Learning and Neural Computation, and the William F. Milton Fund from Harvard University. This work has been made possible in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence.

Impact Statement

This paper studies how to prioritize which experiments to run in scientific discovery campaigns spanning proteins, RNA, DNA, and small molecules. More sample-efficient experimental design can accelerate beneficial applications such as therapeutic, enzyme, and regulatory-element design, and can reduce the material and energy cost of large screening campaigns. The same optimization machinery could in principle be directed at harmful biological or chemical targets. However, the methods here operate over fixed, already-measured candidate pools and confer no capability beyond standard sequence- and molecule-property modeling. We do not foresee consequences that require specific highlighting beyond those already inherent to AI-for-Science.

References

Ament, S., Daulton, S., Eriksson, D., Balandat, M., and Bakshy, E. Unexpected improvements to expected improvement for Bayesian optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. arXiv:2310.20708.

Andreasson, J. O. L., Savinov, A., Block, S. M., and Greenleaf, W. J. Comprehensive sequence-to-function mapping of cofactor-dependent RNA catalysis in the glmS ribozyme. *Nature Communications*, 11:1663, 2020. doi: 10.1038/s41467-020-15540-1.

Arora, R., Angelo, M., Choe, C. A., Shearer, C. A., Kollasch, A. W., Qu, F., Weitzman, R., Gazizov, A., Gurev, S., Xie, E., Marks, D. S., and Notin, P. RNAGym: Large-scale benchmarks for RNA fitness and structure prediction. *bioRxiv preprint*, 2025. doi: 10.1101/2025.06.16.660049. Also ICLR 2025 GEM Workshop.

Balandat, M., Karrer, B., Jiang, D. R., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. arXiv:1910.06403.

Barbadilla-Martínez, L., Klaassen, N., Franceschini-Santos, V. H., Breda, J., Yücel, H., Hernández-Quiles, M., van Lieshout, T., Urzua Traslaviña, C. G., Boi, M. C. L., Akbarzadeh, M., et al. Regulatory grammar in human

promoters uncovered by mpra-based deep learning. *Nature*, pp. 1–10, 2026.

Barceló, R., Alcázar, C., and Tobar, F. Avoiding mode collapse in diffusion models fine-tuned with reinforcement learning. arXiv:2410.08315, 2024. Hierarchical Reward Fine-tuning (HRF); preserves sample diversity when fine-tuning diffusion models with RL rewards.

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023.

Boshar, S., Evans, B., Tang, Z., Picard, A., Adel, Y., Lorbeer, F. K., Rajesh, C., Karch, T., Sidbon, S., Emms, D., Mendoza-Revilla, J., Al-Ani, F., Seitz, E., Schiff, Y., Bornachot, Y., Hernandez, A., Lopez, M., Laterre, A., Beguir, K., Koo, P., Kuleshov, V., Stark, A., de Almeida, B. P., and Pierrot, T. A foundational model for joint sequence-function multi-species modeling at scale for long-range genomic prediction. *bioRxiv*, 2025. doi: 10.64898/2025.12.22.695963.

Brenan, L., Andreev, A., Cohen, O., Pantel, S., Kamburov, A., Cacchiarelli, D., Persky, N. S., Zhu, C., Bagul, M., Goetz, E. M., Burgin, A. B., Garraway, L. A., Getz, G., Mikkelsen, T. S., Piccioni, F., Root, D. E., and Johannessen, C. M. Phenotypic characterization of a comprehensive set of MAPK1/ERK2 missense mutants. *Cell Reports*, 17(4):1171–1183, 2016. doi: 10.1016/j.celrep.2016.09.061.

Brixi, G., Durrant, M. G., Ku, J., Naghipourfar, M., Poli, M., Sun, G., Brockman, G., Chang, D., Fanton, A., Gonzalez, G. A., et al. Genome modelling and design across all domains of life with evo 2. *Nature*, 652(8112):1349–1361, 2026.

Brunzema, P., Jordahn, M., Willes, J., Trimpe, S., Snoek, J., and Harrison, J. Bayesian optimization via continual variational last layer training. *International Conference on Learning Representations (ICLR)*, 2025.

Bryant, D. H., Bashir, A., Sinai, S., Jain, N. K., Ogden, P. J., Riley, P. F., Church, G. M., Colwell, L. J., and Kelsic, E. D. Deep diversification of an AAV capsid protein by machine learning. *Nature Biotechnology*, 39:691–696, 2021. doi: 10.1038/s41587-020-00793-4.

Burger, B., Maffettone, P. M., Gusev, V. V., Aitchison, C. M., Bai, Y., Wang, X., Li, X., Alston, B. M., Li, B., Clowes, R., Rankin, N., Harris, B., Sprick, R. S., and Cooper, A. I. A mobile robotic chemist. *Nature*, 583:237–241, 2020.

Bushuiev, A., Bushuiev, R., Pimenova, O., Zadorozhny, N., Samusevich, R., Manaskova, E., Kim, R. S., Stärk, H., Sedlar, J., Steinegger, M., Pluskal, T., and Sivic, J. One protein is all you need. In *International Conference on Learning Representations (ICLR)*, 2026. ProteinTTT; arXiv:2411.02109.

Cai, F., Zacour, K., Zhu, T., Tzeng, T.-R., Duan, Y., Liu, L., Pilla, S., Li, G., and Luo, F. ChemFM as a scaling law guided foundation model pre-trained on informative chemicals. *Communications Chemistry*, 9:3, 2026. doi: 10.1038/s42004-025-01793-8.

Chang, C.-Y., Azvar, M., Okwudire, C., and Al Kontar, R. LLINBO: Trustworthy LLM-in-the-loop Bayesian optimization. arXiv:2505.14756, 2025. arXiv preprint.

Chen, J., Hu, Z., Sun, S., Tan, Q., Wang, Y., Yu, Q., Zong, L., Hong, L., Xiao, J., Shen, T., et al. Interpretable rna foundation model from unannotated data for highly accurate rna structure and function predictions. *arXiv preprint arXiv:2204.00300*, 2022.

Dalla-Torre, H., Gonzalez, L., Mendoza-Revilla, J., Carranza, N. L., Grzywaczewski, A. H., Oteri, F., Dallago, C., Trop, E., de Almeida, B. P., Sirelkhatim, H., Richard, G., Skwark, M., Beguir, K., Lopez, M., and Pierrot, T. Nucleotide Transformer: Building and evaluating robust foundation models for human genomics. *Nature Methods*, 22:287–297, 2025. doi: 10.1038/s41592-024-02523-z.

Dallago, C., Mou, J., Johnston, K. E., Wittmann, B. J., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. FLIP: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv preprint*, 2021. doi: 10.1101/2021.11.09.467890.

de Almeida, B. P., Reiter, F., Pagani, M., and Stark, A. DeepSTARR predicts enhancer activity from DNA sequence and enables the de novo design of synthetic enhancers. *Nature Genetics*, 54(5):613–624, 2022. doi: 10.1038/s41588-022-01048-5.

Desautels, T., Krause, A., and Burdick, J. W. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. *Journal of Machine Learning Research*, 15:3873–3923, 2014.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186, 2019.

Didi, K., Alamdari, S., Lu, A. X., Wittmann, B. J., Johnston, K. E., Amini, A. P., Madani, A., Czeneszew, M., Dallago, C., and Yang, K. K. FLIP2: Expanding protein fitness landscape benchmarks for real-world machine learning applications. *bioRxiv*, 2026. doi: 10.64898/2026.02.23.707496.

Domingo, J., Diss, G., and Lehner, B. Pairwise and higher-order genetic interactions during the evolution of a tRNA. *Nature*, 558(7708):117–121, 2018. doi: 10.1038/s41586-018-0170-7.

Doud, M. B. and Bloom, J. D. Accurate measurement of the effects of all amino-acid mutations on influenza hemagglutinin. *Viruses*, 8(6):155, 2016. doi: 10.3390/v8060155.

EvolutionaryScale Team. ESM Cambrian (ESM-C): A new family of efficient protein language models. EvolutionaryScale, 2024.

Gantz, M., Mathis, S. V., Nintzel, F. E. H., Penner, M., Zurek, P. J., Knaus, T., Tseliou, V., Patel, E., Boros, D., Weberling, F.-M., Kenneth, M. R. A., Klein, O. J., Medcalf, E. J., Moss, J., Herger, M., Kaminski, T. S., Mutti, F. G., Lió, P., and Hollfelder, F. Microdroplet screening rapidly profiles a biocatalyst to enable its AI-assisted engineering. *bioRxiv*, 2024. doi: 10.1101/2024.04.08.588565.

García-Ortegón, M., Simm, G. N. C., Tripp, A. J., Hernández-Lobato, J. M., Bender, A., and Bacallado, S. DOCKSTRING: Easy molecular docking yields better benchmarks for ligand design. *Journal of Chemical Information and Modeling*, 62(15):3486–3502, 2022. doi: 10.1021/acs.jcim.1c01334.

Giacomelli, A. O., Yang, X., Lintner, R. E., McFarland, J. M., Duby, M., Kim, J., Howard, T. P., Takeda, D. Y., Ly, S. H., Kim, E., Gannon, H. S., Hurhula, B., Sharpe, T., Goodale, A., Fritchman, B., Steelman, S., Vazquez, F., Tsherniak, A., Aguirre, A. J., Doench, J. G., Piccioni, F., Roberts, C. W. M., Meyerson, M., Getz, G., Johannessen, C. M., Root, D. E., and Hahn, W. C. Mutational processes shape the landscape of TP53 mutations in human cancer. *Nature Genetics*, 50(10):1381–1387, 2018. doi: 10.1038/s41588-018-0204-y.

Ginsbourger, D., Le Riche, R., and Carraro, L. Kriging is well-suited to parallelize optimization. In Tenne, Y. and Goh, C.-K. (eds.), *Computational Intelligence in Expensive Optimization Problems*, volume 2 of *Adaptation Learning and Optimization*, pp. 131–162. Springer, 2010. doi: 10.1007/978-3-642-10701-6_6.

González, J., Dai, Z., Hennig, P., and Lawrence, N. D. Batch Bayesian optimization via local penalization. In *Artificial Intelligence and Statistics (AISTATS)*, 2016.

González-Duque, M., Michael, R., Bartels, S., Zainchkovskyy, Y., Hauberg, S., and Boomsma, W. A survey and benchmark of high-dimensional Bayesian optimization of discrete sequences. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, volume 37, 2024. doi: 10.52202/079017-4459.

Gosai, S. J., Castro, R. I., Fuentes, N., Butts, J. C., Mouri, K., Alasoadura, M., Kales, S., Nguyen, T. T. L., Noche, R. R., Rao, A. S., Joy, M. T., Sabeti, P. C., Reilly, S. K., and Tewhey, R. Machine-guided design of cell-type-targeting cis-regulatory elements. *Nature*, 634(8036):1211–1220, 2024. doi: 10.1038/s41586-024-08070-z.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Palepu, A., Sirkovic, P., Myaskovsky, A., Weissenberger, F., Rong, K., Tanno, R., Saab, K., Popovici, D., Blum, J., Zhang, F., Chou, K., Hassidim, A., Gokturk, B., Vahdat, A., Kohli, P., Matias, Y., Carroll, A., Kulkarni, K., Tomasev, N., Guan, Y., Dhillon, V., Vaishnav, E. D., Lee, B., Costa, T. R. D., Penadés, J. R., Peltz, G., Xu, Y., Pawlosky, A., Karthikesalingam, A., and Natarajan, V. Towards an AI co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.

Gupta, R., Hartford, J., and Liu, B. LLMs for Bayesian optimization in scientific domains: Are we there yet? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 15482–15510. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.838.

Guy, M. P., Young, D. L., Payea, M. J., Zhang, X., Kon, Y., Dean, K. M., Grayhack, E. J., Mathews, D. H., Fields, S., and Phizicky, E. M. Identification of the determinants of tRNA function and susceptibility to rapid tRNA decay by high-throughput in vivo analysis. *Genes & Development*, 28(15):1721–1732, 2014. doi: 10.1101/gad.245936.114.

Hachmann, J., Olivares-Amaya, R., Atahan-Evrenk, S., Amador-Bedolla, C., Sánchez-Carrera, R. S., Gold-Parker, A., Vogt, L., Brockway, A. M., and Aspuru-Guzik, A. The Harvard Clean Energy Project: Large-scale computational screening and design of organic photovoltaics on the World Community Grid. *The Journal of Physical Chemistry Letters*, 2(17):2241–2251, 2011. doi: 10.1021/jz200866s.

Haddox, H. K., Dingens, A. S., Hilton, S. K., Overbaugh, J., and Bloom, J. D. Mapping mutational effects along the evolutionary landscape of HIV envelope. *eLife*, 7:e34420, 2018. doi: 10.7554/eLife.34420.

Harrison, J., Willes, J., and Snoek, J. Variational Bayesian last layers. In *International Conference on Learning Representations (ICLR)*, 2024.

Hartman, E., Wallin, J., Malmström, J., and Olsson, J. Controllable protein design with particle-based Feynman-Kac steering. *arXiv preprint arXiv:2511.09216*, 2025.

Häse, F., Roch, L. M., and Aspuru-Guzik, A. Chimera: Enabling hierarchy based multi-objective optimization for self-driving laboratories. *Chemical Science*, 9:7642–7655, 2018.

Hernández-Lobato, J. M., Requeima, J., Pyzer-Knapp, E. O., and Aspuru-Guzik, A. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In *International Conference on Machine Learning (ICML)*, 2017.

Hie, B. L., Shanker, V. R., Xu, D., Bruun, T. U. J., Weidenbacher, P. A., Tang, S., Wu, W., Pak, J. E., and Kim, P. S. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2): 275–283, 2024. doi: 10.1038/s41587-023-01763-2.

Hino, T., Omura, S. N., Nakagawa, R., Togashi, T., Takeda, S. N., Hiramoto, T., Tasaka, S., Hirano, H., Tokuyama, T., Uosaki, H., et al. An ascas12f-based compact genome-editing tool derived by deep mutational scanning and structural analysis. *Cell*, 186(22):4920–4935, 2023.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Jarzab, A., Kurzawa, N., Hopf, T., Moerch, M., Zecha, J., Leijten, N., Bian, Y., Musiol, E., Maschberger, M., Stoehr, G., Becher, I., Daly, C., Samaras, P., Mergner, J., Spanier, B., Angelov, A., Werner, T., Bantscheff, M., Wilhelm, M., Klingenspor, M., Lemeer, S., Liebl, W., Hahne, H., Savitski, M. M., and Kuster, B. Meltome atlas—thermal proteome stability across the tree of life. *Nature Methods*, 17(5):495–503, 2020. doi: 10.1038/s41592-020-0801-4.

Jiang, K., Yan, Z., Di Bernardo, M., Sgrizzi, S. R., Villiger, L., Kayabolen, A., Kim, B., Carscadden, J. K., Hiraizumi, M., Nishimasu, H., Gootenberg, J. S., and Abudayyeh, O. O. Rapid in silico directed evolution by a protein language model with EVOLVEpro. *Science*, 387(6731): eadr6006, 2024. doi: 10.1126/science.adr6006.

Johnston, K. E., Almhjell, P. J., Watkins-Dulaney, E. J., Liu, G., Porter, N. J., Yang, J., and Arnold, F. H. A combinatorially complete epistatic fitness landscape in an enzyme active site. *Proceedings of the National Academy of Sciences*, 121(32):e2400439121, 2024. doi: 10.1073/pnas.2400439121.

Jones, E. M., Lubock, N. B., Venkatakrishnan, A. J., Wang, J., Tseng, A. M., Paggi, J. M., Latorraca, N. R., Cancilla, D., Satyadi, M., Davis, J. E., Babu, M. M., Dror, R. O., and Kosuri, S. Structural and functional characterization of G protein-coupled receptors with deep mutational scanning. *eLife*, 9:e54895, 2020. doi: 10.7554/eLife.54895.

Kalai, A. T., Nachum, O., Vempala, S. S., and Zhang, E. Why language models hallucinate. *arXiv preprint arXiv:2509.04664*, 2025.

Ke, S., Anquetil, V., Zamalloa, J. R., Maity, A., Yang, A., Arias, M. A., Kalachikov, S., Russo, J. J., Ju, J., and Chasin, L. A. Saturation mutagenesis reveals manifold determinants of exon definition. *Genome Research*, 28 (1):11–24, 2018. doi: 10.1101/gr.219683.116.

Klarner, L., Rudner, T. G. J., Morris, G. M., Deane, C. M., and Teh, Y. W. Context-guided diffusion for out-of-distribution molecular and protein design. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2407.11942.

Kobori, S. and Yokobayashi, Y. High-throughput mutational analysis of a twister ribozyme. *Angewandte Chemie International Edition*, 55(35):10354–10357, 2016. doi: 10.1002/anie.201605470.

Kobori, S., Takahashi, K., and Yokobayashi, Y. Deep sequencing analysis of aptazyme variants based on a pistol ribozyme. *ACS Synthetic Biology*, 6(7):1283–1288, 2017. doi: 10.1021/acssynbio.7b00057.

Kristiadi, A., Strieth-Kalthoff, F., Skreta, M., Poupart, P., Aspuru-Guzik, A., and Pleiss, G. A sober look at LLMs for material discovery: Are they actually good for Bayesian optimization over molecules? In *International Conference on Machine Learning (ICML)*, 2024.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Lee, J. M., Huddleston, J., Doud, M. B., Hooper, K. A., Wu, N. C., Bedford, T., and Bloom, J. D. Deep mutational scanning of hemagglutinin helps predict evolutionary fates of human H3N2 influenza variants. *Proceedings of the National Academy of Sciences*, 115(35):E8276–E8285, 2018. doi: 10.1073/pnas.1806133115.

Li, C., Qian, W., Maclean, C. J., and Zhang, J. The fitness landscape of a tRNA gene. *Science*, 352(6287):837–840, 2016. doi: 10.1126/science.aae0568.

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

M. Bran, A., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. ChemCrow: Augmenting large-language models with chemistry tools. *Nature Machine Intelligence*, 2024. arXiv:2304.05376 (2023).

Maclaurin, D., Duvenaud, D., and Adams, R. P. Early stopping is nonparametric variational inference. *arXiv preprint arXiv:1504.01344*, 2015.

McRae, E. K., Wan, C. J., Kristoffersen, E. L., Hansen, K., Gianni, E., Gallego, I., Curran, J. F., Attwater, J., Holliger, P., and Andersen, E. S. Cryo-em structure and functional landscape of an rna polymerase ribozyme. *Proceedings of the National Academy of Sciences*, 121(3):e2313332121, 2024.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. ESM-1v; demonstrates that protein language models capture mutation effects zero-shot and that fine-tuning on sequences from the target protein family further improves variant-effect prediction.

Mitchener, L., Yiu, A., Chang, B., Bourdenx, M., Nadolski, T., Sulovari, A., Landsness, E. C., Barabasi, D. L., Narayanan, S., Evans, N., Reddy, S., Foiani, M., Kamal, A., Shriver, L. P., Cao, F., Wassie, A. T., Laurent, J. M., Melville-Green, E., Caldas, M., Bou, A., Roberts, K. F., Zagorac, S., Orr, T. C., Orr, M. E., Zwezdaryk, K. J., Ghareeb, A. E., McCoy, L., Gomes, B., Ashley, E. A., Duff, K. E., Buonassisi, T., Rainforth, T., Bateman, R. J., Skarlinski, M., Rodriques, S. G., Hinks, M. M., and White, A. D. Kosmos: An AI scientist for autonomous discovery. *arXiv preprint arXiv:2511.02824*, 2025.

Nafar, A., Venable, K. B., and Kordjamshidi, P. Learning vs retrieval: The role of in-context examples in regression with large language models. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2025. arXiv:2409.04318 (2024).

Ngo, G., Trong, D. P., Nguyen, D., Gupta, S., and Venkatesh, S. Adaptive acquisition selection for bayesian optimization with large language models. *arXiv preprint arXiv:2602.07904*, 2026.

Notin, P., Kollasch, A. W., Ritter, D., van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., Frazer, J., Dias, M., Franceschi, D., Gal, Y., and Marks, D. S. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks*, 2023.

Ober, S. W., Rasmussen, C. E., and van der Wilk, M. The promises and pitfalls of deep kernel learning. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

Penić, R. J., Vlašić, T., Huber, R. G., Wan, Y., and Šikić, M. RiNALMo: General-purpose RNA language models can generalize well on structure prediction tasks. *Nature Communications*, 16:5671, 2025. doi: 10.1038/s41467-025-60872-5.

Peri, G., Gibard, C., Shults, N. H., Crossin, K., and Hayden, E. J. Dynamic RNA fitness landscapes of a group I ribozyme during changes to the experimental environment. *Molecular Biology and Evolution*, 39(3):msab373, 2022. doi: 10.1093/molbev/msab373.

Podgornaia, A. I. and Laub, M. T. Pervasive degeneracy and epistasis in a protein-protein interface. *Science*, 347 (6222):673–677, 2015. doi: 10.1126/science.1257360.

Pokusaeva, V. O., Usmanova, D. R., Putintseva, E. V., Espinar, L., Sarkisyan, K. S., Mishin, A. S., Bogatyreva, N. S., Ivankov, D. N., Akopyan, A. V., Avvakumov, S. Y., Povolotskaya, I. S., Filion, G. J., Carey, L. B., and Kondrashov, F. A. An experimental assay of the interactions of amino acids from orthologous sequences shaping a complex fitness landscape. *PLOS Genetics*, 15(4): e1008079, 2019. doi: 10.1371/journal.pgen.1008079.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning (ICML)*, 2023.

Pollice, R., dos Passos Gomes, G., Aldeghi, M., Hickman, R. J., Krenn, M., Lavigne, C., Lindner-D'Addario, M., Nigam, A., Ser, C. T., Yao, Z., and Aspuru-Guzik, A. Data-driven strategies for accelerated materials design. *Accounts of Chemical Research*, 54(4):849–860, 2021. doi: 10.1021/acs.accounts.0c00785.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Rafi, A. M., Nogina, D., Penzar, D., Lee, D., Lee, D., Kim, N., Kim, S., Kim, D., Shin, Y., Kwak, I.-Y., et al. A community effort to optimize sequence-based deep learning models of gene regulation. *Nature biotechnology*, 43(8): 1373–1383, 2025.

Ramos, M. C., Michtavy, S. S., Porosoff, M. D., and White, A. D. Bayesian optimization of catalysis with in-context learning. *ACS Central Science*, 2026. doi: 10.1021/acscentsci.5c02418. Articles ASAP, online April 2026; preprint arXiv:2304.05341 (2023).

Ranković, B., Griffiths, R.-R., and Schwaller, P. Large language models as uncertainty-calibrated optimizers for experimental discovery. *arXiv preprint arXiv:2504.06265*, 2025.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Rector-Brooks, J., Lambert, T., Skreta, M., Roth, D., Long, Y., Li, Z.-Q., Zhang, X., Cretu, M., Li, F.-Z., Ganapathy, T., Jin, E., Bose, A. J., Yang, J., Neklyudov, K., Bengio, Y., Tong, A., Arnold, F. H., and Liu, C.-H. General multimodal protein design enables DNA-encoding of chemistry. *arXiv preprint arXiv:2604.05181*, 2026. DISCO.

Riquelme, C., Tucker, G., and Snoek, J. Deep Bayesian bandits showdown: An empirical comparison of Bayesian deep networks for Thompson Sampling. In *International Conference on Learning Representations (ICLR)*, 2018.

Roberts, J. M., Beck, J. D., Pollock, T. B., Bendixsen, D. P., and Hayden, E. J. RNA sequence to structure analysis from comprehensive pairwise mutagenesis of multiple self-cleaving ribozymes. *eLife*, 12:e80360, 2023. doi: 10.7554/eLife.80360.

Romero, P. A., Krause, A., and Arnold, F. H. Navigating the protein fitness landscape with Gaussian processes. *Proceedings of the National Academy of Sciences*, 110 (3):E193–E201, 2013. doi: 10.1073/pnas.1215251110.

Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 234–241, 2015.

Russo, D. and Van Roy, B. Learning to optimize via posterior sampling. *Mathematics of Operations Research*, 39(4):1221–1243, 2014. doi: 10.1287/moor.2014.0650. arXiv:1301.2609.

Rychert, A., Spagnolo, G., and Posashkov, E. Reproducibility study of large language model Bayesian optimization. arXiv:2511.18891, 2025. arXiv preprint.

Salazar, J., Liang, D., Nguyen, T. Q., and Kirchhoff, K. Masked language model scoring. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 2699–2712, 2020. doi: 10.18653/v1/2020.acl-main.240.

Shields, B. J., Stevens, J., Li, J., Parasram, M., Damani, F., Martinez Alvarado, J. I., Janey, J. M., Adams, R. P., and Doyle, A. G. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 590(7844):89–96, 2021. doi: 10.1038/s41586-021-03213-y.

Soares, E., Vital Brazil, E., Shirasuna, V., Zubarev, D., Cerqueira, R., and Schmidt, K. An open-source family of large encoder-decoder foundation models for chemistry. *Communications Chemistry*, 8:193, 2025. doi: 10.1038/s42004-025-01585-0.

Soo, V. W. C., Swadling, J. B., Faure, A. J., and Warnecke, T. Fitness landscape of a dynamic RNA structure. *PLOS Genetics*, 17(2):e1009353, 2021. doi: 10.1371/journal.pgen.1009353.

Sourisseau, M., Lawrence, D. J. P., Schwarz, M. C., Storrs, C. H., Veit, E. C., Bloom, J. D., and Evans, M. J. Deep mutational scanning comprehensively maps how Zika envelope protein mutations affect viral growth and antibody escape. *Journal of Virology*, 93(23):e01291–19, 2019. doi: 10.1128/JVI.01291-19.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. W. Gaussian Process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

Stanton, S., Maddox, W. J., Gruver, N., Maffettone, P., Delaney, E., Greenside, P., and Wilson, A. G. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. In *International Conference on Machine Learning (ICML)*, 2022. LaMBO.

Stiffler, M. A., Hekstra, D. R., and Ranganathan, R. Evolvability as a function of purifying selection in TEM-1 β-lactamase. *Cell*, 160(5):882–892, 2015. doi: 10.1016/j.cell.2015.01.035.

Su, J., Han, C., Zhou, Y., Shan, J., Zhou, X., and Yuan, F. SaProt: Protein language modeling with structure-aware vocabulary. In *International Conference on Learning Representations (ICLR)*, 2024.

Swanson, K., Wu, W., Bulaong, N. L., Pak, J. E., and Zou, J. The virtual lab of AI agents designs new SARS-CoV-2 nanobodies. *Nature*, 646:716–723, 2025. doi: 10.1038/s41586-025-09442-9.

Tian, K., Mitchell, E., Zhou, A., Sharma, A., Rafailov, R., Yao, H., Finn, C., and Manning, C. D. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Tom, G., Schmid, S. P., Baird, S. G., Cao, Y., Darvish, K., Hao, H., Lo, S., Pablo-García, S., Rajaonson, E. M., Skreta, M., Yoshikawa, N., Corapi, S., Akkoc, G. D., Strieth-Kalthoff, F., Seifrid, M., and Aspuru-Guzik, A. Self-driving laboratories for chemistry and materials science. *Chemical Reviews*, 124(16):9633–9732, 2024.

Tome, J. M., Ozer, A., Pagano, J. M., Gheba, D., Schroth, G. P., and Lis, J. T. Comprehensive analysis of RNA-protein interactions by high-throughput sequencing-RNA affinity profiling. *Nature Methods*, 11(6):683–688, 2014. doi: 10.1038/nmeth.2970.

van der Flier, F., Estell, D., Pricelius, S., Dankmeyer, L., van Stigt Thans, S., Mulder, H., Otsuka, R., Goedegebuur, F., Lammerts, L., Staphorst, D., van Dijk, A. D. J., de Ridder, D., and Redestig, H. Enzyme structure correlates with variant effect predictability. *Computational and Structural Biotechnology Journal*, 23:3489–3497, 2024. doi: 10.1016/j.csbj.2024.09.007.

van Tilborg, D., Alenicheva, A., and Grisoni, F. Exposing the limitations of molecular machine learning with activity cliffs. *Journal of Chemical Information and Modeling*, 62(23):5938–5951, 2022. doi: 10.1021/acs.jcim.2c01073.

Wang, A. and Cho, K. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation (NeuralGen)*, pp. 30–36, 2019. doi: 10.18653/v1/W19-2304.

Wang, C., Uehara, M., He, Y., Wang, A., Biancalani, T., Lal, A., Jaakkola, T., Levine, S., Wang, H., and Regev, A. Fine-tuning discrete diffusion models via reward optimization with applications to DNA and protein design. *arXiv preprint arXiv:2410.13643*, 2024. DRAKES.

Wilson, J. T., Moriconi, R., Hutter, F., and Deisenroth, M. P. The reparameterization trick for acquisition functions. In *NeurIPS Workshop on Bayesian Optimization*, 2017.

Wilson, J. T., Borovitskiy, V., Terenin, A., Mostowsky, P., and Deisenroth, M. P. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning (ICML)*, 2020. arXiv:2002.09309.

Wu, N. C., Dai, L., Olson, C. A., Lloyd-Smith, J. O., and Sun, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife*, 5:e16965, 2016. doi: 10.7554/eLife.16965.

Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., and Hooi, B. Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. In *International Conference on Learning Representations (ICLR)*, 2024.

Yang, J., Chu, W., Khalil, D., Astudillo, R., Wittmann, B. J., Arnold, F. H., and Yue, Y. Steering generative models with experimental data for protein fitness optimization. *arXiv preprint arXiv:2505.15093*, 2025. SGPO; MSA-trained generative prior + classifier guidance on fitness labels.

Yoshizawa, T., Ishida, S., Sato, T., Ohta, M., Honma, T., and Terayama, K. A data-driven generative strategy to avoid reward hacking in multi-objective molecular design. *Nature Communications*, 16, 2025. doi: 10.1038/s41467-025-57582-3. DyRAMO.

Zhang, Z., Xiong, P., Zhang, T., Wang, J., Zhan, J., and Zhou, Y. Accurate inference of the full base-pairing structure of RNA by deep mutational scanning and covariation-induced deviation of activity. *Nucleic Acids Research*, 48 (3):1451–1465, 2020. doi: 10.1093/nar/gkz1192.

Zhang, Z., Hong, X., Xiong, P., Wang, J., Zhou, Y., and Zhan, J. Minimal twister sister-like self-cleaving ribozymes in the human genome revealed by deep mutational scanning. *eLife*, 12:RP90254, 2024. doi: 10.7554/eLife.90254.3.

A. Foundation models

The frozen main sweep fixes one primary foundation model (FM) per modality and holds it constant across every surrogate head and acquisition rule in App. B. For each (FM, task) pair we precompute two quantities offline: a D-dimensional pooled embedding per candidate, and a length-normalized log-likelihood-style score that we call the model's pseudo-likelihood (PLL). The BO loop never instantiates the FM: learned surrogates consume the precomputed embedding, and the `Zero_Shot` surrogate consumes the precomputed PLL. This decouples the comparison from FM inference throughput and lets us swap FMs on a fixed candidate pool without changing anything else.

A.1. Models considered

For every modality we use one primary FM in the 61-task sweep and pair it against one principal alternative for the backbone-sensitivity sweep (App. E). The primary FMs are domain-specific, publicly released, and compatible with the raw candidate representation in the corresponding benchmark. Each alternative varies the choice along the dimension most likely to matter in that modality: a structure-aware tokenizer for proteins (SaProt 650M vs. ESM-C 600M), a different RNA-encoder family at larger scale (RiNALMo-giga vs. RNA-FM), an autoregressive decoder vs. an encoder for DNA (Evo2-1B vs. NTv3-650M), and an encoder/decoder reconstruction model vs. a causal SMILES decoder for chemistry (SMI-TED vs. ChemFM-1B). Table 2 summarizes the eight models.

Table 2. **Foundation models used in this work.** Primary models are used for the 61-task surrogate sweep; alternatives are used in the paired backbone-sensitivity sweep (App. E). D is the pooled embedding dimension consumed by every learned surrogate. *PLL family* indicates which scoring procedure in §A.3 produces the cached prior score. "WT-marginal" is single-pass marginal scoring under a masked-LM head in the style of (Meier et al., 2021); "K-stripe masked" is a K-pass batched masking scheme used for U-Net-style nucleotide models (§A.3); "AR" is left-to-right next-token log-likelihood; "Recon." is encoder/decoder reconstruction cross-entropy. Length normalization divides the per-sequence log-probability by the number of scored non-special tokens.

Modality	Model	Role	Architecture	Params	D	PLL family
Protein	ESM-C 600M (EvolutionaryScale Team, 2024)	primary	encoder, transformer (rotary, SwiGLU)	600M	1152	WT-marginal
Protein	SaProt 650M (AF2) (Su et al., 2024)	alternative	encoder, MLM (446-vocab AA $\times$ 3Di bigrams)	650M	1280	WT-marginal, AA-marg.
RNA	RNA-FM (Chen et al., 2022)	primary	encoder, BERT-style MLM	100M	640	WT-marginal
RNA	RiNALMo-giga (Penić et al., 2025)	alternative	encoder, transformer (RoPE, SwiGLU)	650M	1280	WT-marginal
DNA	NTv3-650M (Boshar et al., 2025; Dalla-Torre et al., 2025)	primary	nucleotide encoder, U-Net + transformer	650M	1536	K-stripe masked
DNA	Evo2-1B (Brixi et al., 2026)	alternative	decoder, StripedHyena (Poli et al., 2023)	1B	1920	AR
Chemistry	ChemFM-1B (Cai et al., 2026)	primary	decoder, Llama-family causal LM	970M	2048	AR
Chemistry	SMI-TED (Soares et al., 2025)	alternative	encoder/decoder	289M	768	Recon.

A.2. Embedding extraction protocol

Each FM is run in inference mode using its native tokenizer. Inputs come from the curated task tables in App. D: protein, RNA, and DNA strings are uppercased before tokenization, while molecules are kept in their curated SMILES form. Rows with non-finite targets, empty inputs, or inputs longer than the model context window are dropped from that FM's candidate pool, and the BO comparisons within a modality use the intersection pool so that primary and alternative FMs see identical candidates. Inference uses bfloat16 mixed-precision autocast on GPU.

The pooled embedding is the mean of the final-layer hidden states over non-special tokens, where the special set is taken from each tokenizer's mask and includes BOS/CLS, EOS/SEP, PAD, MASK, UNK, and model-specific equivalents. We apply the same mean-over-non-special-tokens rule to every FM in Table 2 so that pooling is not a confound in the comparison, with two exceptions dictated by the model itself. For SMI-TED we use the autoencoder bottleneck vector that the decoder is trained to reconstruct from, since that is the representation the model is designed to expose. For NTv3 we pool the hidden state at the LM-head input rather than the post-skip output, so that the embedding stays on the same surface the model uses internally; the corresponding prior score is computed under the masking scheme below to avoid a trivial-copy path through the U-Net's skip connection. For Evo2 we additionally retain an intermediate Hyena-block representation alongside the final-layer pool, because several near-final block outputs are empirically degenerate and pooling at those depths would silently degrade the embedding.

A.3. Pseudo-likelihood and log-likelihood scoring

The cached prior is the model's length-normalized log-probability of the candidate under its native head, averaged over non-special tokens. For autoregressive decoders this is the standard one-token-shift log-likelihood. For encoder masked-LMs we use the single-pass wild-type-marginal scoring of (Meier et al., 2021), which approximates the true pseudo-log-likelihood of (Wang & Cho, 2019; Salazar et al., 2020) at one forward pass per candidate and is the standard cheap surrogate for protein-LM zero-shot variant scoring (Meier et al., 2021; Hie et al., 2024). For SMI-TED we use the model's native reconstruction objective: the per-token log-probability under the autoencoder decoder, averaged over the non-special positions. We refer to all four scalars uniformly as the PLL, and the `Zero_Shot` surrogate ranks candidates by this scalar with no further modeling.

Two FMs require a non-standard construction. SaProt's vocabulary is a 446-token product of amino-acid and 3Di structural tokens, so a direct gather would conflate sequence and structure preferences; we instead marginalize the bigram softmax over the 3Di axis within each input amino-acid letter, so the score reflects the model's distribution over the channel that varies in our candidates. NTv3 is a U-Net encoder (Ronneberger et al., 2015) whose full-resolution skip connection feeds a transform of the token embeddings into the pre-LM-head representation, making a single unmasked forward dominated by a trivial-copy identity rather than the trunk's predictions; we instead use a K-pass masked scheme that masks $1/K$ of the positions on each pass and reads log-probabilities only at the masked positions, with $K{=}7$ chosen to match the $\approx 15\%$ MLM training rate of (Devlin et al., 2019).

B. Surrogate heads

All surrogates in the main sweep consume the same per-task FM embeddings, cached once per (FM, task) pair as a parquet of fixed-size float vectors (Sec. 2, App. A). The only thing that varies across rows of Table 1 is the surrogate head and its acquisition rule. Any gap in NBSF or Top-5% count between two methods isolates the head, not the representation. This appendix records the eight surrogates compared in the main results, the exact hyperparameters used for each, and the ablations we ran when picking those settings.

B.1. Random (`Random`)

Uniform sampling without replacement from the heldout pool. The configuration omits both the surrogate and the acquisition blocks and sets `batch_strategy=random` on the optimizer. The featurizer is set to `identity` so no FM forward pass is performed.

B.2. Zero Shot (`Zero_Shot`)

The zero-shot baseline ranks the heldout pool by the FM's own pseudo-likelihood at each candidate, with no labeled-data fit and no parameter update over the campaign. This is the natural "FM-prior-only" control: it isolates how much of the optimization signal is already in the FM's pretraining likelihood, before any task-specific learning takes place. For protein and RNA assays this baseline is well-known to correlate with measured fitness for many DMS landscapes (Notin et al., 2023; Meier et al., 2021), which is what makes it a meaningful anchor rather than a trivial floor.

The cached score column is the per-FM PLL constructed by the scoring procedures of App. A.3: WT-marginal scoring for the masked-LM encoders (ESM-C 600M, RNA-FM), K-stripe masked PLL for NTv3-650M, and autoregressive log-likelihood for ChemFM-1B. All three are length-normalized so candidates of different sizes sit on the same scale and higher always means more likely under the FM. Acquisition is `ArgMax` on this scalar. Because the score is fixed at parquet-creation time and the surrogate has no parameters, the ranking of the heldout pool never changes between BO rounds; the loop simply removes each round's pick from the heldout set and selects the next-highest-scoring candidate. This is the sense in which `zero_shot` is open-loop: measured labels are recorded but ignored.

B.3. Random Forest (`RF_Greedy`, `RF_LogEI`)

Sklearn's `RandomForestRegressor` with 100 trees and standardized targets, fed the raw FM embedding as a fixed-size feature vector. Tree-level hyperparameters are sklearn's defaults: `criterion=friedman_mse`, unbounded depth, `min_samples_leaf=1`, and `max_features=1.0` (every split considers all features). The random-state seed is fixed at 42 so tree splits are deterministic across methods and seeds; the only stochasticity in the baseline is the BO loop. We use

the predictive mean of the forest (`rf.predict`) as the surrogate mean and the variance of the per-tree predictions as a heteroscedastic uncertainty estimate.

This recipe matches the EvolvePro setup (Jiang et al., 2024): a frozen protein-FM embedding, an unconstrained sklearn random forest with 100 estimators, and a greedy step that picks the candidate with the largest forest mean. We also report the `LogEI`-on-tree-variance variant in the same row of Table 1, which uses the tree variance as a posterior variance under BoTorch's `LogExpectedImprovement` (Ament et al., 2023) with $f^\star$ set to the running maximum on the labeled set. `LogEI` adds an exploration term but does not change the surrogate; the comparison between the two acquisitions inside this row isolates the effect of variance-aware exploration on top of an EvolvePro-style head.

B.4. Linear (`Linear`)

A single linear layer on the raw FM embedding, $f(x) = w^\top z + b$ with $z \in \mathbb{R}^D$, trained end-to-end with SGD rather than the closed-form ridge solution. Training is identical to the `MLP_Greedy` head described next: AdamW (`lr=3e-4`, `wd=1e-3`), batch size 32, gradient clip 1.0, MSE loss against standardized targets, up to 3000 epochs per BO round with patience-100 early stopping on the per-epoch training loss. The head is reinitialized at every BO round before retraining, so nothing carries between rounds beyond the labeled set. Acquisition is `ArgMax`.

We use this SGD-trained linear head rather than a closed-form ridge solution so the only architectural difference between `Linear` and `MLP_Greedy` is the head depth, with optimizer, loss, schedule, and reinitialization protocol all matched. This isolates the effect of the nonlinearity in the MLP from the effect of the optimization recipe.

B.5. MLP (`MLP_Greedy`)

A 2-layer MLP head on the raw FM embedding, with `hidden_features=128`, `num_layers=1`, `activation=elu`. Training uses AdamW (`lr=3e-4`, `wd=1e-3`), mini-batch 32, gradient clip 1.0, MSE on standardized targets, up to 3000 epochs with patience-100 early stopping on the per-epoch training loss. Acquisition is `ArgMax` on the predictive mean.

The head is reinitialized at every BO round before retraining; the model weights, the optimizer moments, and any learning-rate state are all dropped between rounds. This matters for the result in Sec. 3.1: the converged weights of an SGD-trained MLP from a fresh init can be intuitively read as an approximate draw from the implicit posterior over heads induced by the init distribution and the optimizer trajectory on the current labeled set, so an `ArgMax` on those weights is effectively a single-sample Thompson rule with the sampler hidden in the optimizer.

B.6. MLP Ensemble (`MLP_Ensemble`)

Five MLP heads with the same architecture and optimizer recipe as `MLP_Greedy`, trained jointly on the labeled set with summed MSE loss. The diversity comes from initialization, not from data: each head's linear weights are Kaiming-uniform initialized under a per-head seed (`torch.manual_seed(`$k \cdot 1337$`)` for the k-th head), so the five heads start at different points in weight space and converge to different solutions under the same SGD trajectory. We do not bootstrap or subsample the labeled data. All five heads are reinitialized at the start of every BO round.

The predictive mean is the average of the head outputs and the predictive variance is the empirical variance across the five outputs, both de-standardized before being passed to BoTorch's `LogExpectedImprovement`. We performed an ablation for this surrogate with LogEI and Thompson Sampling and found that Thompson Sampling performs far better. For batch BO the ensemble exposes a Thompson rule: each batch slot draws one head index uniformly and selects the candidate with the largest predicted mean under that head.

B.7. Gaussian Process (`GP`)

An exact GP from BoTorch with a constant mean, a scaled Matérn-5/2 ARD kernel directly on the raw FM embedding, and a Gaussian likelihood, fit by `fit_gpytorch_mll` (Balandat et al., 2020). Inputs are mapped to the unit cube (`Normalize`) and targets are standardized (`Standardize`). The kernel is initialized with output scale 1.0, lengthscale 1.0, and noise variance 1.0 under the constraint $\sigma_y^2 \geq 10^{-3}$, with the L-BFGS optimizer using a learning rate of 0.2 on the GP hyperparameters. Acquisition is BoTorch's `LogExpectedImprovement` (Ament et al., 2023) with $f^\star$ set to the running maximum.

The default `GP` runs the kernel on the full D-dimensional FM embedding (e.g. D=1152 for ESM-C 600M). We also

tested a shallow deep-kernel variant (`gp_bottleneck`) that prepends a trainable $\mathrm{Linear}(D \to 128) + \mathrm{ELU}$ projector before the kernel and co-trains the projector with the GP hyperparameters by marginal likelihood, using a separate AdamW parameter group at `lr=1e-3` and `wd=1e-3` (the kernel still uses `lr=0.2`). The bottleneck shape mirrors the VBLL projection exactly so any GP-vs-VBLL difference attributable to the projector is held fixed. Across the cohort the bottleneck variant performed worse than the no-projection `GP`, consistent with the well-known DKL overfitting failure mode at small labeled-set sizes (Ober et al., 2021), so we report the no-projection GP as our headline GP baseline.

B.8. Variational Bayesian Last Layer (`VBLL`)

Generative model. A Bayesian linear regression head sits on top of a learned $\mathrm{Linear}(D \to d) + \mathrm{ELU}$ projection $\phi_\theta : \mathbb{R}^D \to \mathbb{R}^d$ of the FM embedding z with $d = 128$ (Harrison et al., 2024). Writing $\phi_i = \phi_\theta(z_i)$ for brevity, the joint generative model on the labeled set $\mathcal{D}_t = \{(z_i, y_i)\}_{i=1}^{|\mathcal{D}_t|}$ is

$$y_i \mid \phi_i, w, \sigma_y^2 \sim \mathcal{N}\big(w^\top \phi_i,\ \sigma_y^2\big) \qquad \text{(likelihood)}, \tag{1}$$

$$w \sim \mathcal{N}(0,\ s_w I_d) \qquad \text{(weight prior)}, \tag{2}$$

$$\sigma_y^2 \sim \mathcal{W}^{-1}(\nu,\ s_\Sigma) \qquad \text{(noise prior)}, \tag{3}$$

where the scalar Inverse-Wishart on σ_y^2 reduces to an Inverse-Gamma in our single-output regression setting. The variational posterior factorizes as $q(w, \sigma_y^2) = q(w)\, q(\sigma_y^2)$ with a Gaussian $q(w) = \mathcal{N}(\mu, \Sigma)$ on the weights (full $d \times d$ covariance) and a deterministic $q(\sigma_y^2)$ parameterized by a single learned log-scale; the latter behaves as a MAP estimate of the noise variance under the Inverse-Wishart prior.

ELBO. Under this conjugate structure both the data-term expectation and $\mathrm{KL}(q \| p)$ are available in closed form, giving a deterministic per-batch ELBO:

$$\begin{aligned}\mathcal{L}(\theta, q) \;=\; \frac{1}{|\mathcal{D}_t|} \sum_{(z_i, y_i) \in \mathcal{D}_t} \Bigg[& \underbrace{\log \mathcal{N}\big(y_i;\ \mu^\top \phi_i,\ \sigma_y^2\big)}_{\text{predictive log-likelihood}} - \underbrace{\tfrac{1}{2}\, \phi_i^\top \Sigma\, \phi_i \,/\, \sigma_y^2}_{\text{weight-uncertainty penalty}} \Bigg] \\ & + \lambda_t \cdot \Bigg[\underbrace{\nu \log \sigma_y^{-2} - \tfrac{1}{2}\, s_\Sigma\, \sigma_y^{-2}}_{\text{Inverse-Wishart prior log-density on } \sigma_y^2} - \underbrace{\mathrm{KL}(\mathcal{N}(\mu, \Sigma) \,\|\, \mathcal{N}(0, s_w I_d))}_{\text{weight KL}} \Bigg].\end{aligned} \tag{4}$$

The data term is the closed-form expectation $\mathbb{E}_{q(w)}[\log \mathcal{N}(y_i;\ w^\top \phi_i,\ \sigma_y^2)]$, which decomposes into $\log \mathcal{N}(y_i;\ \mu^\top \phi_i,\ \sigma_y^2)$ minus the weight-uncertainty penalty $\frac{1}{2}\phi_i^\top \Sigma \phi_i / \sigma_y^2$. The Inverse-Wishart term acts as a regularizer on the noise variance under the deterministic $q(\sigma_y^2)$, preventing collapse to $\sigma_y^2 \to 0$. σ_y^2 inside the ELBO denotes the current variational point estimate.

Predictive distribution. For a candidate x with embedding $z(x)$, the posterior predictive over the noiseless function value $f(x) = w^\top \phi_\theta(z(x))$ is closed-form Gaussian:

$$p(f(x) \mid \mathcal{D}_t) \;=\; \mathcal{N}\Bigg(\underbrace{\mu^\top \phi_\theta(z(x))}_{\text{predictive mean}},\ \underbrace{\phi_\theta(z(x))^\top \Sigma\, \phi_\theta(z(x))}_{\text{predictive variance}} \Bigg). \tag{5}$$

The mean is the linear readout under the variational mean μ, and the variance is the quadratic form of the projected feature against the posterior covariance Σ (with the noise variance σ_y^2 added if a predictive over y is required). This gives any variance-aware acquisition rule a mean and variance per candidate without Monte Carlo sampling.

Hyperparameters. Table 3 lists the head's hyperparameters, grouped by role.

C. Acquisition functions

The acquisition rule turns the surrogate's predictive distribution at each candidate into a single score and selects the candidate (sequentially) or batch of candidates (under batched selection) that maximizes it. Three rules cover every method in Table 1: *Greedy* on the predictive mean, *LogEI* on the predictive mean and variance, and *Thompson sampling* on a posterior draw. This appendix gives the sequential form of each rule and the batch construction we pair with it.

Table 3. **VBLL head hyperparameters.** Grouped by role. The same settings are used across every task, modality, and FM in the main sweep; the only quantity that changes between rounds is the regularization weight λ_t, which is rescaled by the current dataset size.

Group	Hyperparameter	Value
Architecture	Projection width d	128
	Activation	ELU
	Covariance parameterization of $q(w)$	dense ($d \times d$)
Priors	Weight prior scale s_w	$1/d$
	Noise prior scale s_Σ	10
	Degrees of freedom ν	10
ELBO weighting	Regularization weight λ_t	$1/\lvert\mathcal{D}_t\rvert$ (per round)
Optimisation	Optimiser	AdamW
	Learning rate	10^{-3}
	Weight decay	10^{-3}
	Mini-batch size	64
Schedule	Gradient clip (max norm)	1.0
	Warmup	5 epochs, linear
	Decay	cosine to zero
	Max epochs / round	3000 (patience 100, smoothing 0.5)
Per-round reset	Target standardization	yes
	FM-embedding normalization	no
	Head / projector reinit each round	yes

C.1. Sequential acquisition rules

Let $\mu_t(x)$ and $\sigma_t^2(x)$ denote the posterior predictive mean and variance produced by the surrogate at round t. The sequential acquisition selects $x_t = \arg\max_{x \in \mathcal{X} \setminus \mathcal{D}_{t-1}} \alpha_t(x)$, where α_t is one of:

Greedy. $\alpha_t(x) = \mu_t(x)$, which selects the candidate with the largest predicted mean. For `Zero_Shot` this reduces to the FM's pseudo-likelihood, since the surrogate is the precomputed PLL itself; for the deterministic heads (`Linear`, `MLP_Greedy`, `RF_Greedy`) it is the predicted scalar with no variance term. This is the only rule available to surrogates without an uncertainty estimate.

LogEI. $\alpha_t(x) = \log \mathbb{E}[(f(x) - f^\star)_+]$, the logarithm of the expected improvement over the running maximum $f^\star = \max_{s \le t-1} y_s$. We use BoTorch's numerically stabilised `LogExpectedImprovement` (Ament et al., 2023) under the surrogate's predictive Gaussian, with $\mu_t(x)$ and $\sigma_t^2(x)$ supplied by the head: the marginal predictive of the GP (Rasmussen & Williams, 2006), the empirical mean and across-head variance of the MLP ensemble (de-standardized before being passed to the acquisition), and the predictive mean and tree-variance of the random forest for `RF_LogEI`. LogEI converts to expected improvement under a strict monotone transform, so the argmax of LogEI matches that of EI; the log form simply avoids the floating-point underflow that EI suffers when $f^\star$ is far above the current posterior mode (Ament et al., 2023).

Thompson sampling. $\alpha_t(x) = \tilde{f}_t(x)$, where $\tilde{f}_t$ is one draw from the surrogate's posterior over functions, and $x_t = \arg\max_x \tilde{f}_t(x)$ is the maximizer of that single draw (Russo & Van Roy, 2014). This matches the Thompson rule's regret guarantees and yields exploration without an explicit acquisition trade-off term: a candidate is selected because it is the optimum of *some* plausible function under the posterior, not because it has high mean or high variance per se.

For VBLL, the predictive in Eq. 5 is closed-form Gaussian, and a function draw is obtained by drawing the last-layer weights once: $\tilde{w} \sim \mathcal{N}(\mu, \Sigma)$ via the Cholesky factor of Σ, with $\tilde{f}_t(x) = \tilde{w}^\top \phi_\theta(z(x))$ scored over the pool by a single matrix-vector product. For the MLP ensemble, a function draw is obtained by sampling one head index $k \sim \mathrm{Uniform}\{1, \ldots, M\}$ and taking $\tilde{f}_t(x) = m_k(x)$, where m_k is the k-th head's prediction. The two constructions are not interchangeable: VBLL exposes an explicit weight posterior so each call is a fresh continuous draw, whereas the ensemble draws from a discrete posterior over M trained heads.

Thompson sampling is in principle also available for the GP, by drawing a joint posterior sample over the candidate pool from the GP's N-dimensional predictive Gaussian and taking its argmax (Hernández-Lobato et al., 2017; Wilson et al.,

2020). We do not use this rule in the main results because each draw requires a Cholesky factorization of an $N \times N$ kernel matrix at every BO round, which is $O(N^3)$ in the candidate pool size and prohibitive at the pool sizes considered here (N up to $\sim$10K) when run across 61 tasks, 5 seeds, and $T = 100$ sequential rounds; the GP is therefore paired with LogEI throughout. The deterministic heads (`Linear`, `MLP_Greedy`, `RF_Greedy`) and the random forest expose neither a weight posterior nor an ensemble of trained functions, so they are paired with `ArgMax` or `LogEI` rather than `Thompson`.

C.2. Batch construction

Batch BO at size k requires the surrogate to commit to k candidates without intermediate observations. The challenge is that the round-start posterior is the only information available for every member of the batch, so the natural single-step argmax of any acquisition concentrates the batch on a single mode of the posterior unless the batching rule explicitly diversifies.

We use the natural batch construction for each acquisition family. For Thompson sampling on VBLL and the MLP ensemble, batch members are independent posterior draws and their argmaxes; for VBLL we additionally apply a *projection-residual* diversity rule (App. C.3) that rejects redundant draws in covariance-weighted feature space. For the LogEI methods (GP, MLP ensemble paired with LogEI, RF paired with LogEI), the batch is the top-k candidates by the round-start LogEI score (App. C.4), since the surrogate's hyperparameters and $f^\star$ are not refit between picks within a single batch. For greedy methods the batch is the top-k candidates by predictive mean, and for the random baseline it is a uniform sample without replacement.

C.3. VBLL: Thompson sampling with projection-residual diversity

Sequential Thompson sampling under VBLL is exact (App. B.8): a single Cholesky factorization of Σ followed by an inner product over the candidate pool. The natural batch extension at size k draws k independent posterior samples $\tilde{w}^{(1)}, \ldots, \tilde{w}^{(k)} \sim q(w)$ and selects the argmax of each over the remaining heldout pool. This is correct when the posterior is broad, but for a sharp posterior the k argmaxes can concentrate on the same region of design space because every draw lies near the posterior mean. We layer a *projection-residual* diversity rule on top of the independent-sample construction to mitigate this collapse without changing the underlying Thompson semantics.

The rule operates in covariance-weighted feature space. For a candidate x, define $z(x) = L^\top \phi_\theta(x)$, where L is the Cholesky factor of the posterior covariance Σ (so $\|z(x)\|^2 = \phi_\theta(x)^\top \Sigma\, \phi_\theta(x)$ is the head's predictive variance at x, up to noise). After committing batch members $X_j = \{x_{(1)}, \ldots, x_{(j)}\}$, we maintain an orthonormal basis $Q \in \mathbb{R}^{r \times d}$ for $\mathrm{span}\{z(x_{(1)}), \ldots, z(x_{(j)})\}$ and accept the $(j{+}1)$-th proposal $x^\star$ iff its residual-energy ratio

$$\rho(x^\star) \;=\; \frac{\|z(x^\star)\|^2 - \|Q\, z(x^\star)\|^2}{\|z(x^\star)\|^2 + \varepsilon} \tag{6}$$

exceeds a fixed threshold δ. The numerator is the squared length of $z(x^\star)$ orthogonal to the span of already-committed members; $\rho \to 1$ when $z(x^\star)$ is fully orthogonal to that span (the candidate carries information about a new direction in the posterior covariance), and $\rho \to 0$ when it is essentially redundant. If $\rho(x^\star) < \delta$ we draw a fresh posterior sample and a fresh argmax, up to R resamples, after which we accept the last candidate as a fallback so the batch budget is always filled. On accept, Q is extended by Gram–Schmidt orthogonalization of $z(x^\star)$ against the current basis. The numerical tolerance ε avoids division by zero; if $\|z(x^\star)\|^2 \le \varepsilon$ the head reports essentially no posterior uncertainty at $x^\star$ and the candidate is accepted unconditionally so that diversity filtering does not block batch completion. We use $\delta = 10^{-3}$, $R = 5$, and $\varepsilon = 10^{-8}$ in all experiments.

The per-batch cost is $O(k(d^2 + Nd) + k^2 d)$: a single Cholesky reused across all draws, k posterior draws plus pool scorings, and a Gram–Schmidt update that grows linearly in the basis size. This is independent of $|\mathcal{D}_{t-1}|$ and of the FM size, since the FM enters only through the precomputed embedding z.

Ablation against plain Thompson sampling. We compared this rule against the plain independent-sample construction (the same algorithm with $\delta = 0$, equivalent to skipping the residual check) at every batch size $k \in \{24, 48, 96, 192, 384\}$ used in the main batch sweep. Across the cohort, the projection-residual variant slightly improved both NBSF and Top-5% count over plain Thompson sampling, with the gap most visible at the largest batch sizes where the posterior is sharpest and the redundancy mode is most pronounced. The improvement is small enough on aggregate that we report only the projection-residual variant in the main results, but the ablation justifies it as the default and is consistent with prior reports that Thompson batches benefit from explicit redundancy filtering at large k (Brunzema et al., 2025).

Algorithm 1 VBLL Batch Thompson Sampling with Projection-Residual Diversity

Require: Posterior $q(w) = \mathcal{N}(\mu, \Sigma)$, projection ϕ_θ, candidate pool $\mathcal{X}$, batch size k, max resamples R, threshold δ, tolerance ε

$L \leftarrow \mathrm{Cholesky}(\Sigma)$
$X_0 \leftarrow \emptyset,\ Q \leftarrow \emptyset$
for $j = 0, 1, \ldots, k-1$ **do**
  for attempt $= 0, 1, \ldots, R$ **do**
    Draw $\tilde{w} \sim \mathcal{N}(\mu, \Sigma)$
    $x^\star \leftarrow \arg\max_{x \in \mathcal{X} \setminus X_j} \tilde{w}^\top \phi_\theta(x)$
    $z \leftarrow L^\top \phi_\theta(x^\star)$
    if $Q = \emptyset$ **or** $\|z\|^2 \le \varepsilon$ **or** attempt $= R$ **then**
      accept $\leftarrow$ **true**
    else
      $\rho \leftarrow \left(\|z\|^2 - \|Qz\|^2\right) / \left(\|z\|^2 + \varepsilon\right)$
      accept $\leftarrow (\rho \ge \delta)$
    end if
    if accept **then**
      $X_{j+1} \leftarrow X_j \cup \{x^\star\}$
      Append the unit-normalized residual of z against Q to Q (Gram–Schmidt)
      break
    end if
  end for
end forX_k

C.4. LogEI surrogates: top-k acquisition argmax

For LogEI on the GP, the LogEI variant of the MLP ensemble, and `RF_LogEI`, the batch is the top-k candidates by the round-start LogEI score over the heldout pool. Within a single batch we do not refit the surrogate or update $f^\star$ between picks; the predictive mean, predictive variance, and improvement target are all held at their round-start values. This is by construction equivalent to the constant-liar batch rule with the model's posterior mean as the lie (Ginsbourger et al., 2010; Wilson et al., 2017): the lie does not change the LogEI score at the remaining points because the surrogate is not refit, so sequentially appending $(x^{(j)}, \mu_t(x^{(j)}))$ and recomputing the argmax produces the same ordering as a single sort by the original LogEI values. We use this top-k form in the main results because the equivalence is exact and the implementation is $O(N \log k)$ instead of $O(kN)$.

This batch rule does not encode any explicit diversity: at large k all k picks come from a single round-start posterior, and they tend to cluster around the same posterior maximum. The collapse is most visible on the GP, whose predictive variance is small in the well-explored part of the pool, and is the failure mode underlying the GP's poor scaling at large batch in Sec. 3.2.

Sequential kriging-believer ablation. We additionally tested a true sequential kriging-believer variant for the GP (Ginsbourger et al., 2010), in which after each pick the GP is conditioned on the pseudo-observation $(x^{(j)}, \mu_t(x^{(j)}))$ via Gaussian conditioning (no hyperparameter refit) and the LogEI is recomputed on the remaining pool. Because conditioning lowers the predictive variance near $x^{(j)}$, this rule diversifies the batch at the cost of k posterior conditionings per round. On our cohort the sequential variant actually performed slightly worse than top-k LogEI, and the runtime was substantially larger. We therefore report top-k LogEI as the headline GP batch rule.

C.5. Greedy and random surrogates

The greedy heads (`Linear`, `MLP_Greedy`, `RF_Greedy`, `Zero_Shot`) have no per-candidate uncertainty and no posterior to sample from. Their batch construction is top-k on the predictive mean (`ArgMax` on μ_t), with no diversity term. This matches their sequential rule one-to-one: nothing about the surrogate changes between members of the batch, so the k best candidates under the round-start posterior are exactly the k candidates the rule would select if run sequentially without intermediate observations. The `Random` baseline samples k candidates uniformly without replacement from the heldout

pool, independently of the surrogate.

D. Datasets

The main benchmark contains 61 fixed-pool scalar optimization tasks: 18 protein, 12 chemistry, 11 DNA, and 20 RNA. The broader benchmark collection contains additional audit, ablation, and legacy tasks; the selected cohort below is the one used for the paper's aggregate figures. Each task is treated as a finite candidate pool with a measured or computed scalar objective, and all methods compare on the same candidate set for that task.

Selection criteria. We include tasks that satisfy four constraints. First, the input is a discrete scientific object that can be passed to the modality FM without changing the representation: amino-acid sequence, DNA sequence, RNA sequence, or SMILES. Second, the target is a scalar regression objective with a well-defined optimization direction; ordinal or classification-only datasets such as `nucb` and `vapdiff` are excluded. Third, the deduplicated candidate pool is large enough for a 96-candidate initialization plus a 100-round sequential campaign without exhausting the landscape. Fourth, the source task is from an established benchmark or primary assay collection with enough provenance to reproduce the curated task. The selected set therefore emphasizes real biological assays and standard molecular-property benchmarks, while excluding chemistry reaction-procedure tasks whose natural-language inputs are not comparable to the SMILES-only chemistry FM pipeline, very small panels, deprecated duplicate tasks, and context-length edge cases.

Preprocessing and target convention. Preprocessing removes rows with missing inputs or missing labels, canonicalizes the input representation, drops exact duplicate candidates when the source permits, and keeps one scalar target per task. All tasks are maximization tasks except `enamine10k`, where lower docking score is better. For each seed, the initial 96 candidates are drawn from the below-median half of the task's target distribution, forcing every method to improve from a deliberately weak starting set.

Subsampling policy. Subsampling is deterministic and task-level, never method-level. When a source landscape is much larger than the campaign budget, the runtime CSV is a fixed subsample, usually stored as `<task>_10k.csv` and generated with seed 42; the full source files remain on disk for ablations when available. Other selected tasks use the full curated assay table because it is already a finite benchmark pool and embeddings were cached for the full CSV; the exact runtime N is reported in the tables. For the SGPO comparison, we use the full combinatorially complete landscapes for `gb1`, `phoq`, and `trpb_4site`, so every sequence proposed by either method can be scored against the complete experimental map. Separate LLM-context subsamples, such as `gb1_1500` and `gosai_sknsh_2100`, are not part of the 61-task BO cohort.

Protein tasks. Protein tasks combine FLIP (Dallago et al., 2021), FLIP2 (Didi et al., 2026), EvolvePro / ProteinGym (Notin et al., 2023), and complete combinatorial landscapes used for the SGPO comparison. The optimized properties cover viral packaging, binding, enzymatic activity, viral fitness, protein stability, and genetic-circuit signaling.

Table 4. **Protein benchmark tasks.** N_{orig} is the original curated dataset size before any campaign subsampling.

Task	Benchmark Suite	N_{orig}	Dataset description	Original citation
`aav`	FLIP	283,775	AAV2 capsid VP1-region DMS; maximize viral packaging viability.	Bryant et al., 2021 (Bryant et al., 2021)
`alpha_amylase`	FLIP2	7,467	Single-mutant *Bacillus* alpha-amylase assay; maximize DP3 substrate-hydrolysis activity.	van der Flier et al., 2024 (van der Flier et al., 2024)
`brenan`	EvolvePro ProteinGym	6,810	Human MAPK1 / ERK2 kinase DMS; maximize SCH-condition kinase signaling activity.	Brenan et al., 2016 (Brenan et al., 2016)
`cas12f`	EvolvePro	7,942	AsCas12f1 miniature CRISPR nuclease DMS; maximize genome-editing / DNA-cleavage efficiency.	Hino et al., 2023 (Hino et al., 2023)
`doud`	EvolvePro ProteinGym	10,716	H1N1 influenza hemagglutinin DMS; maximize viral replicative fitness.	Doud and Bloom, 2016 (Doud & Bloom, 2016)
`gb1`	FLIP ProteinGym	149,361	Four-position GB1 combinatorial DMS; maximize IgG-Fc binding fitness.	Wu et al., 2016 (Wu et al., 2016)
`giacomelli`	EvolvePro ProteinGym	7,448	Human TP53 DMS under etoposide selection; maximize tumor-suppressor function.	Giacomelli et al., 2018 (Giacomelli et al., 2018)
`haddox`	EvolvePro ProteinGym	12,863	HIV-1 Env DMS; maximize viral infectivity / replicative fitness.	Haddox et al., 2018 (Haddox et al., 2018)

Task	Benchmark Suite	N_{orig}	Dataset description	Original citation
hydro	FLIP2	24,935	HIS3 ortholog combinatorial DMS; maximize fitness across hydrophobic-core mutations.	Pokusaeva et al., 2019 (Pokusaeva et al., 2019)
ired	FLIP2	10,905	Imine reductase DMS; maximize catalytic fitness on the active substrate.	Gantz et al., 2024 (Gantz et al., 2024)
jones	EvolvePro ProteinGym	7,800	Human beta-2 adrenergic receptor DMS; maximize agonist-dependent signaling activity.	Jones et al., 2020 (Jones et al., 2020)
lee	EvolvePro ProteinGym	10,754	H3N2 influenza hemagglutinin DMS; maximize viral replication capacity.	Lee et al., 2018 (Lee et al., 2018)
meltome_human	FLIP	9,792	Human proteome melting-temperature dataset; maximize protein thermostability (T_m).	Jarzab et al., 2020 (Jarzab et al., 2020)
meltome_mixed	FLIP	23,340	Mixed-species proteome melting-temperature dataset; maximize protein thermostability (T_m).	Jarzab et al., 2020 (Jarzab et al., 2020)
phoq	PhoQ landscape	140,517	PhoQ four-site histidine-kinase landscape; maximize signaling fitness.	Podgornaia and Laub, 2015 (Podgornaia & Laub, 2015)
stiffler	EvolvePro ProteinGym	4,978	TEM-1 beta-lactamase DMS; maximize ampicillin-resistance fitness.	Stiffler et al., 2015 (Stiffler et al., 2015)
trpb_4site	FLIP2	159,129	Four-site PfTrpB combinatorial landscape; maximize beta-replacement reaction fitness.	Johnston et al., 2024 (Johnston et al., 2024)
zikv_E	EvolvePro ProteinGym	9,577	Zika-virus envelope-protein DMS; maximize viral growth / replication fitness.	Sourisseau et al., 2019 (Sourisseau et al., 2019)

Chemistry tasks. The chemistry cohort is SMILES-only. It includes three GOLLuM molecular-property tasks (Ranković et al., 2025) and nine MoleculeACE activity-cliff panels (van Tilborg et al., 2022). Reaction-yield tasks with `procedure` text are intentionally excluded from the main cohort because they use a different input representation.

Table 5. **Chemistry benchmark tasks.**

Task	Benchmark Suite	N_{orig}	Dataset description	Original citation
d3r_dopamine	MoleculeACE	3,651	Dopamine D3 receptor ligands; maximize pChEMBL / pK_i binding affinity.	van Tilborg et al., 2022 (van Tilborg et al., 2022)
delta_opioid	MoleculeACE	2,598	Delta-opioid receptor ligands; maximize pChEMBL / pK_i binding affinity.	van Tilborg et al., 2022 (van Tilborg et al., 2022)
enamine10k	GOLLuM DOCKSTRING	10,449	Enamine REAL molecules docked to a target receptor; minimize docking score.	García-Ortegón et al., 2022 (García-Ortegón et al., 2022)
f2_thrombin	MoleculeACE	2,750	Thrombin / coagulation factor II inhibitors; maximize pChEMBL potency.	van Tilborg et al., 2022 (van Tilborg et al., 2022)
fx_factor_x	MoleculeACE	3,093	Factor Xa inhibitors; maximize pChEMBL potency.	van Tilborg et al., 2022 (van Tilborg et al., 2022)
hrh3_histamine	MoleculeACE	2,862	Histamine H3 receptor ligands; maximize pChEMBL / pK_i binding affinity.	van Tilborg et al., 2022 (van Tilborg et al., 2022)
kappa_opioid	MoleculeACE	2,603	Kappa-opioid receptor ligands; maximize pChEMBL / pK_i binding affinity.	van Tilborg et al., 2022 (van Tilborg et al., 2022)
laser_multi10k	GOLLuM	10,000	Laser / photonic candidate molecules; maximize computed fluorescence oscillator strength.	Pollice et al., 2021 (Pollice et al., 2021)
mor_mu_opioid	MoleculeACE	3,142	Mu-opioid receptor ligands; maximize pChEMBL / pK_i binding affinity.	van Tilborg et al., 2022 (van Tilborg et al., 2022)

Task	Benchmark Suite	N_{orig}	Dataset description	Original citation
photovoltaics_pce10k	GOLLuM CEP	2,322,849	Organic photovoltaic donor molecules; maximize simulated power-conversion efficiency.	Hachmann et al., 2011 (Hachmann et al., 2011)
pparg	MoleculeACE	2,349	PPAR-gamma ligands; maximize pChEMBL activity / potency.	van Tilborg et al., 2022 (van Tilborg et al., 2022)
serotonin_5ht1a	MoleculeACE	3,317	Serotonin 5-HT1A receptor ligands; maximize pChEMBL / pK_i binding affinity.	van Tilborg et al., 2022 (van Tilborg et al., 2022)

DNA tasks. DNA tasks are MPRA or promoter-expression datasets. They are grouped into random/synthetic promoter benchmarks, DeepSTARR enhancer activity, Gosai human cis-regulatory element activity, and PARM / Barbadilla promoter MPRA tasks. The five `barbadilla_*` tasks are derived from the same promoter-fragment library and differ only in the expression or perturbation target.

Table 6. **DNA benchmark tasks.**

Task	Benchmark Suite	N_{orig}	Dataset description	Original citation
barbadilla_hepg2	PARM MPRA	4,449,234	Human promoter-fragment MPRA in HepG2 cells; maximize basal expression.	Barbadilla-Martínez et al., 2026 (Barbadilla-Martínez et al., 2026)
barbadilla_hepg2_nutlin3a_lfc	PARM MPRA	4,449,234	Same promoter-fragment library in HepG2; maximize Nutlin-3a perturbation log-fold change.	Barbadilla-Martínez et al., 2026 (Barbadilla-Martínez et al., 2026)
barbadilla_k562	PARM MPRA	4,449,234	Human promoter-fragment MPRA in K562 cells; maximize basal expression.	Barbadilla-Martínez et al., 2026 (Barbadilla-Martínez et al., 2026)
barbadilla_k562_heatshock_lfc	PARM MPRA	4,449,234	Same promoter-fragment library in K562; maximize heat-shock perturbation log-fold change.	Barbadilla-Martínez et al., 2026 (Barbadilla-Martínez et al., 2026)
barbadilla_k562_pma_lfc	PARM MPRA	4,449,234	Same promoter-fragment library in K562; maximize PMA perturbation log-fold change.	Barbadilla-Martínez et al., 2026 (Barbadilla-Martínez et al., 2026)
deepstarr_dev	DeepSTARR	484,052	Drosophila STARR-seq enhancer library; maximize developmental-enhancer activity.	de Almeida et al., 2022 (de Almeida et al., 2022)
deepstarr_hk	DeepSTARR	484,052	Same enhancer library; maximize housekeeping-enhancer activity.	de Almeida et al., 2022 (de Almeida et al., 2022)
dream	DREAM	7,413,183	Random synthetic yeast promoters; maximize measured promoter expression.	Rafi et al., 2025 (Rafi et al., 2025)
gosai_hepg2	Gosai CRE MPRA	798,064	Human cis-regulatory elements assayed in HepG2 cells; maximize regulatory activity.	Gosai et al., 2024 (Gosai et al., 2024)
gosai_k562	Gosai CRE MPRA	798,064	Human cis-regulatory elements assayed in K562 cells; maximize regulatory activity.	Gosai et al., 2024 (Gosai et al., 2024)
gosai_sknsh	Gosai CRE MPRA	798,064	Human cis-regulatory elements assayed in SK-N-SH cells; maximize regulatory activity.	Gosai et al., 2024 (Gosai et al., 2024)

RNA tasks. The RNA cohort is dominated by RNAgym (Arora et al., 2025) ribozyme and tRNA fitness landscapes, with tasks spanning self-cleavage, splicing, aptamer binding, mRNA expression, and tRNA suppression. Several sources contain

multiple assays on related ribozyme families; these are kept as separate tasks when they have distinct candidate pools or distinct measured targets.

Table 7. **RNA benchmark tasks.**

Task	Benchmark Suite	N_{orig}	Dataset description	Original citation
andreasson_ribozyme	RNAgym	7,343	Cofactor-dependent *glmS* self-cleaving ribozyme variants; maximize cleavage activity.	Andreasson et al., 2020 (Andreasson et al., 2020)
cpeb3_ribozyme	RNAgym	111,417	Human CPEB3 HDV-like ribozyme DMS; maximize self-cleavage activity.	Zhang et al., 2020 (Zhang et al., 2020)
domingo_trna	RNAgym	4,175	Yeast tRNA fitness landscape; maximize translation / aminoacylation fitness.	Domingo et al., 2018 (Domingo et al., 2018)
hairpin_ribozyme	RNAgym	22,578	Hairpin self-cleaving ribozyme DMS; maximize cleavage activity.	Roberts et al., 2023 (Roberts et al., 2023)
hammerhead_ribozyme	RNAgym	9,046	Type-III hammerhead ribozyme mutational library; maximize self-cleavage activity.	Roberts et al., 2023 (Roberts et al., 2023)
hdv_ribozyme	RNAgym	33,930	Hepatitis-delta-virus ribozyme DMS; maximize self-cleavage activity.	Roberts et al., 2023 (Roberts et al., 2023)
ke_mrna	RNAgym	5,559	mRNA exon-definition saturation mutagenesis; maximize splicing-reporter expression.	Ke et al., 2018 (Ke et al., 2018)
mcrae_5tu_ribozyme	RNAgym	74,942	5TU catalytic subunit of a triplet polymerase ribozyme (TPR); maximize polymerase activity.	McRae et al., 2024 (McRae et al., 2024)
mcrae_t1_ribozyme	RNAgym	47,503	t1 scaffolding subunit of a triplet polymerase ribozyme (TPR); maximize polymerase activity.	McRae et al., 2024 (McRae et al., 2024)
nelfe_aptamer	RNAgym	2,652	RNA aptamer variants selected against NELF-E; maximize protein-binding activity.	Tome et al., 2014 (Tome et al., 2014)
osa_twister_ribozyme	RNAgym	10,296	Oryza sativa twister-class ribozyme variants; maximize self-cleavage activity.	Kobori and Yokobayashi, 2016 (Kobori & Yokobayashi, 2016)
peri_ribozyme	RNAgym	16,383	Azoarcus group-I ribozyme activity landscape; maximize activity score.	Peri et al., 2022 (Peri et al., 2022)
pistol_aptazyme	RNAgym	16,383	Pistol aptazyme saturation library; maximize log-enrichment / cleavage activity.	Kobori et al., 2017 (Kobori et al., 2017)
soo_ribozyme	RNAgym	63,430	Tetrahymena group-I intron self-splicing ribozyme fitness landscape; maximize splicing activity.	Soo et al., 2021 (Soo et al., 2021)
sup4_trna_tyr	RNAgym	25,491	SUP4-oc tRNA-Tyr DMS; maximize amber-suppression / RTD-survival function.	Guy et al., 2014 (Guy et al., 2014)
trna_arg	RNAgym	65,537	Yeast SUP4-o tRNA-Arg(UCU) saturation landscape; maximize suppression-based fitness.	Li et al., 2016 (Li et al., 2016)
twister_ribozyme	RNAgym	10,296	Twister self-cleaving ribozyme DMS; maximize cleavage activity.	Roberts et al., 2023 (Roberts et al., 2023)
zhang_line1_full_ribozyme	RNAgym	69,583	Full LINE1-associated twister-sister-like self-cleaving ribozyme library; maximize cleavage activity.	Zhang et al., 2024 (Zhang et al., 2024)
zhang_line1_mini_ribozyme	RNAgym	149,710	Minimized LINE1-associated twister-sister-like ribozyme library; maximize cleavage activity.	Zhang et al., 2024 (Zhang et al., 2024)
zhang_or4k15_ribozyme	RNAgym	61,393	OR4K15-associated twister-sister-like self-cleaving ribozyme library; maximize cleavage activity.	Zhang et al., 2024 (Zhang et al., 2024)

E. Foundation-model comparison per modality

To isolate the effect of the foundation-model backbone, we rerun the entire surrogate sweep with the alternative FM swapped in for each modality (Table 2) while holding the task list, the BO loop, the seeds, and every surrogate hyperparameter fixed. The pairings are ESM-C 600M vs. SaProt 650M (protein), RNA-FM vs. RiNALMo-giga (RNA), NTv3-650M vs. Evo2-1B

(DNA), and ChemFM-1B vs. SMI-TED (chemistry). Each modality is evaluated on the intersection of candidate pools that both FMs admit so that any difference is attributable to the representation rather than to coverage. For every (FM, surrogate, task, seed) tuple we record two BO outcomes: the normalized best-so-far at the end of the budget, and the cumulative count of acquired candidates whose true objective lies above the 95th percentile of the pool.

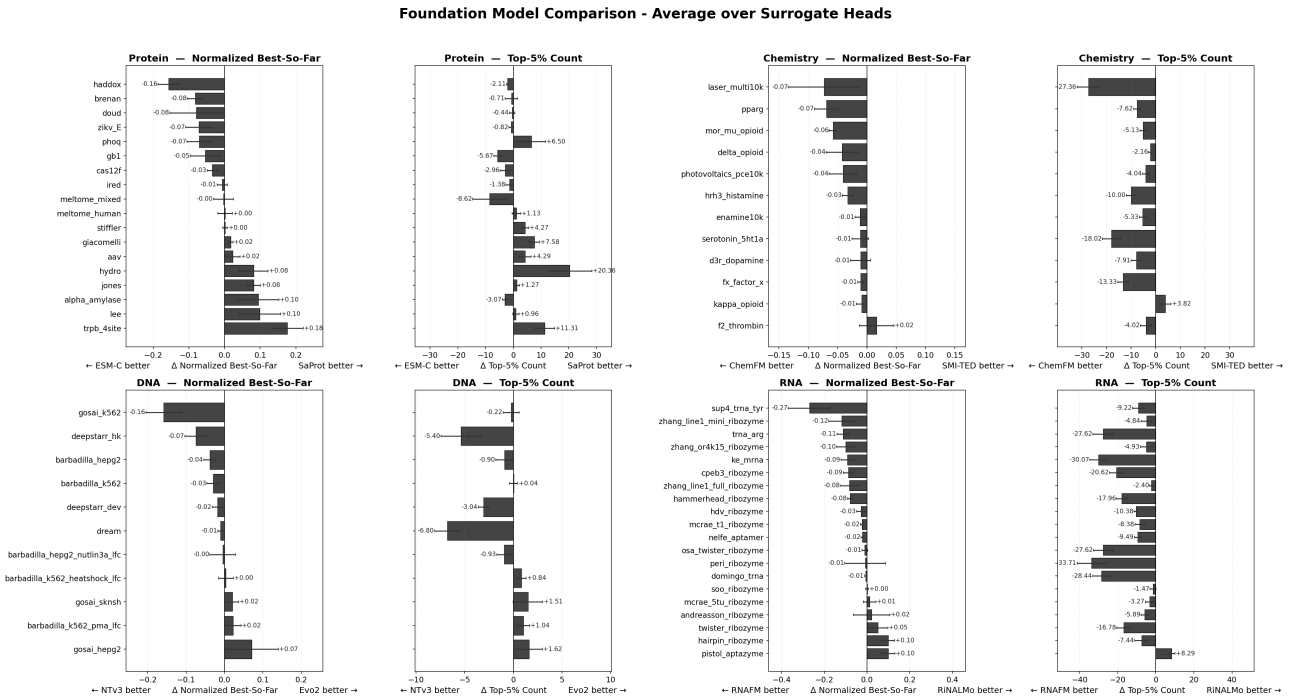

Figure 6. **Backbone sensitivity per modality.** Two panels per modality (normalized best-so-far on the left, top-5% acquired count on the right) report a per-task signed difference between the primary FM and its principal alternative (Table 2), averaged over surrogate heads and seeds. Each horizontal bar corresponds to one task; the sign indicates which FM is preferred (the labels under each axis name the winning side), and the magnitude reports the size of the gap on that metric. ESM-C 600M and SaProt 650M trade wins task by task and end up roughly on par overall, indicating that the structure-aware 3Di vocabulary helps on some landscapes but does not displace the sequence-only encoder as a default. Chemistry and RNA show clearer separations in favor of the primaries (ChemFM over SMI-TED, RNA-FM over RiNALMo-giga), with RNA exhibiting the largest and most uniform per-task gap despite RiNALMo-giga's higher parameter count. DNA is consistent with the rest: NTv3-650M is preferred over Evo2-1B across most tasks on both metrics, validating the encoder choice for sequences whose downstream signal is dominated by short-range regulatory motifs. The takeaway is that scale alone does not predict suitability for BO; the encoder vs. decoder distinction and the alignment between the pretraining objective and the candidate space matter at least as much.

F. Fine-Tuning

The main results in Section 3 freeze the foundation model and train only the surrogate head on cached embeddings. This appendix reports the two regimes that update the FM itself: *supervised* LoRA fine-tuning on the labeled BO set, and *self-supervised* LoRA pretraining on a target-family MSA. Both are evaluated on protein tasks; the SSL regime requires a wild-type sequence and an alignable family, which has no clean equivalent in the other modalities.

F.1. Supervised Fine-Tuning

In the Supervised regime, LoRA adapters (Hu et al., 2022) are co-trained with the surrogate head on $\mathcal{D}_{t-1}$ each round; adapters and head are reinitialized between rounds. We test three representative heads: `MLP_Greedy`, `VBLL`, and `GP_LogEI`, the last reproducing the deep-kernel-GP-with-LoRA recipe from GoLLuM (Ranković et al., 2025) with FM, LoRA rank, and acquisition matched to the frozen baseline.

Figure 7 shows that gains are head-dependent: `MLP_Greedy` and `VBLL` both improve with co-training, especially on Top-5%. `GP_LogEI` does not run to completion (cross marker, GP panel): backpropagating the marginal likelihood through both the kernel matrix and the adapter exhausts GPU memory partway through every protein task. This is intrinsic to the

GoLLuM-style recipe at our FM scale, not a tuning issue, and motivates our focus on variational and deterministic heads for SFT.

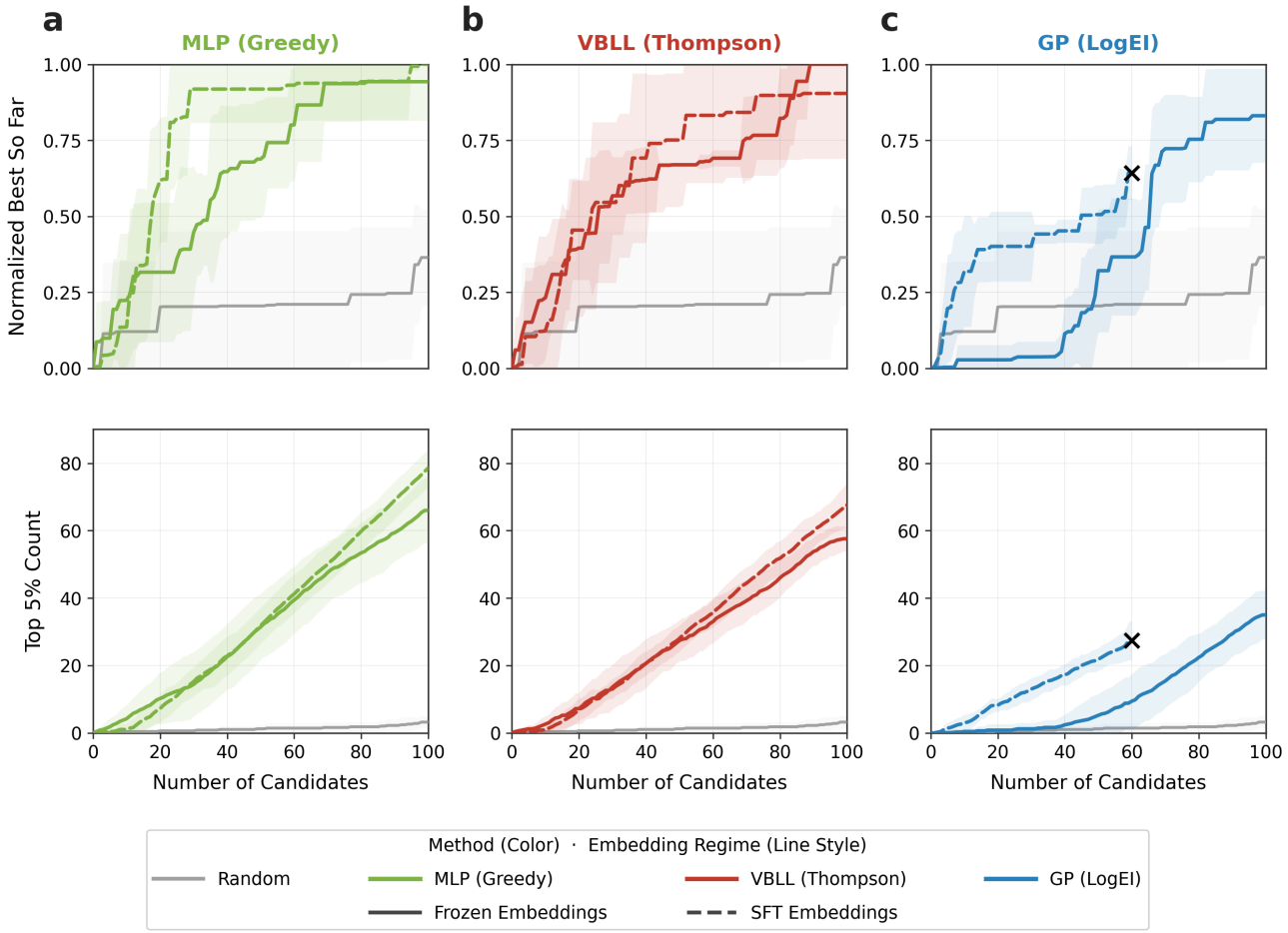

Figure 7. **Supervised LoRA fine-tuning on representative protein tasks.** NBSF (top) and Top-5% count (bottom) over T=100 sequential rounds for `MLP_Greedy`, `VBLL`, and `GP_LogEI`. Solid lines: frozen embeddings. Dashed lines: supervised LoRA. Bands are ±1 SEM across tasks and seeds. Gains are largest for `MLP_Greedy` and `VBLL`; `GP_LogEI` ends up consistently breaking at that step due to memory issues.

F.2. Self-Supervised Fine-Tuning

In the Self-Supervised regime, a single LoRA adapter is pretrained once *before* the BO loop using a masked-language-model objective on a target-family MSA, in the style of ESM-1v (Meier et al., 2021). For each task, the wild-type sequence is used as a query for a sequence-database search, hits are clustered at 80% identity, and training samples are weighted by inverse cluster frequency to prevent over-representation of dense regions of sequence space. The pretrained adapter is then loaded in place of a random init and frozen, so SSL acts as a *target-aware feature-extractor delta* on top of the base FM. We evaluate this regime on 15 of the 18 protein tasks; the remaining three (combinatorial-library or ortholog-panel datasets) lack a single wild-type query sequence, so the family-MSA construction step does not apply.

Figures 8 and 9 report per-task ΔNBSF and ΔTop-5% for each (task, head) pair. The picture is strongly heterogeneous: SSL is clearly beneficial on a subset of tasks (large green bars) and clearly harmful on another (large red bars), with the sign and magnitude depending on both task and head. Aggregating across tasks, mean NBSF changes are small and within 1 SEM for every method, while Top-5% coverage tends to improve for the simpler greedy heads (`Zero_Shot`, `RF_Greedy`, `Linear`, `MLP_Greedy`) and stays roughly flat for the stronger uncertainty-aware heads (`RF_LogEI`, `MLP_Ensemble`, `GP`, `VBLL`). Tasks where the wild type is already a high-fitness reference, and where the family MSA is dense and well-aligned, are the most consistent winners. We therefore do *not* adopt SSL as a default in the main results; it is

a useful task-conditional lever rather than a uniform improvement.

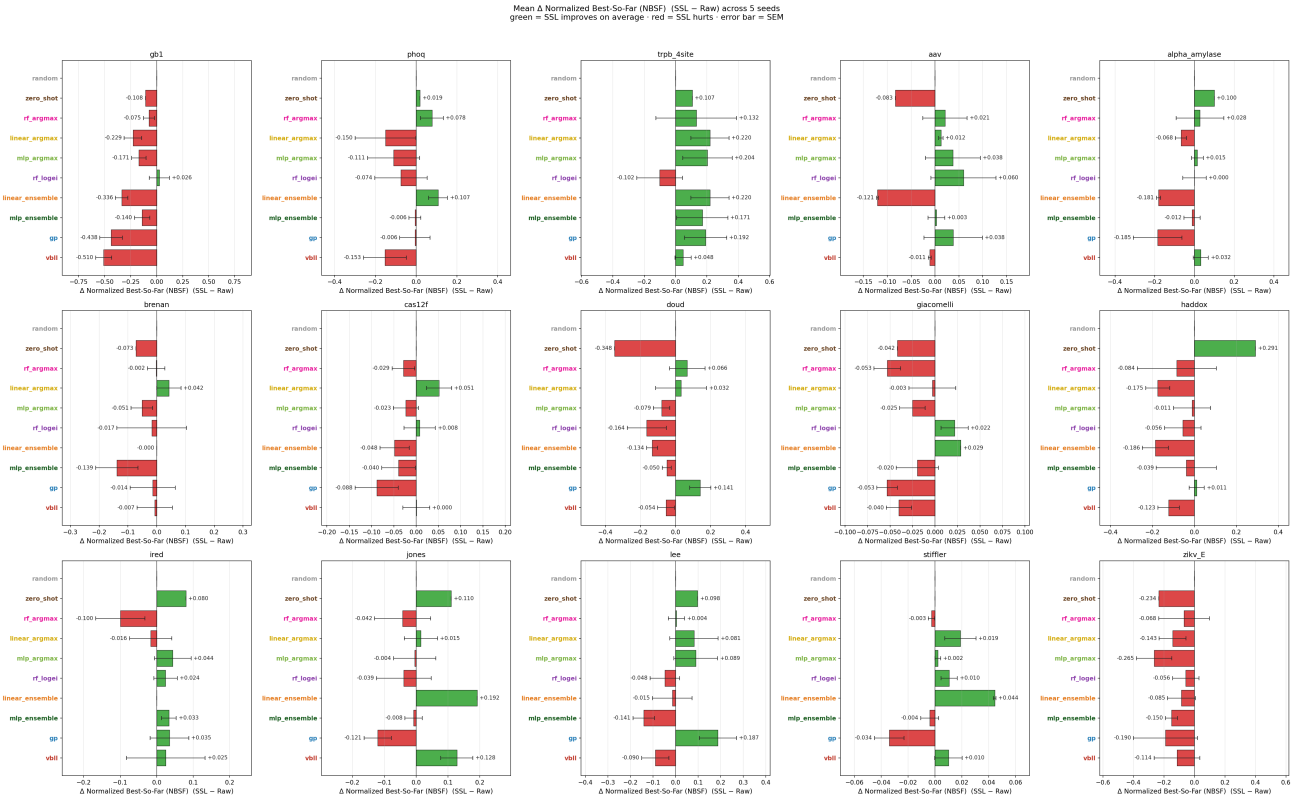

Figure 8. **Per-task ΔNBSF (SSL − original),** 15 **protein tasks.** One panel per task; bars are method-level differences, green for SSL improving, red for SSL degrading. Error bars are ±1 SEM across seeds. Aggregate NBSF changes are within 1 SEM, but per-task swings are large and bidirectional.

Figure 9. **Per-task ΔTop-5% (SSL − original), 15 protein tasks.** Same layout as Figure 8 for elite-candidate coverage. Greedy heads gain most often; uncertainty-aware heads are roughly unchanged on aggregate.

G. Sequential Bayesian Optimization

This appendix expands on the sequential results in §3.1. We first show the across-task Pareto frontier and the round-by-round learning curves that produce it, then turn to the variance side via expected calibration error (ECE) for the uncertainty-aware heads.

G.1. Detailed Results

Figure 10 pools all 61 tasks into a single endpoint summary per method, with both metrics evaluated at the end of the T=100-round budget. `MLP_Ensemble` and `VBLL` sit at the upper-right frontier, `MLP_Greedy` is essentially tied with them on best-so-far and only marginally behind on top-5% count, and `GP` matches the leaders on best-so-far but acquires meaningfully fewer top-5% candidates. `Linear`, `RF_LogEI`, and `RF_Greedy` form the middle tier, while `Zero_Shot` and `Random` are dominated on both axes. Figure 11 unrolls those endpoints over rounds and splits by modality: rows are protein, chemistry, RNA, and DNA, the left column is normalized best-so-far, and the right column is top-5% count. The two columns behave differently in time. Best-so-far gaps narrow late as most methods eventually find a near-optimum, while top-5% gaps keep widening for the remainder of the budget, because that metric rewards continuing to acquire high-value designs after the first one is in hand. `Random` and `Zero_Shot (Likelihood)` stay roughly linear throughout, consistent with non-adaptive sampling.

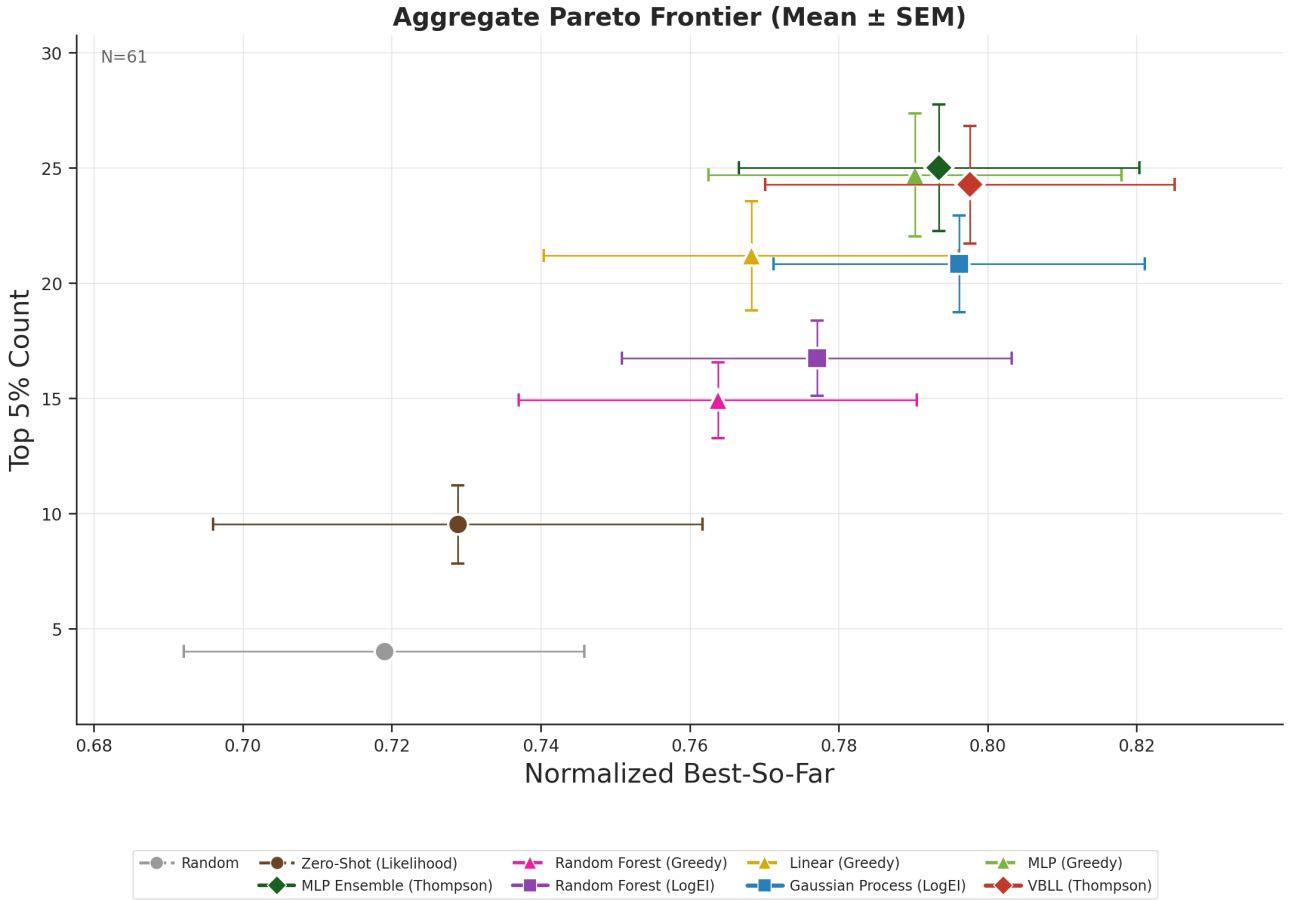

Figure 10. **Aggregate Pareto frontier of normalized best-so-far vs. top-5% count, pooled across all 61 sequential tasks (T=100, k=1).** Each marker is one method aggregated over tasks and seeds; error bars are ±1 SEM. The upper-right corner is the joint optimum.

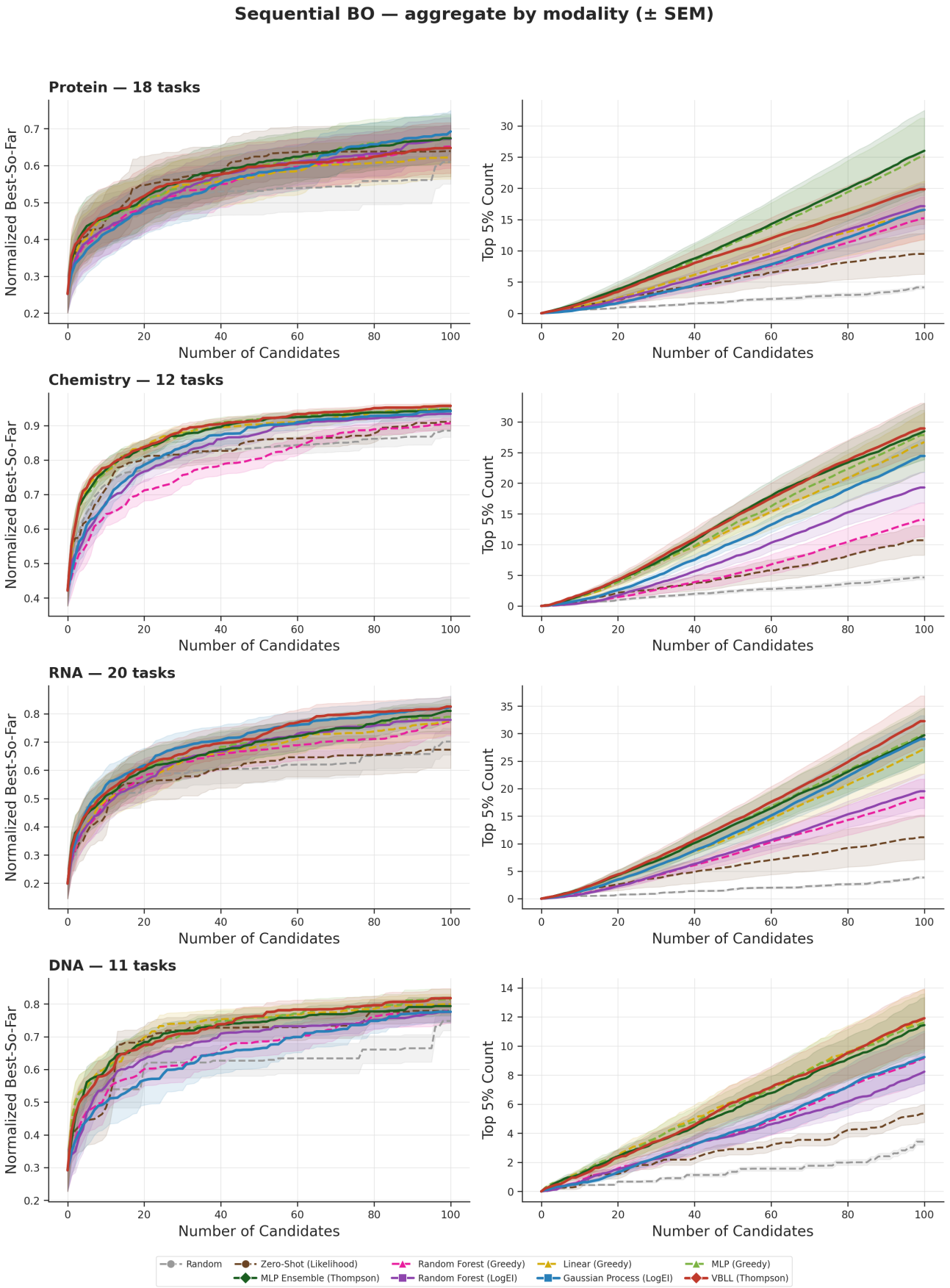

Figure 11. **Per-modality sequential learning curves.** Rows are protein (18 tasks), chemistry (12), RNA (20), and DNA (11); the left column is normalized best-so-far and the right column is top-5% count, both as a function of round t. Bands are ± 1 SEM across tasks and seeds. Best-so-far gaps tighten as the budget grows; top-5% gaps widen.

G.2. Calibration

For the three surrogates that expose a closed-form Gaussian predictive distribution (`GP`, `VBLL`, and `MLP_Ensemble`) we additionally track cumulative expected calibration error (ECE) on the heldout pool, recomputed at every BO round. ECE is the standard binned reliability gap between predicted and empirical coverage; a perfectly calibrated regression posterior would drive it to zero as the loop accumulates labels. The exact GP is the calibration reference by construction. VBLL starts uncalibrated but tightens within the first $\sim$20 rounds and then trails the GP by a roughly constant offset for the remainder of the budget. The MLP ensemble stays badly miscalibrated throughout: the disagreement signal across heads does not match the true predictive variance and barely improves with more data. The cold-start `MLP_Greedy` and the random-forest heads are omitted because their uncertainty is either implicit (round-to-round reinitialization) or non-Gaussian (tree variance), and ECE under a Gaussian assumption is not the right diagnostic for them.

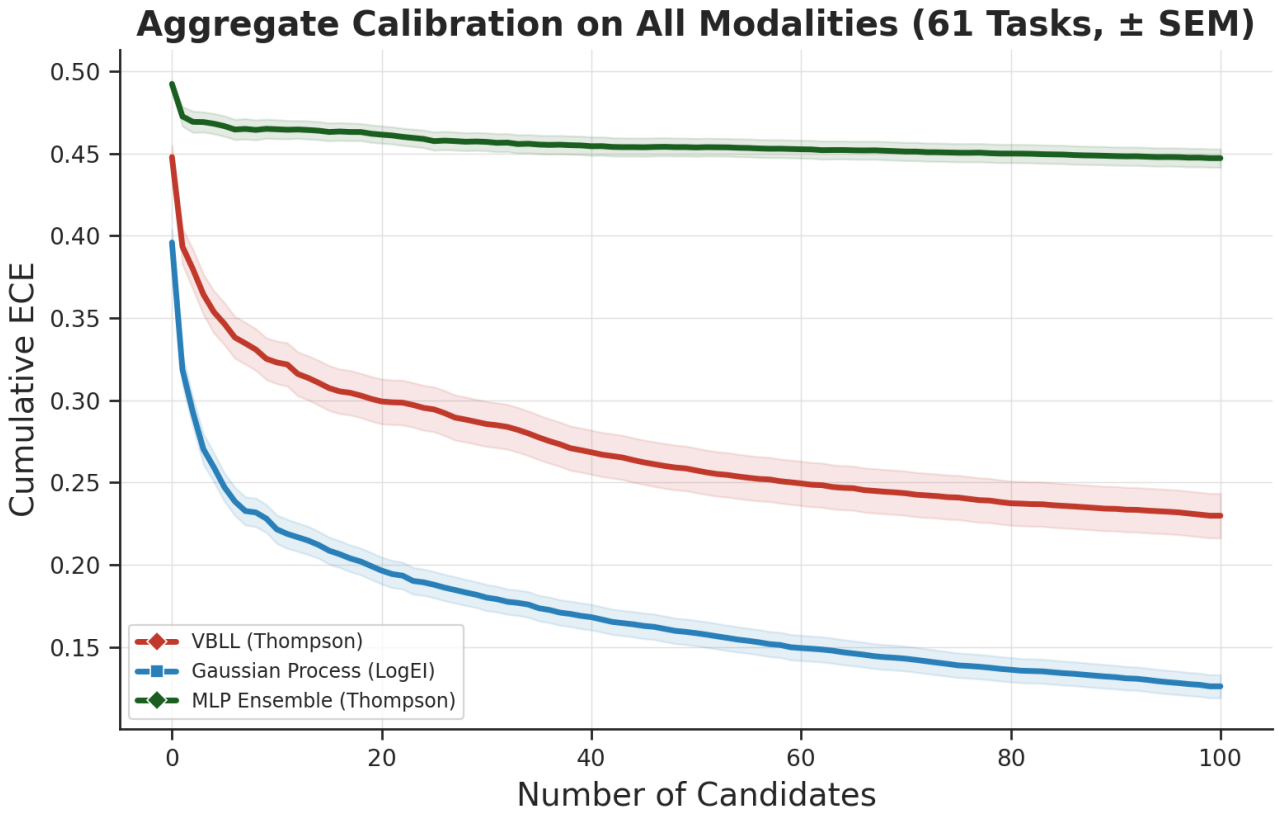

Figure 12. **Aggregate cumulative ECE over all** 61 **sequential tasks (± SEM).** Lower is better. `GP` is the calibration ceiling. `VBLL` tightens quickly and trails the GP by a roughly constant offset. `MLP_Ensemble` stays nearly flat at high ECE.

The surprising takeaway is that calibration quality and BO performance are essentially decoupled in this regime. The MLP ensemble has by far the worst ECE in Figure 12, yet it sits on the Pareto frontier in Figure 10 and matches or beats the GP on top-5% count across modalities in Figure 11. The GP, conversely, is the best-calibrated method by a wide margin and still loses on the acquisition metric. What appears to matter for discrete BO over a fixed candidate pool is whether the posterior preserves the *ranking* of high-value candidates and supplies enough disagreement to keep Thompson sampling exploratory; faithfully reproducing posterior coverage is neither necessary nor sufficient for that. A natural interpretation is that ECE is computed over the entire heldout pool, whereas BO only ever queries the upper tail, so a head whose variances are globally miscalibrated can still produce a useful ordering of the candidates that actually get acquired. The two axes should therefore be read together; the calibration plot is informative about the posterior, but it is not a reliable proxy for downstream optimization performance.

H. Batch Bayesian Optimization

This appendix expands on the batch results in §3.2. We fix the total acquisition budget at 768 candidates and sweep batch sizes $k \in \{24, 48, 96, 192, 384\}$ over $T \in \{32, 16, 8, 4, 2\}$ rounds. At the practical k=96 setting (one 96-well plate per round), we report the Pareto frontier of NBSF against Top-5% count both pooled and split by modality, then show the round-by-round curves underlying the scaling result of Figure 3.

H.1. Detailed Results

Figure 13 pools across all 61 tasks at k=96, and Figure 14 disaggregates by modality. The qualitative picture closely matches the sequential setting (§G.1): `VBLL` and `MLP_Ensemble` occupy the upper-right frontier with `MLP_Greedy` adjacent, the random-forest, linear, and GP variants form the middle tier, and `Random`/`Zero_Shot` trail. The main batching effect is on coverage: the Top-5% axis compresses for every method relative to sequential, since the surrogate cannot incorporate within-batch labels before committing all k picks. Figure 15 shows the round-by-round trajectories at each k. Smaller batches reach a given NBSF level in fewer cumulative candidates, and the gap on Top-5% count remains visible at the end of the budget; the curves converge slowly as k grows, consistent with elite-set coverage being feedback-round-limited rather than candidate-budget-limited.

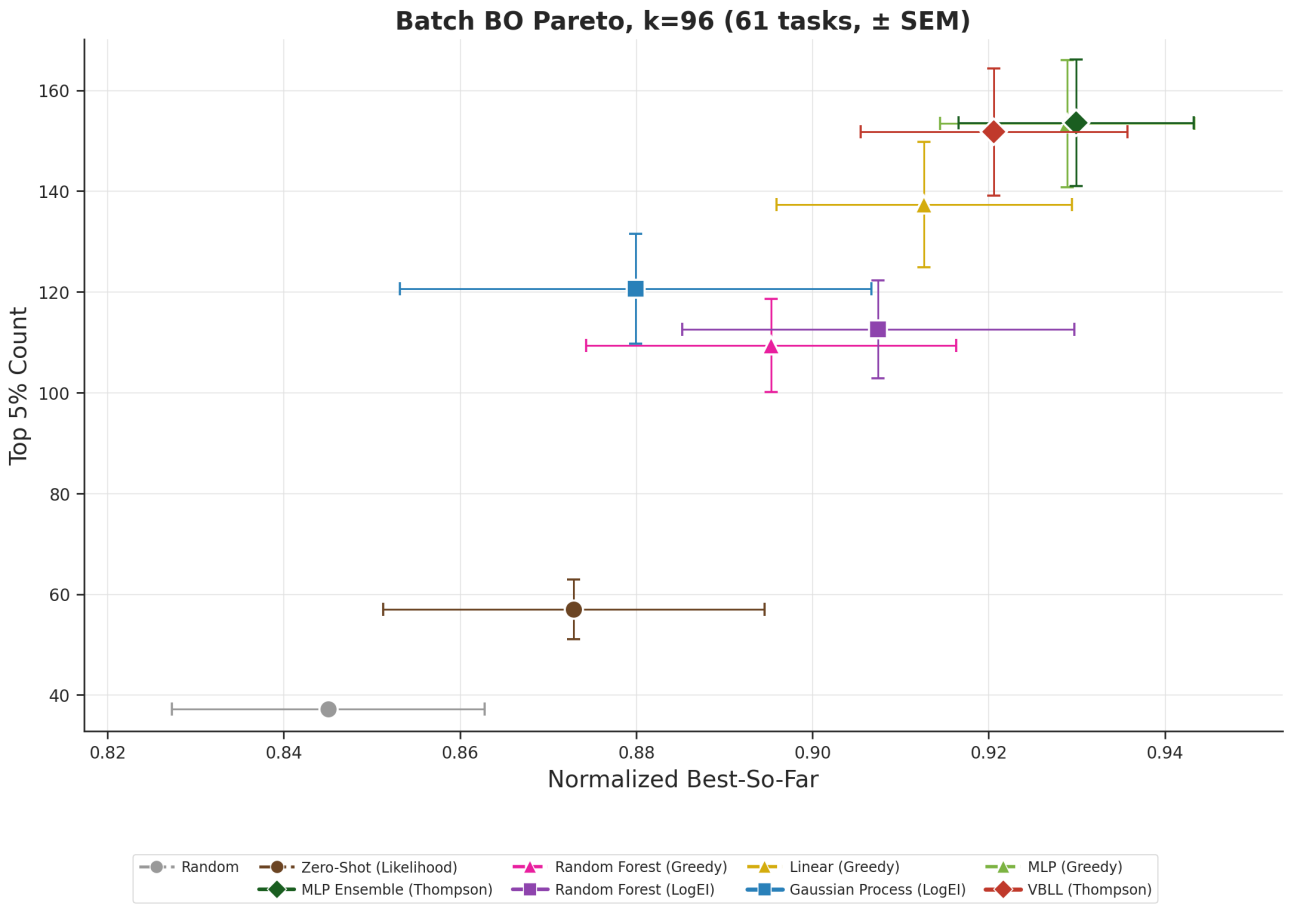

Figure 13. **Pareto frontier of NBSF vs. Top-5% count, pooled across all 61 tasks (batch, k=96, T=8).** Each marker is one method aggregated over tasks and seeds; error bars are ±1 SEM. The frontier ordering matches the sequential setting (Figure 10); absolute Top-5% counts are lower because batches commit k picks per posterior update.

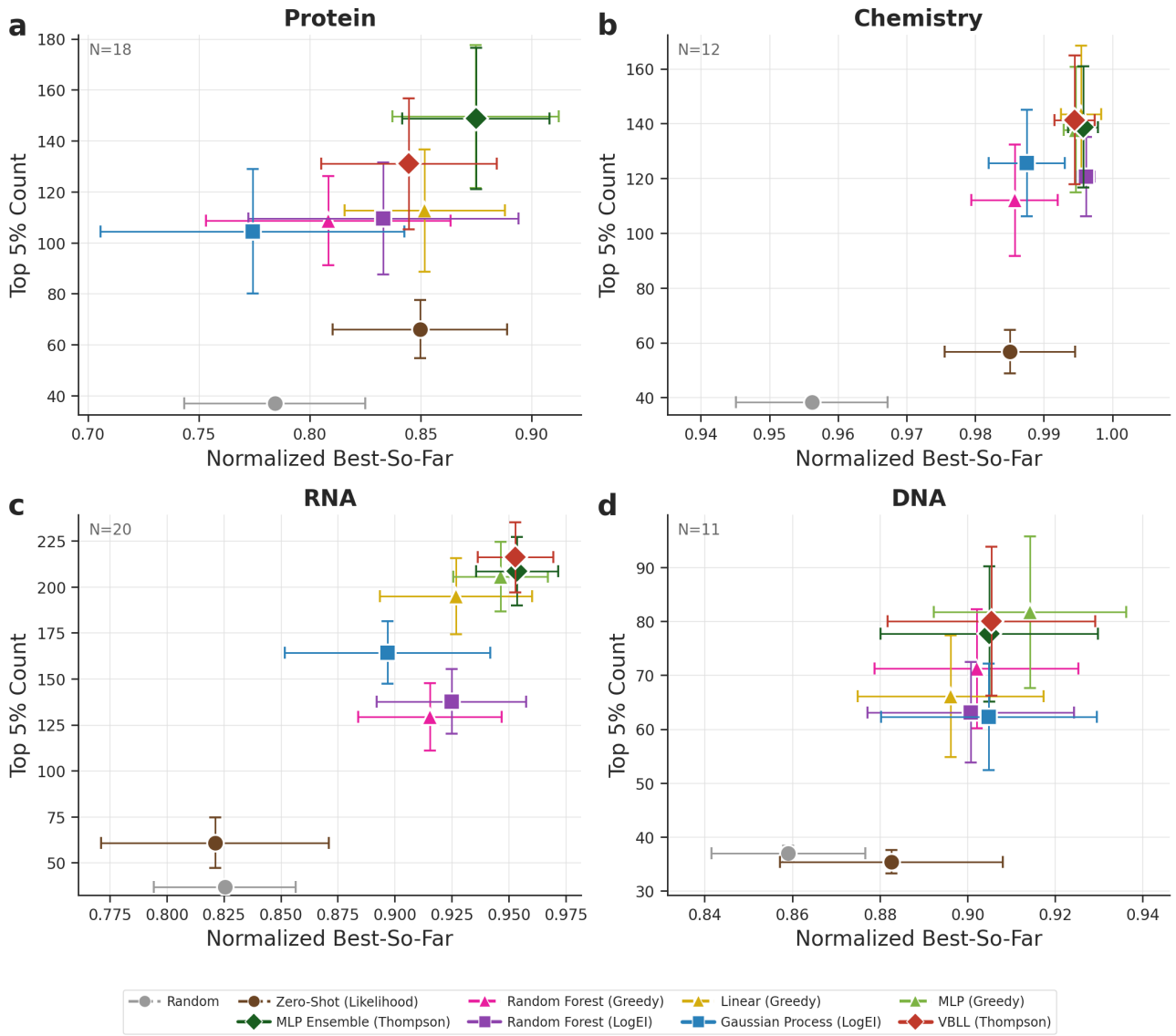

Figure 14. **Per-modality Pareto frontier at k=96, T=8.** Same axes as Figure 13 split by modality. `VBLL` leads on chemistry, RNA, and DNA; `MLP_Ensemble` and `MLP_Greedy` are most competitive on protein, mirroring the sequential per-modality picture.

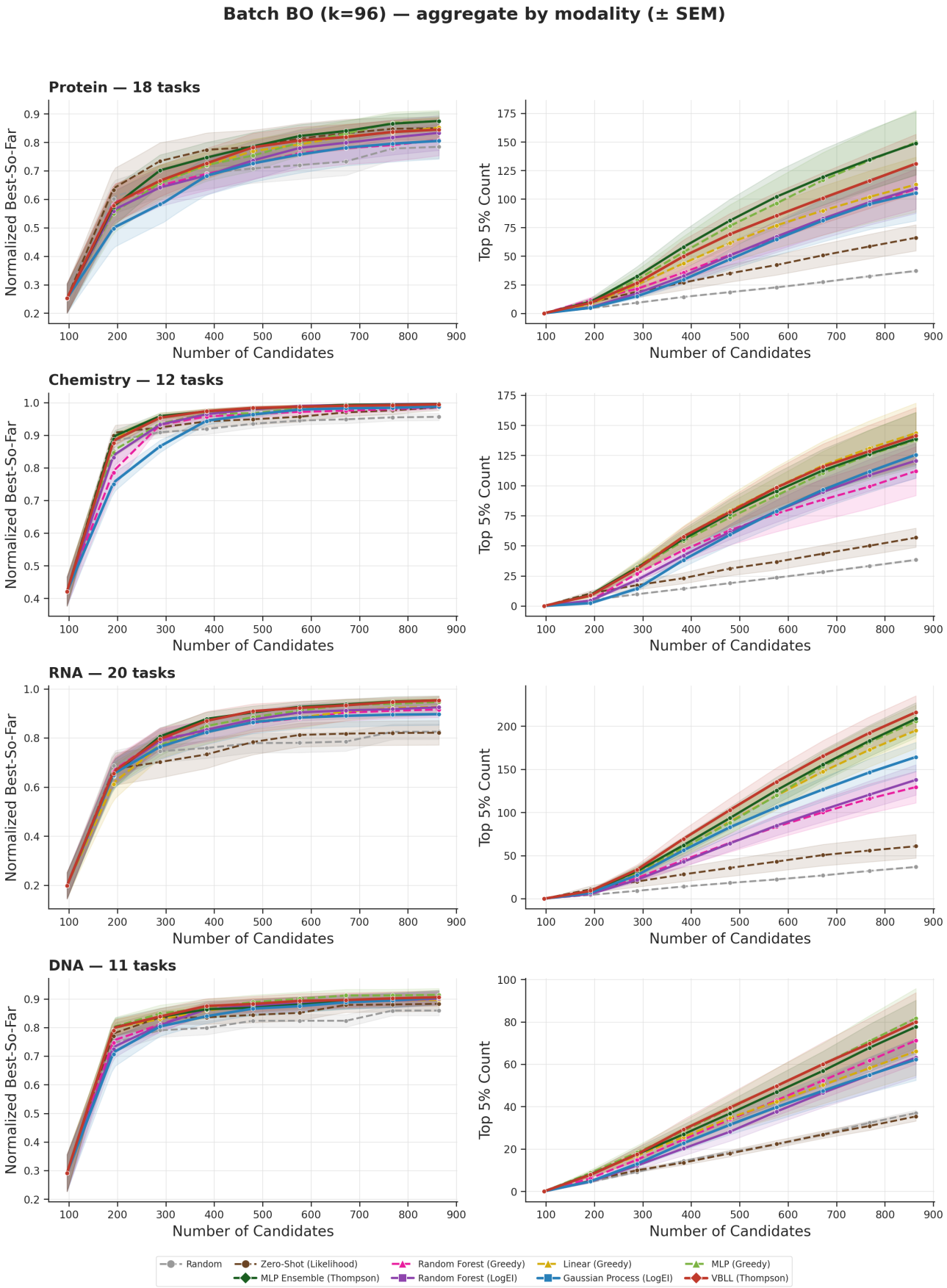

Figure 15. **Round-by-round batch curves at fixed total budget.** NBSF (top) and Top-5% count (bottom) vs. cumulative candidates measured for $k \in \{24, 48, 96, 192, 384\}$. Aggregated over 61 tasks, 9 methods, and 5 seeds. Bands are ± 1 SEM. Smaller k dominates throughout on Top-5% and converges with larger k on NBSF by end-of-budget.

I. Comparison to LLM

This appendix documents reproducibility details for the LLM-ICL baseline used in §3.3. None of the choices below are load-bearing for the paper's claim, but reviewers and reimplementors will want them.

Prompt structure. The prompt is fully pool-stable: the candidate pool is rendered once, sorted by candidate ID, and never re-permuted. This is critical for serving cost. vLLM's prefix cache survives round to round, only the appended selection row needs cold prefill at each step, and round 2 wall-clock is roughly an order of magnitude faster than round 1. The prompt contains six sections in fixed order:

- `## Pool`: stable, listing $(id,$ sequence$)$ for unobserved candidates and never the fitness label (the central no-leakage invariant).
- `## Initial Screen`: all 96 below-median initial candidates with their measured fitness, sorted by the task metric.
- `## Selections`: append-only table of $(id,$ sequence, fitness$)$ tuples for rounds 1 through $t-1$.
- `## Tested IDs`: append-only list, used by the model to filter its own candidate set.
- `## Progress`: summary of best-so-far, the top-5 selections, and the number of remaining candidates.
- `## Task`: defines the two BO objectives (NBSF and top-5% count) and the balanced exploit/explore framing.

Coverage-aware framing. The `## Task` block defines the two objectives and instructs the model in language that does not bias toward either strategy: *"Both exploitation (refining within a known high-fitness region) and exploration (probing a distinct cluster) can be productive [. . .] balance the two based on what the data suggests."* The model is then required to emit a brief `## diversity_check ##` block stating which strategy it has chosen and why, followed by `## selected_id ##` N. The diversity-check block is a forcing function for the model to commit to a stated strategy each round, which in turn anchors its selection.

Contamination controls. The task-description block never names the source paper, the benchmark suite, the original authors, or the assay year, and identifies only the underlying biology in domain-generic terms: *"GB1 protein B_1-domain"*; *"MPRA for human enhancers in the SK-N-SH neuroblastoma line"*; *"synthetic 5′ UTR plus start-codon library driving an mRNA reporter"*; *"small-molecule agonists of PPARγ, a ligand-activated nuclear receptor"*. Specific receptor or library identities are retained because they are pre-existing public knowledge that any domain expert would also have, and excluding them would render the task uninterpretable to the model. We did not implement an additional purge of the pretraining corpus: the LLM may have seen the underlying datasets indirectly through derivative literature, which is the residual contamination concern noted in §3.3.

Strict parser. We use a strict parser that strips the `<think>...</think>` block and then the `## diversity_check ##` block before scanning for `## selected_id ##` N. If the delimiter is missing, the parser returns one of four failure flags (`no_delimiter`, `ambiguous`, `out_of_pool`, `tested`) rather than a silent integer match, and the round is logged as `parse_failure` for post-hoc audit. We do not use a "last integer in the response" fallback (which would silently match an integer in the diversity-check sentence or a position number in the reasoning). Failed rounds fall back to a uniform random pick from the untested pool, so parse failure does not bias the LLM toward any sequence region but does cap the upper bound on its top-5% recall. Per-task parse-success rates ranged from 84.6% (`gosai_sknsh`) to 92.2% (`ke_mrna`); detailed per-seed counts appear in Table 9.

Inference. Inference uses vLLM with `max_model_len`$=262{,}144$ tokens, `max_tokens`$=32{,}768$, FP8 KV cache, prefix caching enabled, and a single H100 GPU under FP8 weights (35B-parameter MoE, 3B active per token). Sampling follows the Qwen3.6-recommended thinking-mode settings: `temperature`$=1.0$, `top_p`$=0.95$, `top_k`$=20$, `presence_penalty`$=1.5$. A multi-seed wrapper loads model weights once and warms up DeepGEMM kernels once, then resets the prefix cache and BO state between seeds (no weight reload, no warmup repeat).

Task selection rationale. We selected one task per modality from the 61-task benchmark using three criteria: (i) the task is in our benchmark and has a competitive surrogate baseline (so any LLM gap reflects a hard regime, not a poorly-tuned baseline); (ii) the pool is small enough to fit in context with ≥ 25K reasoning tokens of headroom at round 100 under the empirical Qwen tokenizer; (iii) within (ii), we prefer the largest pool that fits, since the comparison is most informative when the LLM is given as much context as it can use. Long-protein and long-RNA tasks (over 400 residues or nucleotides)

overflow the 256K-token window once the pool exceeds a few hundred candidates and were dropped from the LLM-eligible cohort entirely. For chemistry (pparg, N=2,349) and RNA (ke_mrna, N=5,559) the entire dataset fits in context. For protein and DNA the full 10K-row datasets are too large, so we deterministically subsample (seed 12345, applied identically across methods on the LLM-side runs) to N=1,500 for gb1 and N=2,100 for gosai_sknsh.

Pool-size confound on top-5% count. The top-5% threshold is the 95th percentile of the full task dataset, computed identically for the LLM and the SciOpt baselines. For the matched-pool tasks (pparg, ke_mrna) both methods see the same candidates, so absolute count is directly comparable. For gb1 and gosai_sknsh the LLM sees a subsample and SciOpt sees the full dataset, so the LLM has fewer top-5% candidates available in its visible pool (Table 8). On these two tasks the absolute-count comparison mechanically caps the LLM's top-5% count by roughly 5×, so we additionally check the recall rate (hits divided by available top-5% in each method's pool); the LLM still trails on coverage rate everywhere except gb1.

Table 8. Pool sizes and top-5% availability per task per method.

	LLM-ICL		SciOpt	
Task	Pool Size	Top-5% avail	Pool Size	Top-5% avail
gb1	1,500	79	10,000	500
pparg	2,349	131	2,349	131
ke_mrna	5,559	277	5,559	277
gosai_sknsh	2,100	98	10,000	500

Per-seed results. Table 9 reports NBSF, top-5% count, and parse-failure count for each (task, seed) at round 100. Run-to-run variance is large for LLM-ICL on every task, often comparable to or exceeding the mean. On gb1 two of five seeds reach NBSF 0.254 and 0.358 while the other three reach 1.000, reflecting the mode-collapse pattern discussed in §3.3.

Table 9. Per-seed final-round (round 100) results for LLM-ICL on the four paper-slate tasks. T5 = top-5% count; PF = parse-failure count out of 100 rounds.

	gb1			pparg			ke_mrna			gosai_sknsh		
Seed	NBSF	T5	PF	NBSF	T5	PF	NBSF	T5	PF	NBSF	T5	PF
1	1.000	28	10	0.830	18	10	0.643	4	6	0.420	5	11
2	1.000	18	9	0.780	9	10	0.752	3	5	1.000	3	18
3	0.254	5	12	0.774	6	24	0.894	41	8	0.420	7	12
4	1.000	25	17	0.934	29	6	0.626	1	15	0.539	8	16
5	0.358	8	11	1.000	37	8	0.685	4	5	0.413	4	20
Mean	0.722	16.8	11.8	0.864	19.8	11.6	0.720	10.6	7.8	0.558	5.4	15.4
S.d.	0.382	10.1	3.1	0.100	13.1	7.4	0.109	17.0	4.1	0.252	2.1	3.8

Random-baseline note for unmatched-pool columns. For gb1 and gosai_sknsh, the random baseline in Figure 4 is computed on the SciOpt full pool (N=10,000), not the LLM's subsample. A random search on the LLM's pool would have a slightly different baseline NBSF, since both pools share the same biology but differ in absolute fitness range. For the matched-pool tasks (pparg, ke_mrna) the random baseline is directly comparable to the LLM.

Reasoning examples. Figures 16–19 display one structured response per task from rounds where LLM-ICL selected a global top-5% candidate. Each example contains the model's full ## reasoning ##, ## diversity_check ##, and ## selected_id ## blocks (the <think> chain-of-thought is omitted since the model emits an explicit final reasoning section). The reasoning is consistently grounded in domain biology (FMN-binding J-helix in GB1; tetralin-carboxylate scaffolds with halogen substitution for PPARγ; conserved core motifs with 5′ UTR variation for ke_mrna; high-GC neuronal-enhancer architecture for gosai_sknsh), which is consistent with the paradigm-appeal claim in Liu et al. 2024 (Liu et al., 2024). Despite this reasoning quality, the run-level metrics in Table 9 show that grounded chains-of-thought do not translate to surrogate-competitive performance under tight measurement budgets.

LLM-ICL on GB1 — One Round of In-Context Learning

Qwen3.6-35B-A3B-FP8 · round 38 of 100 · selected ID 737 → fitness 4.916, top-5 %

STEP 1 · Optimization setup

```
You are an expert protein biochemist performing iterative experimental
   optimization.

Your goal: discover high-performing amino acid sequence variants. We
   score you on TWO objectives, and both matter:
  1. **Best-so-far** — the single highest Fitness you find across all
     selections.
  2. **Coverage** — how many of the top 5% of the full library you
     discover (top_5_count).

Both exploitation (refining within a known high-fitness region) and
   exploration (probing a distinct cluster) can be productive.
   Exploitation is most useful when a high-fitness pattern in your
   selections hasn't been fully mapped yet; exploration is most useful
   when your top selections cluster narrowly and other high-fitness
   regions likely remain undiscovered. Balance the two based on what
   the data suggests. After each selection you will receive the
   measured fitness. You have a budget of 100 rounds.

About this experiment:
GB1 protein B1-domain. Fitness measures binding affinity to IgG-Fc.

About the data format:
Sequences use amino acid single-letter code. Key properties:
   hydrophobic (A, V, L, I, M, F, W, P), polar (S, T, Y, N, Q),
   charged (D, E, K, R, H), special (C disulfide, G flexible, P
   rigid).

Reasoning budget guidance: keep your private reasoning concise and
   decision-oriented. **Never re-write a candidate's full sequence in
   your reasoning — refer to candidates ONLY by their integer ID
   number.** The pool is shown above; you can locate any ID instantly.
   Verbose sequence enumeration in `<think>` wastes the response
   budget and risks truncating before the final `## selected_id ##`
   block. Aim for ≤2-3 K tokens of reasoning before committing.
```

STEP 2 · Iteration state (abridged for display)

```
## Pool (1404 total candidates, sorted by ID)


## Initial Screen (Round 0 — 96 candidates from the low-fitness region, sorted by Fitness)
| ID | Sequence | Fitness |
|----|----------------------------------------|---------|
| 57 | MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGRFKEWTYDDATK…| 0.000 |
| 14 | MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGSTFEWTYDDATK…| 0.000 |
| 18 | MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGCFNEWTYDDATK…| 0.000 |
| 73 | MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGKMREWTYDDATK…| 0.000 |
| 94 | MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGWPWEWTYDDATK…| 0.000 |
Round 0 distribution: mean=0.000, median=0.000, min=0.000, max=0.000, stdev=0.000.


## Selections (Rounds 1–38)
| Round | ID | Sequence | Fitness |
|-------|-----|----------------------------------------|---------|
| 2 | 200 | MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGHLQEWTYDDATK…| 0.000 |
| 3 | 106 | MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGPWWEWTYDDATK…| 0.000 |
| 4 | 1150 | MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGCCLEWTYDDATK…| 0.000 |
  [... +33 more selections ...]


## Tested IDs (37 — exclude from your selection)
96, 100, 106, 111, 130, 141, 143, 200, ... (37 total)


## Progress
Round: 38 / 100
Untested candidates remaining: 1367
Best Fitness so far: 1.000 (ID 833, Round 22)
Selections so far: n=36, mean=0.049, stdev=0.179
  Top-5 of your selections (Top-5 table elided for figure; see ## Selections above) →


## Task
Analyze the patterns in your selections (and the top of Round 0). Consider which sequence features
  correlate with high Fitness.

**Select ONE candidate ID from `## Pool` whose ID is NOT in `## Tested IDs`.** That candidate should
  be:
  1. **Likely high-Fitness** — features consistent with your top selections.
  2. **A deliberate strategic choice** — either exploitation (reinforcing a productive pattern in
     your top-5 selections to refine an under-mapped region) or exploration (probing a
     sequence-distinct region of sequence space).

Choose exploitation when a clear high-fitness pattern is emerging and there is room to refine it.
  Choose exploration when your top selections cluster narrowly and additional similar picks would
  yield diminishing returns. Both objectives — best-so-far AND top_5_count — benefit from the right
  balance.

  [response format spec — see Step 3]
```

STEP 3 · Model reasoning + selection

reasoning

Analyzing the top performers, I notice a key structural distinction: **ID 833** (fitness 1.0) uniquely has the pattern `ANDNGRAEEWTYDDATKTFTHTE` — specifically, it carries a rare `KTFTH` variant (position 37 shows H instead of the more common E/T/N). Meanwhile, many moderately performing sequences share `KTFTE` or `KTFTH` but differ in their central motif.

The critical variable region spans approximately positions 31-34 (`XXE WTY`). My highest performers cluster around motifs like `RAEE`, `FWLE`, `SDDE`, `VLQE`. These all contain polar or charged side chains (E, D) which likely contribute to IgG-Fc binding surface complementarity.

My current coverage is poor — I've only tested 37 of 1404 candidates despite needing to find top 5% (~70 sequences). With one sequence hitting 1.0 and another at 0.42, there's clearly under-mapped space. Rather than exploit the narrow `KTFTH` region further, I should probe nearby untested motifs with similar physicochemical properties.

I'll select **ID 737** (ANDNGWDNEWTYDDATKTFTPTE) — this carries `WDNE` motif (aromatic W + polar D/N), which structurally mirrors features of my top performers. While its `KTFTP` variant differs slightly from the best performer's `KTFTH`, the motif composition in the critical 31-34 region is promisingly similar to the successful `RAEE`/`FWLE`/`SDDE` classes.

diversity_check

This pick is **exploration** — probing a different motif region (WDNE vs RAEE/FWLE/SDDE) while maintaining favorable physicochemical properties (polar/aromatic character), targeting a largely unexplored cluster in sequence space rather than reinforcing the narrow KTFTH pattern of our single best hit.

selected_id

737

STEP 4 · Measured fitness returned to the model (next round's prompt)

```
After Step 3, the BO loop measures the fitness of the chosen variant
and concatenates the measurement onto the running ## Selections ## block at the start
of round 39:

    | Round | ID    | Sequence                                                |  Fitness |
    |-------|-------|---------------------------------------------------------|----------|
    |    38 |   737 | MQYKLILNGKTLKGETTTEAVDAATAEKVFKQYANDNGRLQEWTYDDATKTFTIT…  |   0.080  |   ← TOP-5%

That single row is the entire feedback signal — no surrogate, no acquisition function. The model's
only learning channel is re-reading its own past selections and their measured fitness.
```

Figure 16. **Reasoning example: GB1 (Protein).** Round 38 of seed 2 on the `gb1_1500` pool. LLM-ICL selects ID 737, fitness 4.916 (top-5%). The model identifies physicochemical motif clusters (`RAEE` / `FWLE` / `SDDE`) shared across high-fitness sequences and explicitly chooses to probe a parallel `WDNE` motif as exploration of an under-mapped region of sequence space.

LLM-ICL on PPARγ — One Round of In-Context Learning

Qwen3.6-35B-A3B-FP8 · round 73 of 100 · selected ID 1079 → pIC_{50} 1.745, global max

STEP 1 · Optimization setup

```
You are an expert medicinal chemist performing iterative experimental
   optimization.

Your goal: discover high-performing molecule variants. We score you on
   TWO objectives, and both matter:
  1. **Best-so-far** — the single highest pIC50 (PPARγ) you find
     across all selections.
  2. **Coverage** — how many of the top 5% of the full library you
     discover (top_5_count).

Both exploitation (refining within a known high-fitness region) and
   exploration (probing a distinct cluster) can be productive.
   Exploitation is most useful when a high-fitness pattern in your
   selections hasn't been fully mapped yet; exploration is most useful
   when your top selections cluster narrowly and other high-fitness
   regions likely remain undiscovered. Balance the two based on what
   the data suggests. After each selection you will receive the
   measured fitness. You have a budget of 100 rounds.

About this experiment:
Optimization of small-molecule agonists of PPARγ
   (peroxisome-proliferator activated receptor γ), a ligand-activated
   nuclear receptor. Fitness measures pIC50 (negative log of
   half-maximal inhibitory concentration). A carboxylate or acidic
   head paired with a lipophilic tail is the canonical PPARγ-binding
   motif; halogen substitutions and substitution-position patterns
   modulate affinity.

About the data format:
Molecules are represented as SMILES strings. Aromatic rings use
   lowercase (c1ccccc1), branches in parentheses, double bonds with =.
   Functional groups, ring systems, and electronic effects drive
   properties.

Reasoning budget guidance: keep your private reasoning concise and
   decision-oriented. **Never re-write a candidate's full sequence in
   your reasoning — refer to candidates ONLY by their integer ID
   number.** The pool is shown above; you can locate any ID instantly.
   Verbose sequence enumeration in `<think>` wastes the response
   budget and risks truncating before the final `## selected_id ##`
   block. Aim for ≤2-3 K tokens of reasoning before committing.
```

STEP 2 · Iteration state (abridged for display)

```
## Pool (2253 total candidates, sorted by ID)


## Initial Screen (Round 0 — 96 candidates from the low-fitness region, sorted by pIC50 (PPARγ))
| ID | SMILES | pIC50 (PPARγ) |
|----|---------------------------------------|---------|
| 68 | COC1Oc2cc(O)c3c(c2C(=O)N1C(=O)NCc1ccco1)OC1=CC(O)=…| 0.334 |
| 57 | CCC#CCOc1cc(COc2ccc(Cl)cc2C)ccc1Sc1ccc(OCC(=O)O)c2…| 0.328 |
| 59 | Cn1c(COc2cccc(C(=O)O)c2)nc2ccc(Oc3cc(Cl)ccn3)cc21 | 0.328 |
| 44 | CCCOc1ccc(CC(CC)C(=O)O)cc1CNC(=O)c1ccc(C(F)(F)F)cc…| 0.324 |
| 73 | CCCCCCC(Sc1nc2ccc(NS(=O)(=O)c3ccc(C(C)(C)C)cc3)cc2…| 0.319 |
Round 0 distribution: mean=0.239, median=0.252, min=0.092, max=0.334, stdev=0.060.


## Selections (Rounds 1–73)
| Round | ID | SMILES | pIC50 (PPARγ) |
|-------|-----|---------------------------------------|---------|
| 2 | 966 | O=C(O)[C@H](Cc1ccccc1)Oc1ccc(-c2ccccc2C(F)(F)F)cc1 | 0.118 |
| 3 | 514 | O=C(O)CCc1ccc(OCCN(Cc2ccc(C(F)(F)F)cc2)c2nc3ccccc3…| 0.297 |
| 4 | 1789 | Cc1oc(-c2ccc(Cl)cc2)nc1CCOc1ccccc(C[C@@H]2C(=O)N(c3…| 0.212 |
  [... +68 more selections ...]


## Tested IDs (72 — exclude from your selection)
106, 109, 118, 127, 136, 152, 168, 171, ... (72 total)


## Progress
Round: 73 / 100
Untested candidates remaining: 2181
Best pIC50 (PPARγ) so far: 0.741 (ID 1617, Round 71)
Selections so far: n=71, mean=0.324, stdev=0.162
  Top-5 of your selections (Top-5 table elided for figure; see ## Selections above) →


## Task
Analyze the patterns in your selections (and the top of Round 0). Consider which smiles features
  correlate with high pIC50 (PPARγ).

**Select ONE candidate ID from `## Pool` whose ID is NOT in `## Tested IDs`.** That candidate should
  be:
  1. **Likely high-pIC50 (PPARγ)** — features consistent with your top selections.
  2. **A deliberate strategic choice** — either exploitation (reinforcing a productive pattern in
     your top-5 selections to refine an under-mapped region) or exploration (probing a
     sequence-distinct region of molecular space).

Choose exploitation when a clear high-fitness pattern is emerging and there is room to refine it.
  Choose exploration when your top selections cluster narrowly and additional similar picks would
  yield diminishing returns. Both objectives — best-so-far AND top_5_count — benefit from the right
  balance.

  [response format spec — see Step 3]
```

STEP 3 · Model reasoning + selection

reasoning

My top performers share two key structural motifs:

1. **Fused bicyclic scaffold with carboxylic acid** — IDs 1367 (0.671), 1714 (0.626), 1453 (0.576) all follow the pattern `CCC(C(=O)O)C1CCc2cc(...)ccc21`: a tetralin-like fused system with a 3-methylbutanoic acid side chain, connected via an ether linker (OCC) to a heteroaromatic ring (oxazole/thiazole) bearing a distal phenyl substituent.

2. **Variation in the distal phenyl group**: unsubstituted phenyl gave 0.671 (ID 1367), unsubstituted ring gave 0.626 (ID 1714), CF_3-phenyl gave 0.576 (ID 1453). The trend suggests that moderate lipophilicity/electronic modulation on the distal group is beneficial but excessive hydrophobicity (CF_3) may penalize.

ID 1079 (`CCC(C(=O)O)C1CCc2cc(OCCc3nc(-c4ccc(Cl)cc4Cl)oc3C)ccc21`) is an **exploitation** pick using the identical scaffold, but substituting the distal phenyl with a dichloro-phenyl. Halogen substitutions are known to fine-tune PPARγ affinity through both electronic effects and van der Waals contacts in the ligand-binding pocket. The dichloro variant probes whether increased lipophilicity/bulk beyond phenyl improves affinity without crossing into the penalty region observed with CF_3.

diversity_check

Exploitation: This candidate uses the exact same fused-bicyclic-tetralin scaffold as my top-3 performers, varying only the distal phenyl substituent from H/C_6H_5 to dichloro. This refines the under-mapped parameter space of halogen substitution within the proven high-fitness pattern.

selected_id

1079

STEP 4 · Measured fitness returned to the model (next round's prompt)

```
After Step 3, the BO loop measures the pic50 of the chosen variant
and concatenates the measurement onto the running ## Selections ## block at the start
of round 74:

    | Round | ID    | Sequence                                                |  Fitness |
    |-------|-------|---------------------------------------------------------|----------|
    |    73 |  1079 | CCC(C(=O)O)C1CCc2cc(OCCc3nc(-c4ccc(Cl)cc4Cl)oc3C)ccc21    |   0.437  |   ← GLOBAL MAX

That single row is the entire feedback signal — no surrogate, no acquisition function. The model's
only learning channel is re-reading its own past selections and their measured fitness.
```

Figure 17. **Reasoning example: PPARγ Binding (Chemistry).** Round 73 of seed 1 on the `pparg_full` pool. LLM-ICL selects ID 1079, pIC_{50} 1.745 (global maximum). The model identifies a fused-bicyclic-tetralin scaffold with a carboxylate head shared by its top performers, and exploits the under-mapped halogen-substitution dimension on the distal aromatic group.

LLM-ICL on ke_mrna — One Round of In-Context Learning

Qwen3.6-35B-A3B-FP8 · round 9 of 100 · selected ID 4244 → DMS score 5.21, top-5 %

STEP 1 · Optimization setup

```
You are an expert RNA biologist performing iterative experimental
   optimization.

Your goal: discover high-performing RNA sequence variants. We score
   you on TWO objectives, and both matter:
  1. **Best-so-far** — the single highest mRNA Expression you find
     across all selections.
  2. **Coverage** — how many of the top 5% of the full library you
     discover (top_5_count).

Both exploitation (refining within a known high-fitness region) and
   exploration (probing a distinct cluster) can be productive.
   Exploitation is most useful when a high-fitness pattern in your
   selections hasn't been fully mapped yet; exploration is most useful
   when your top selections cluster narrowly and other high-fitness
   regions likely remain undiscovered. Balance the two based on what
   the data suggests. After each selection you will receive the
   measured fitness. You have a budget of 100 rounds.

About this experiment:
Optimization of a synthetic 5' UTR + start-codon context library
   driving an mRNA reporter. Fitness measures relative protein output
   (translation efficiency × mRNA stability). Kozak-context strength,
   5'-end secondary structure, upstream open reading frames (uORFs),
   and start-codon accessibility modulate expression.

About the data format:
RNA sequences use nucleotide single-letter code (A, C, G, U). Base
   pairing (A-U, G-C, G-U wobble) drives secondary structure. Sequence
   context affects folding, stability, and catalytic activity.

Reasoning budget guidance: keep your private reasoning concise and
   decision-oriented. **Never re-write a candidate's full sequence in
   your reasoning — refer to candidates ONLY by their integer ID
   number.** The pool is shown above; you can locate any ID instantly.
   Verbose sequence enumeration in `<think>` wastes the response
   budget and risks truncating before the final `## selected_id ##`
   block. Aim for ≤2-3 K tokens of reasoning before committing.
```

STEP 2 · Iteration state (abridged for display)

```
## Pool (5463 total candidates, sorted by ID)


## Initial Screen (Round 0 — 96 candidates from the low-fitness region, sorted by mRNA Expression)
| ID | Sequence | mRNA Expression |
|-----|----------------------------------------|---------|
| 30 | AGUUGAAGAAGGGAGCUCCAGCACAGUGAAAUGGACCUAAGGGCAGAGCA…| 0.065 |
| 60 | AGUUGAAGAAGGGAGCUACAGCACAGUGAAAUGGACAGAAGGGCAGAGCA…| 0.063 |
| 89 | AGUUGAAGAAGGGAGCUCACGCACAGUGAAAUGGACAGAAGGGCAGAGCA…| 0.062 |
| 0 | AGUUGAGUACGGGAGCUCCAGCACAGUGAAAUGGACAGAAGGGCAGAGCA…| 0.062 |
| 43 | AGUUCGCGCCGUAAGCUCCAGCACAGUGAAAUGGACAGAAGGGCAGAGCA…| 0.062 |
Round 0 distribution: mean=0.014, median=0.006, min=0.000, max=0.065, stdev=0.018.


## Selections (Rounds 1–9)
| Round | ID | Sequence | mRNA Expression |
|-------|-----|----------------------------------------|---------|
| 2 | 2443 | AGUUGAAGAAGGGAGCUCCAGCACAGUGAAAUGGACAGAAGGGCAAGGCA…| 0.125 |
| 3 | 3967 | AGUUACCACCGGGAGCUCCAGCACAGUGAAAUGGACAGAAGGGCAGUCCA…| 0.038 |
| 4 | 4242 | AGUUGAACACGGGAGCUCCAGCACAGUGAAAUGGACAGAAGGGCAGAGCA…| 0.245 |
  [... +4 more selections ...]


## Tested IDs (8 — exclude from your selection)
2443, 3317, 3967, 4242, 4243, 4245, 4247, 4289


## Progress
Round: 9 / 100
Untested candidates remaining: 5455
Best mRNA Expression so far: 0.305 (ID 4243, Round 6)
Selections so far: n=7, mean=0.229, stdev=0.098
  Top-5 of your selections (Top-5 table elided for figure; see ## Selections above) →


## Task
Analyze the patterns in your selections (and the top of Round 0). Consider which sequence features
  correlate with high mRNA Expression.

**Select ONE candidate ID from `## Pool` whose ID is NOT in `## Tested IDs`.** That candidate should
  be:
  1. **Likely high-mRNA Expression** — features consistent with your top selections.
  2. **A deliberate strategic choice** — either exploitation (reinforcing a productive pattern in
    your top-5 selections to refine an under-mapped region) or exploration (probing a
    sequence-distinct region of sequence space).

Choose exploitation when a clear high-fitness pattern is emerging and there is room to refine it.
  Choose exploration when your top selections cluster narrowly and additional similar picks would
  yield diminishing returns. Both objectives — best-so-far AND top_5_count — benefit from the right
  balance.

  [response format spec — see Step 3]
```

STEP 3 · Model reasoning + selection

reasoning

Analysis of the top 5 performers (IDs 4242–4289) reveals a highly consistent core motif: `CUCCAGCACAGUGAAAUGGACAGAAGGGCAGAGCAA`. All five share this exact backbone. They differ primarily in their 5' termini, with successful variants starting with `AGUU` followed by diverse intermediate sequences (e.g., `UGAAGAGCGG`, `CUUCUCGGG`, `UGAACACGG`). The consensus indicates that the core sequence is essential for high expression, while the 5' region can tolerate significant variation—likely affecting ribosome accessibility, secondary structure, or initiation efficiency.

Given only 7 total selections and a clear signal emerging around this conserved core, exploitation is the right move. By selecting another candidate that preserves the winning core but varies the 5' region, I can map the neighborhood of this high-fitness peak more thoroughly. Candidate 4244 matches the core perfectly (`CUCCAGCACAGUGAAAUGGACAGAAGGGCAGAGCAA`) and has a GC-rich 5' start (`AGUUGACGAAGGG`) distinct from the 5' variants already tested in rounds 2–8. This targeted exploitation should either identify a better local optimum or confirm the boundaries of the current peak.

diversity_check

Exploitation — selecting a candidate with the same proven core motif but a different 5' sequence to refine the mapped high-fitness region before broader exploration.

selected_id

4244

STEP 4 · Measured fitness returned to the model (next round's prompt)

```
After Step 3, the BO loop measures the dms score of the chosen variant
and concatenates the measurement onto the running ## Selections ## block at the start
of round 10:

    | Round | ID    | Sequence                                                  |  Fitness |
    |-------|-------|-----------------------------------------------------------|----------|
    |     9 |  4244 | AGUUACCACCGGGAGCUCCAGCACAGUGAAAUGGACAACAGGGCAGAGCAA       |   0.298  |   ← TOP-5%

That single row is the entire feedback signal — no surrogate, no acquisition function. The model's
only learning channel is re-reading its own past selections and their measured fitness.
```

Figure 18. **Reasoning example: mRNA Splicing (RNA).** Round 9 of seed 3 on the `ke_mrna_full` pool. LLM-ICL selects ID 4244, DMS score 5.21 (top-5%). The model identifies a conserved 36-nt core motif (`CUCCAGCACAGUGAAAUGGACAGAAGGGCAGAGCAA`) preserved across the top performers and exploits within the 5′-end variation neighborhood of the existing high-fitness peak.

LLM-ICL on gosai_sknsh — One Round of In-Context Learning

Qwen3.6-35B-A3B-FP8 · round 67 of 100 · selected ID 1106 → STARR-seq 2.93, top-5 %

STEP 1 · Optimization setup

```
You are an expert regulatory genomicist performing iterative
   experimental optimization.

Your goal: discover high-performing DNA sequence variants. We score
   you on TWO objectives, and both matter:
  1. **Best-so-far** — the single highest Enhancer Activity (SK-N-SH)
     you find across all selections.
  2. **Coverage** — how many of the top 5% of the full library you
     discover (top_5_count).

Both exploitation (refining within a known high-fitness region) and
   exploration (probing a distinct cluster) can be productive.
   Exploitation is most useful when a high-fitness pattern in your
   selections hasn't been fully mapped yet; exploration is most useful
   when your top selections cluster narrowly and other high-fitness
   regions likely remain undiscovered. Balance the two based on what
   the data suggests. After each selection you will receive the
   measured fitness. You have a budget of 100 rounds.

About this experiment:
MPRA for human enhancers in SK-N-SH neuroblastoma cells. Fitness
   measures STARR-seq signal. NEUROD, ASCL1, and POU3F2 motifs are
   characteristic of neuronal enhancers.

About the data format:
DNA sequences use nucleotide single-letter code (A, T, C, G). Base
   pairing (A-T, G-C) drives structure. Motifs like TATA boxes, GC
   content, and transcription factor binding sites affect function.

Reasoning budget guidance: keep your private reasoning concise and
   decision-oriented. **Never re-write a candidate's full sequence in
   your reasoning — refer to candidates ONLY by their integer ID
   number.** The pool is shown above; you can locate any ID instantly.
   Verbose sequence enumeration in `<think>` wastes the response
   budget and risks truncating before the final `## selected_id ##`
   block. Aim for ≤2-3 K tokens of reasoning before committing.
```

STEP 2 · Iteration state (abridged for display)

```
## Pool (2004 total candidates, sorted by ID)


## Initial Screen (Round 0 — 96 candidates from the low-fitness region, sorted by Enhancer Activity (SK-N-SH))
| ID | Sequence | Enhancer Activity (SK-N-SH) |
|-----|----------------------------------------|---------|
| 93 | CTCCTGCACACAGAAGGACACTACTTGTCATCCTTATGGATTCATGAGTG…| 0.050 |
| 85 | GGGTGTGGGTGGAGGCTGGGTATGGGTGGCTGGGGTGGGGTGGAGCACAC…| 0.047 |
| 24 | AAATTAGCCGGGCGTGGTGGCGGGCGCCTGTAGTCCCAGCTACTTGGGAG…| 0.046 |
| 88 | TTTTTTTTTTTTTAATTAGCCAAGTAAGGCCAGGCATGGTGGCTCATGCC…| 0.046 |
| 84 | CTAACCCGCTACCCCATTGGTGGCGTCCGGCGGCGCGGCTGCTGTTATTT…| 0.046 |
Round 0 distribution: mean=0.026, median=0.026, min=0.000, max=0.050, stdev=0.013.


## Selections (Rounds 1–67)
| Round | ID | Sequence | Enhancer Activity (SK-N-SH) |
|-------|-----|----------------------------------------|---------|
| 2 | 122 | AAATTCAAATGGATGCTATTCTGATGAAACTTGGGCTGGATGGGGCACAG…| 0.011 |
| 3 | 103 | GACTTATAGGTGAAAGAAACTCATCTCCAGGTGAGACTTTGGACATGGGA…| 0.004 |
| 4 | 111 | AGGATTCATATACAGTTATTCTAGACCCAGGGCTTATATCTATGATGCCA…| 0.002 |
  [... +62 more selections ...]


## Tested IDs (66 — exclude from your selection)
101, 103, 111, 112, 113, 122, 151, 242, ... (66 total)


## Progress
Round: 67 / 100
Untested candidates remaining: 1938
Best Enhancer Activity (SK-N-SH) so far: 0.420 (ID 1207, Round 25)
Selections so far: n=65, mean=0.108, stdev=0.092
  Top-5 of your selections (Top-5 table elided for figure; see ## Selections above) →


## Task
Analyze the patterns in your selections (and the top of Round 0). Consider which sequence features
  correlate with high Enhancer Activity (SK-N-SH).

**Select ONE candidate ID from `## Pool` whose ID is NOT in `## Tested IDs`.** That candidate should
  be:
  1. **Likely high-Enhancer Activity (SK-N-SH)** — features consistent with your top selections.
  2. **A deliberate strategic choice** — either exploitation (reinforcing a productive pattern in
    your top-5 selections to refine an under-mapped region) or exploration (probing a
    sequence-distinct region of sequence space).

Choose exploitation when a clear high-fitness pattern is emerging and there is room to refine it.
  Choose exploration when your top selections cluster narrowly and additional similar picks would
  yield diminishing returns. Both objectives — best-so-far AND top_5_count — benefit from the right
  balance.

  [response format spec — see Step 3]
```

STEP 3 · Model reasoning + selection

reasoning

Top performers (IDs 1207, 1935, 825, 2010, 1398) consistently feature high GC content, dense purine/pyrimidine tracts, and shared regulatory cores (e.g., `GGGAGGCAGAGGTTGCAGTGAG`, `CACACAC`). These motifs strongly align with neuronal enhancer architecture expected in SK-N-SH cells. With 33 rounds remaining, exploitation is optimal: refining within this high-fitness sequence neighborhood can reveal additive context effects and potentially push activity past 0.420. ID 1106 preserves the exact high-value core modules seen in my top-5 while presenting novel upstream/downstream flanks, making it a precise target for performance calibration.

diversity_check

Exploitation. This pick directly targets the high-GC/multimeric enhancer pattern dominating the top-5, aiming to refine motif spacing and flanking context rather than branching into a divergent sequence regime.

selected_id

1106

STEP 4 · Measured fitness returned to the model (next round's prompt)

After Step 3, the BO loop measures the starr-seq of the chosen variant and concatenates the measurement onto the running ## Selections ## block at the start of round 68:

```
| Round | ID    | Sequence                                                  |  Fitness |
|-------|-------|-----------------------------------------------------------|----------|
|    67 |  1106 | GCTAGGCGTGGTGGTGCATGCCTGTAATCCCAGCTACTCGGGAGGCTGAGGTAGG…  |   0.102  |   ← TOP-5%
```

That single row is the entire feedback signal — no surrogate, no acquisition function. The model's only learning channel is re-reading its own past selections and their measured fitness.

Figure 19. **Reasoning example: Neuroblastoma CRE (DNA).** Round 67 of seed 1 on the `gosai_sknsh` subsample. LLM-ICL selects ID 1106, STARR-seq activity 2.93 (top-5%). The model identifies a high-GC architecture with shared regulatory cores (`GGGAGGCAGAGGTTGCAGTGAG`, `CACACAC`) characteristic of neuronal enhancers in the SK-N-SH line, and exploits the motif-spacing neighborhood of its current top performers.

J. Comparison to SGPO

J.1. Experimental Setup

We evaluate on three combinatorially complete four-site libraries: GB1 (positions $39, 40, 41, 54$), TrpB4 (positions $34, 35, 78, 79$), and PhoQ (positions $50, 51, 54, 55$). For each seed, all methods share an identical cold-start set of 100 sequences drawn uniformly from the subset of the labeled pool whose true fitness lies below the 0.5 quantile, as described in the main text. The total labeling budget is $T = 1000$ sequences per seed, allocated as 100 initial points followed by 9 acquisition rounds of 100 candidates each. This protocol matches the iterative SGPO setting of (Yang et al., 2025) up to one additional acquisition round, included to align the budget across all methods compared in this work (their reported budget is $T = 900$). Each method is run for five independent seeds ($\{0, 1, 2, 3, 4\}$); shaded regions denote ± 1 sample standard deviation across seeds. We compare the ESM-C backbone under the MLP and VBLL SciOpt configurations, together with a uniform-random baseline.

J.2. SGPO Configuration

We adopt the configuration that (Yang et al., 2025) identify as strongest overall (their Section 5): a masked-diffusion language-model prior steered by decoupled annealing posterior sampling (DAPS), with Thompson sampling over an ensemble of value functions. At each round t, the value functions are retrained on the accumulated set of measured candidates $\mathcal{D}_{t-1}$. We use the bolded "ideal" DAPS hyperparameters from their Table A3 together with the unreported repository defaults (Table 10); the $5\times$ oversampling cap on per-round generation is the hard-coded default in their `iterativeBO.py`. For each dataset, the data-config `residues` field is set to the four combinatorially varied positions; this field is the upstream scope-restriction mechanism consumed by their `sampling/base.py` and used by DAPS to constrain the Metropolis–Hastings proposal scope, so the assignment is a configuration choice rather than a modification of the method.

Table 10. DAPS hyperparameters used in all SGPO experiments.

Parameter	Value	Description
$1/\beta$	128	Guidance strength (`alpha`)
K	50	Annealing steps (`num_steps`)
Euler steps	20	ODE solver steps (`ode_steps`)
MH steps	1,000	MH steps per annealing step
`max_dist`	1	Mutations per MH proposal
Iteration cap	$5\times$	Per-round oversampling limit

We observe substantial per-seed variability in the cumulative number of unique candidates selected, consistent with mode collapse in the diffusion prior. At the terminal round, the cumulative unique-candidate counts across the five seeds are

$$\begin{aligned} \text{GB1:} \quad & [333, 412], \quad 371 \pm 38; \\ \text{TrpB:} \quad & [771, 914], \quad 865 \pm 57; \\ \text{PhoQ:} \quad & [782, 952], \quad 863 \pm 76. \end{aligned}$$

For each task, the SGPO trajectory is summarized as the cross-seed mean of cumulative unique candidates, with a cross marker denoting the trajectory endpoint.

K. Contamination audit

The benchmark labels are wet-lab measurements that no pretraining corpus contains directly, so a literal (x, y) leak is implausible. The realistic risks are (a) *sequence-level* memorization, where the FM has seen the candidate x and assigns it an unusually informative likelihood, and (b) *narrative* memorization, where a chat-tuned LLM has read the assay paper and carries a prior over high-fitness motifs or scaffolds. Channel (a) is what `Zero_Shot` would exploit; channel (b) is what LLM-ICL would exploit.

FM-side: candidate pools vs. pretraining corpora. Each primary FM has been pretrained on a corpus that overlaps the candidate space at the wild-type level but not at the variant level. ESM-C 600M (EvolutionaryScale Team, 2024) sees UniRef parents but not the combinatorial DMS variants; RNA-FM (Chen et al., 2022) sees RNAcentral but not the synthetic

UTR / ribozyme libraries; NTv3-650M (Boshar et al., 2025) sees multi-species genomes but not the 200 bp MPRA libraries; ChemFM-1B (Cai et al., 2026) is trained on PubChem and so is the highest-overlap modality, with plausible exact-molecule overlap on the GOLLuM-derived chemistry tasks.

Implications for `Zero_Shot`. `Zero_Shot` ranks candidates by the FM's pseudo-likelihood without ever seeing labels (§B.2), so it is the surrogate most exposed to channel (a). If pretraining contamination were producing a useful fitness prior, `Zero_Shot` would be expected to dominate `Random`, especially on chemistry. Empirically it sits on or below the random-search frontier on every modality (Figure 2), consistent with the FM prior tracking sequence naturalness rather than the assay-specific target.

LLM-side: Qwen3.6 web-text overlap. Qwen3.6-35B-A3B-FP8 (Qwen Team, 2026) has a knowledge cutoff postdating every benchmark publication on our slate, so the model has plausibly read each source paper and derivative literature. We mitigate the most direct path via the prompt controls in App. I: the task block names only the underlying biology in domain-generic terms (never the source paper, benchmark, authors, or assay year), and the prompt never includes unobserved fitness labels. We did not perform a corpus purge, so this is a soft control. The empirical pattern is again inconsistent with a useful narrative leak: LLM-ICL trails learned surrogates on NBSF on every task, the `gb1` mode-collapse (two of five seeds end at NBSF 0.254 and 0.358) and high `gosai_sknsh` parse-failure rate (84.6% parse-success) match the prior-collapse failure mode documented for instruction-tuned LLMs under tight budgets (Rychert et al., 2025; Xiong et al., 2024; Tian et al., 2023), and the reasoning traces in App. I are domain-grounded but generic enough to come from background knowledge alone.

Direction of any residual leak. Residual contamination would inflate the leaked baseline, not the SciOpt result: a more contaminated FM helps `Zero_Shot`, which already trails the learned surrogates, and a more contaminated LLM helps LLM-ICL, which already trails them too. The main-paper gaps are therefore conservative with respect to leakage. Two honest caveats: (i) PubChem overlap may give every method on chemistry a uniform lift, so absolute NBSF on `pparg` should be read as "with PubChem-overlap allowed," and (ii) the LLM's `gb1` mode-collapse may be partly memorization-driven, which would make the favorable `gb1` LLM number a less informative comparison than the other three tasks.