TRACTABLE BAYESIAN OPTIMIZATION OVER COMBINATORIAL PROTEIN SPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein engineering requires optimizing sequences with limited experimental data over combinatorial spaces of up to 20^L variants. Gaussian processes (GPs) provide data-efficient surrogates, but optimizing acquisition functions over these spaces is computationally challenging. We introduce BaLiSq, a framework that addresses this bottleneck by recasting any kernel on single-mutant variants as a Bayesian linear regression prior over one-hot variant features. BaLiSq is provably equivalent to GP regression with the original kernel on single-mutant variants and extends additively to multi-mutant variants. This reformulation enables exact constrained Thompson sampling over the combinatorial design space. Using the Kermut kernel, BaLiSq substantially outperforms established multi-mutant predictors in the low-data regime while matching their performance as data grows. BaLiSqO combines BaLiSq with Thompson sampling and achieves lower mean regret than directed evolution based baselines on simulated landscapes. Together, these results demonstrate that kernel-informed additive models combine data-efficient prediction with tractable acquisition optimization over large combinatorial sequence spaces.

1 INTRODUCTION

Protein engineering is the task of improving a protein sequence with respect to some target property. Often starting with a wild type sequence of length L, mutations are introduced to discover a better variant of the wild type. Even if we restrict the number of mutations we can add to the wild type to at most $M \in \mathbb{N}$, the search space is still vast, containing in the order of $\sum_{m=1}^{M} \binom{L}{m} 19^m$ potential candidates. Practical campaigns are often constrained to batches of 96 wet-lab evaluations per round, with a total budget of a few hundred sequences (Bixby et al., 2026; Yang et al., 2025). The combination of an astronomically large design space and extreme data scarcity makes protein engineering one of the most challenging black-box optimization problems in science.

Gaussian process models can in principle capture structural similarity between variants via expressive kernels, but optimizing acquisition functions over the full design space remains difficult. Naive dense joint sampling for Thompson sampling is infeasible since $\sum_{m=1}^{M} \binom{L}{m} 19^m > 10^{14}$ when $L = 50$ and $M = 6$. Purely additive models (e.g. MoCHI (Faure & Lehner, 2024), GMMA (Johansson et al., 2023)) extrapolate gracefully to multi-mutant variants by summing per-site effects, but lack the structural prior needed to predict the effect of *unseen* single mutations in the scarce-data regime, and do not support principled uncertainty-guided exploration. Neither family simultaneously provides a structural kernel prior to enable learning at unseen substitutions *and* tractable acquisition over the combinatorial space. ALDE (Active Learning Directed Evolution) (Yang et al., 2025) uses batched Bayesian optimization over selected mutational sites.

In this paper we construct a Bayesian linear regression model from a positive-definite kernel K on the set I of single-mutant variants, by inducing a Gaussian prior on the weights of a Bayesian Linear Regression over Sequences $x \in \{0,1\}^p$, $p = LA$ ($A = 19$ non-wild type amino acids per site). This framework, termed **BaLiSq** (**Ba**yesian **Li**near Regression of **Seq**uences), has the following properties: (a) it is *equivalent* to GP regression with kernel K on single-mutant variants; (b) becomes *additive* on multi-mutant variants, enabling interpretable per-site effect estimates; and (c) enables exact maximization of Thompson samples over the full combinatorial design space, including exclusion of previously selected variants. The framework is kernel-agnostic. BaLiSq can account

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

for global epistasis by adding a sigmoid transformation to its output, being equivalent to a Bayesian neural network with no hidden layers.

Contributions:

- **The BaLiSq and BaLiSqO frameworks.** BaLiSq is a GP induced Bayesian linear regression over protein variants, defined for any positive-definite kernel. BaLiSqO is a Bayesian optimization method combining BaLiSq with Thompson sampling over the entire mutation-constrained design space.
- **GP equivalence and additivity.** We show that BaLiSq is exactly equivalent to GP regression with kernel K on single-mutant variants. Moreover, we characterize the additive mean and posterior variance on multi-mutant variants.
- **Efficient Thompson sampling with BaLiSqO.**
  BaLiSqO requires one p-dimensional weight sample per slot followed by exact maximisation over unselected variants. Without exclusions, this is a per-site argmax and top-M selection; exclusions are handled by the ranked search in Appendix A. Naive dense joint GP sampling uses $O(N_M^2)$ memory, where N_M is the number of feasible variants; this is not a lower bound for all GP sampling methods.
- **Closed-form information gain.** BaLiSq permits closed-form evaluation of information gain (Appendix A.4).
- **Empirical predictive evaluation.** We evaluate **BaLiSq** (with the Kermut kernel (Groth et al., 2024)), ablated against MoCHI (Faure & Lehner, 2024) and the recent MULTI-evolve (Tran et al., 2026) on ProteinGym multi-mutant variant dataset, and show increased performance on small and modest size datasets and on par performance on large data sets.
- **Protein engineering evaluation.** We evaluate **BaLiSqO** on *in silico* benchmarks and find lower mean regret and more diverse high-scoring variants than directed evolution based methods on these landscapes, at the same oracle-evaluation budget.

2 Background

2.1 Bayesian linear regression and Gaussian processes

Let $y \in \mathbb{R}$ be a response variable and $x \in \mathbb{R}^p$ a feature vector. A Bayesian linear regression (BLR) model assumes $f(x) = x^\top w$ with a Gaussian prior $w \sim \mathcal{N}(\mu_w, \Sigma_w)$ and Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$. Given a design matrix $X \in \mathbb{R}^{n \times p}$ and observations $Y \in \mathbb{R}^n$, the posterior over w is Gaussian with

$$\mu_{w|X,Y} = \mu_w + \Sigma_w X^\top (\sigma^2 I_n + X \Sigma_w X^\top)^{-1} (Y - X\mu_w) \tag{1}$$

$$\Sigma_{w|X,Y} = \Sigma_w - \Sigma_w X^\top (\sigma^2 I_n + X \Sigma_w X^\top)^{-1} X \Sigma_w \tag{2}$$

and the noise-free predictive posterior at x^* is $p(f|x^*, X, Y) = \mathcal{N}(f;\, x^{*\top} \mu_{w|X,Y},\, x^{*\top} \Sigma_{w|X,Y} x^*)$.

Any BLR model with prior $w \sim \mathcal{N}(\mu_w, \Sigma_w)$ is a GP on $f(x) = x^\top w$ with kernel $k(x, x') = x^\top \Sigma_w x'$; conversely, a GP with a finite-dimensional feature map ϕ and kernel $k(x, x') = \phi(x)^\top \Sigma_w \phi(x')$ is exactly a BLR model. See Rasmussen & Williams (2006) for an in depth introduction of Gaussian processes for machine learning.

2.2 Bayesian optimization

In Bayesian optimization (BO) we seek $x^* = \arg\max_{x \in \mathbb{X}} f(x)$ for a non-differentiable black-box function f using a limited evaluation budget. A surrogate model (typically a GP) is fit to observations and an acquisition function selects the next evaluation point by balancing exploration and exploitation. Popular acquisition functions include Thompson sampling (TS) (Russo et al., 2018), Upper Confidence Bound (UCB) (Srinivas et al., 2010), Expected Improvement (EI) (Jones et al., 1998), Predictive Entropy Search (PES) (Hernández-Lobato et al., 2014), Max-value Entropy Search

(MES) (Wang & Jegelka, 2017), and Knowledge Gradient (Frazier et al., 2009). In batched BO, up to q inputs are evaluated per round.

3 BaLiSq: A KERNEL-PRIOR BLR FRAMEWORK FOR PROTEIN VARIANTS

Previous studies such as MoCHI (Faure & Lehner, 2024) and GMMA (Johansson et al., 2023) showed that additive models perform well for multi-mutant variant effect prediction. Moreover, they have the advantage of being easy to optimize. Kermut (Groth et al., 2024), a GP for proteins, achieves state-of-the-art performance on the ProteinGym supervised track (single-mutant variants) by encoding structural and sequence similarity through a composite kernel. We combine the strengths of both families: BaLiSq uses any positive-definite kernel on single-mutant variants as a Gaussian prior on the BLR weights, giving a model that (a) inherits the kernel's prior, (b) predicts multi-mutant variants additively from per-site weight estimates, and (c) allows exact constrained Thompson sampling.

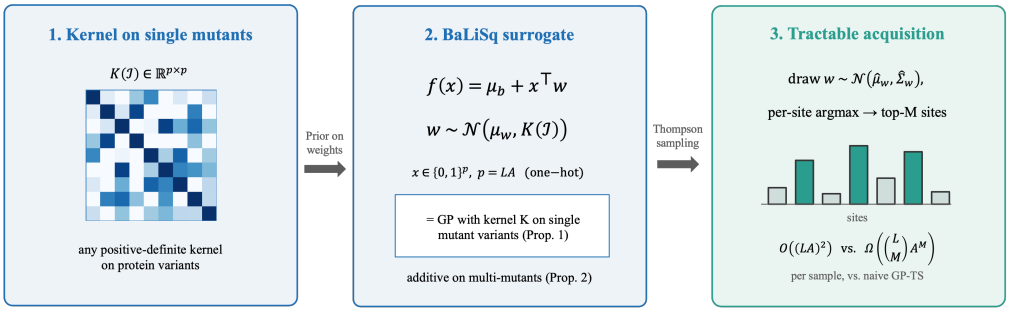


Figure 1: Overview of BaLiSq. Any positive-definite kernel on single-mutant variants is used as the prior covariance of a Bayesian linear regression over one-hot features. The resulting model is exactly GP regression on single mutants (Prop. 1), additive on multi-mutants (Prop. 2), and admits Gaussian process Thompson sampling (GP-TS) over the entire $\leq M$-mutant design space via per-site maximization and top-M selection without exclusions, or ranked search with exclusions. For the core linear model, each weight draw costs $O((LA)^2)$ after a shared $O((LA)^3)$ factorization; maximization without exclusions costs $O(LA + L\log L)$.

Definition. Let I denote the set of all single-mutant variants relative to the wild type, encoded as one-hot features $x \in \mathcal{X} := \{0,1\}^p$ with $p = L \cdot A$ ($A = 19$ non-wild type amino acids). The wild type is the zero vector $\mathbf{0}$. Let $k : \tilde{\mathcal{X}} \times \tilde{\mathcal{X}} \to \mathbb{R}$ be any positive-definite kernel for protein sequences, where $\mathcal{I} \subseteq \tilde{\mathcal{X}} \subseteq \mathcal{X}$ and $\mathcal{I}$ is the set of single mutant variants (that is, k should at least be defined on the single mutant variants). Let $K(I) \in \mathbb{R}^{p\times p}$ denote the corresponding kernel matrix on all single-mutant variants, $K(I)_{i,j} = k(I_i, I_j)$. **BaLiSq** is the BLR model $f(x) = \mu_b + x^\top w$ where the bias $\mu_b \in \mathbb{R}$ and weights $w \sim \mathcal{N}(\mu_w, K(I))$, with μ_w an optional data-driven prior mean.

Posterior. Given training data (X, Y), the BaLiSq posterior follows equation 1-equation 2 with $\Sigma_w = K(I)$ and Y replaced by $Y - \mu_b\mathbf{1}$. The posterior mean for any variant x (including multi-mutants) is $\hat{f}(x) = \mu_b + x^\top \mu_{w|X,Y}$, which decomposes as a sum of per-site weight contributions - directly interpretable without reference to other mutations.

Global epistasis may be modeled by pushing the output of the BLR through a sigmoid scaling. This is equivalent to a tiny Bayesian neural network, with no hidden layers and a possibly advanced prior on the parameters. Without the prior, this gives an unpenalized additive model with a sigmoid head, comparable to MoCHI when the parameterization and loss agree. All Gaussian posterior and acquisition results below concern the linear model, conditional on its fitted hyperparameters.

Definition. We define **BaLiSqO** as a batched Bayesian optimization method over protein variants with up to M mutations of a wild type. With BaLiSq as surrogate function, the acquisition function does Thompson sampling, by sampling over the posterior of the parameters. For each batch slot, we draw the weights once and find the *argmax* of that sampled function over variants that have neither been evaluated nor selected earlier in the batch (Algorithm 1). We do not redraw the weights when the

unconstrained maximizer has already been selected. The posterior remains fixed within each batch. Naive dense sampling of GP function values over this combinatorial space is infeasible. Sampling in weight space lets us overcome this hurdle.

4 THEORETICAL PROPERTIES OF BALISQ

In the following we show the formal properties of BaLiSq: equivalence to a GP on single mutant variants, additivity, and exact constrained Thompson sampling. Conditional convergence and information-gain evaluation are discussed in Appendix A. All statements are for an arbitrary positive-definite kernel $K : I \times I \to \mathbb{R}$ on the set I of single-mutant variants encoded as one-hot indicators $x \in \{0,1\}^p$ ($p = LA$, $A = 19$). $K(I) \in \mathbb{R}^{p\times p}$ is the kernel Gram matrix with rows and columns ordered consistently with the one-hot feature index. The BaLiSq prior is $w \sim \mathcal{N}(\mu_w, K(I))$ with Gaussian observation noise σ^2.

Proposition 1 (Equivalence on single-mutant variants)**.** *Let $X_s \in \{0,1\}^{n\times p}$ be a design matrix whose rows are one-hot encodings of single-mutant variants in I (each row has exactly one non-zero entry). Then the BaLiSq predictive posterior at any test set $X_* \subseteq I$ equals the GP posterior obtained from prior covariance K on the function values, with the same mean $m(e_i) = \mu_b + \mu_{w,i}$, noise level σ^2, and fixed hyperparameters.*

Proof. See appendix A □

Proposition 2 (Additivity on multi-mutant variants)**.** *Fix a posterior $w|X, Y \sim \mathcal{N}(\hat{\mu}_w, \hat{\Sigma}_w)$ obtained from equation 1-equation 2. For any variant $x \in \{0,1\}^p$ with $|x|_1 = m$ mutations and $\mathrm{supp}(x) \subset \{1,\dots,p\}$ the indices of its non-zero entries, the BaLiSq posterior predictive at x is Gaussian with*

$$\mathbb{E}[f(x)|X,Y] = \mu_b + \sum_{i\in\mathrm{supp}(x)} \hat{\mu}_{w,i}, \tag{3}$$

$$\mathrm{Var}[f(x)|X,Y] = x^\top \hat{\Sigma}_w\, x = \sum_{i\in\mathrm{supp}(x)} \hat{\Sigma}_{w,ii} + 2 \sum_{\substack{i<j \\ i,j\in\mathrm{supp}(x)}} \hat{\Sigma}_{w,ij}. \tag{4}$$

Proof. See appendix A □

4.1 ACQUISITION FUNCTIONS IN THE COMBINATORIAL SPACE

A primary advantage of BaLiSq is that the surrogate model is linear in the weight space: $f(x) = x^\top w$. Given the one-hot encoding $x \in \{0,1\}^{LA}$ for protein sequences of length L and alphabet size A, the function decomposes as a sum of per-site contributions $f(x) = \sum_{l=1}^{L}\sum_{a=1}^{A} x_{la} w_{la}$. We now show that this structure permits exact Thompson sampling. The bias is constant in x and does not affect its maximizer.

Proposition 3 (Thompson-sampling complexity)**.** *Let L be the sequence length, $A = 19$, and M the maximum number of mutations, with $1 \le M \le L$. Let $N_M = \sum_{m=0}^{M} \binom{L}{m} A^m$ be the size of the $\le M$-mutant design space. Then:*

- *(i) Naive dense joint GP sampling over the full design space constructs an $N_M \times N_M$ posterior covariance and uses $O(N_M^2)$ memory.*
- *(ii) BaLiSq Thompson sampling draws $w \sim \mathcal{N}(\hat{\mu}_w, \hat{\Sigma}_w)$ after one $O(p^3)$ Cholesky factorization shared across all slots in the round; each per-slot sample then costs $O(p^2)$ (one triangular multiply). Without exclusions, selecting the $\le M$-mutant maximiser of $\mu_b + w^\top x$ is a per-site argmax followed by a top-M selection of positive contributions, costing $O(LA + L\log L)$. Algorithm 1 instead maximises over $\mathcal{X}_M \setminus E$, where E contains evaluated variants and earlier selections in the batch. This requires an exact ranked search when the unconstrained maximiser is excluded; its cost is given in Appendix A.*

These algebraic costs do not depend on the choice of kernel K once the posterior covariance is available. The constrained search also depends on the rank of the highest-scoring variant outside E.

Algorithm 1 BaLiSqO: batched Thompson sampling over protein variants

Require: Wild type x_{wt}, mutation limit M, number of batches B, batch size q, single-mutant covariance $K(I)$, oracle y

$\mathcal{X}_M \leftarrow \{x : d_H(x, x_{\mathrm{wt}}) \leq M\}$
$\mathcal{A}_1 \leftarrow q$ distinct single mutants from the shared cold-start pool
$\mathcal{D}_1 \leftarrow \{(x, y(x)) : x \in \mathcal{A}_1\}$
for $r = 2, \ldots, B$ **do**
    Fit the BaLiSq surrogate $f(x) = \mu_b + x^\top w$ to $\mathcal{D}_{r-1}$
    Compute $w \mid \mathcal{D}_{r-1} \sim \mathcal{N}(\mu_{w|X,Y}, \Sigma_{w|X,Y})$ ▷ X, Y are the variants and oracle values in $\mathcal{D}_{r-1}$
    $E \leftarrow \{x : (x, y(x)) \in \mathcal{D}_{r-1}\}$; $\mathcal{A}_r \leftarrow \varnothing$
    for $j = 1, \ldots, q$ **do**
        Draw $w_r^{(j)} \sim \mathcal{N}(\mu_{w|X,Y}, \Sigma_{w|X,Y})$
        Set $f_r^{(j)}(x) \leftarrow \mu_b + x^\top w_r^{(j)}$
        $x_r^{(j)} \leftarrow \arg\max_{x \in \mathcal{X}_M \setminus E} f_r^{(j)}(x)$ ▷ exact constrained search
        $\mathcal{A}_r \leftarrow \mathcal{A}_r \cup \{x_r^{(j)}\}$; $E \leftarrow E \cup \{x_r^{(j)}\}$
    end for
    Evaluate $\mathcal{A}_r$ in parallel; $\mathcal{D}_r \leftarrow \mathcal{D}_{r-1} \cup \{(x, y(x)) : x \in \mathcal{A}_r\}$
end for
return $\arg\max_{(x,y(x)) \in \mathcal{D}_B} y(x)$

Proof. See appendix A □

4.2 BaLiSqO Algorithm

Algorithm 1 states the core linear model. The engineering extension adds jointly sampled mutation-depth offsets as described in Appendix A. Assume $1 \leq M \leq L$, $B \geq 1$, $Bq \leq |\mathcal{X}_M|$, and at least q distinct single mutants in the shared cold-start pool.

5 Empirical evaluation

5.1 BaLiSq-Kermut for protein variant effect prediction

BaLiSq-Kermut instantiates BaLiSq with $K = K_{\text{Kermut}}$ (the Kermut composite kernel) and an ESM-2 log-likelihood-ratio (LLR) affine prior mean μ_w (Lin et al., 2023). The hyperparameters are fit by marginal log-likelihood (AdamW, 150 steps, learning rate 0.1). This prediction model uses substitution features without the mutation-depth features used in the engineering experiments.

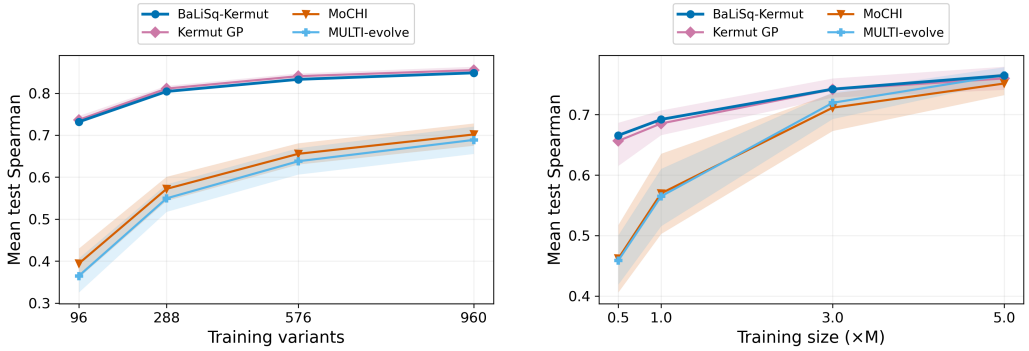
Previous studies such as MoCHI (Faure & Lehner, 2024) and GMMA (Johansson et al., 2023) have shown that additive models perform well in protein variant effect prediction for multi-mutant variants. Motivated to combine the extrapolation strength of additive models with Kermut's ability to predict unseen mutations in low-data regimes, we evaluate BaLiSq-Kermut on ProteinGym datasets with multi-mutant variants. We compare against Kermut (GP), MoCHI, and MULTI-evolve, an MLP that claims to model local epistasis through pairwise amino-acid interactions. Both neural network models (MoCHI, MULTI-evolve) are implemented as described in their respective papers and repositories.

In figure 2a we (for each data set) sample four nested training data sets of $n \times 96$ variants, where $n = 1, 3, 6, 10$; points not in the largest training set are held out as test data. Kermut and BaLiSq-Kermut achieve similar performance across all training set sizes; similarly MoCHI and MULTI-evolve show similar performance across training set sizes. The two Bayesian models have higher mean predictive performance than the neural network models across all training set sizes, with the largest gap for small n.

In figure 2b we use $n \times M$ training variants, where M is the number of unique single-mutant variants and $n = 0.5, 1, 3, 5$. BaLiSq-Kermut matches Kermut at small n and matches additive models at large n. This is an empirical comparison, rather than a consequence of the conditional MAP result in

Appendix A.2. Kermut also matches the performance of the neural networks on larger n. However, unlike Kermut, BaLiSq-Kermut scales efficiently to larger datasets, offers greater interpretability, and enables tractable evaluation of acquisition functions.

Appendix Figure C1 shows that without the prior mean function the Bayesian methods still outperform the neural network based methods.



(a) (b)

Figure 2: Average Spearman correlation across ProteinGym multi-mutant datasets. Shaded areas are pointwise 95% paired bootstrap intervals for differences from BaLiSq, shifted onto each method's mean curve; BaLiSq is the reference and has no paired band. Repeats are averaged within datasets before resampling datasets 10,000 times. (a) $n \times 96$ training points ($n = 1, 3, 6, 10$), averaged over 43 datasets. (b) $n \times M$ training points ($n = 0.5, 1, 3, 5$), averaged over 10 datasets. We ran the methods on 7 more datasets, but Kermut GP did not finish on those. Panel (a) uses three repeats and panel (b) one repeat. Dataset lists and interval details are in Appendix B.

5.2 Bayesian optimization of combinatorial protein variant space

5.2.1 Experimental setup

BaLiSqO is a Bayesian optimization (BO) method making Thompson sampling feasible on the entire protein design space with BaLiSq. We benchmark this against directed evolution (DE) (Romero & Arnold, 2009) and machine learning guided directed evolution (MLDE). The setup is that we have a budget of B batches, a batch size of 96, and a design space of all variants within M mutations of a wild type protein. For the purpose of this paper we will use the following definitions of BO, DE, and MLDE. In BO we select the set of variants in the search space that maximizes an acquisition function, which depends on the surrogate model. In DE we have a reference sequence, which for the first batch is the wild type. Then 96 single mutant variants are sampled uniformly at random among all possible single mutant variants and added to the batch, after each batch as been evaluated by the black box experiment, the best variant is selected as a new reference sequence. MLDE is similar to DE, except that the 96 variants are selected as the top scoring single mutant variants of the reference sequence according to the surrogate model. In our setup the surrogate is the same for MLDE and BO. Thus, the difference between BO and MLDE is that MLDE is a greedy selection, one mutation at a time, while BO explores the entire design space from early on. To make the methods comparable, we set the first batch to 96 random single mutant variants. Moreover we set $M = B$, i.e. the maximum number of allowed mutations equals the number of iterations. This is to have DE and MLDE end at the mutation bound in the last batch, so that they are more comparable to BO. Finally, we compare to a random baseline, which selects 6 mutant variants at random.

We optimize 23 different wild types with 3 seeds each. Running sufficient laboratory experiments to compare these methods, would be extremely expensive and time consuming. Instead we use 4 *oracles*. Three additive landscapes sum single-substitution scores from FoldX Stability, FoldX SASA (Solvent Accessible Surface Area), and RaSP (stability predictions), accessed through poli (González-Duque et al., 2024). These landscapes are additive by construction; they do not evaluate structural interactions in multi-mutants. We also use ProGen2 (Nijkamp et al., 2023) sequence

likelihood as a non-additive (epistatic) surrogate score. None of these oracles is a measurement of experimental fitness. The normalized ProGen2 analysis excludes CALM1 because its normalization denominator is zero; all panels and exclusions are listed in Appendix B.

Surrogate implementation. The main additive-landscape BaLiSqO arm is BaLiSq, the weighted-sum Kermut kernel, and an affine ESM-2 LLR prior mean. The kernel combines sequence and structural components as $K = \pi K_{\text{struct}} + (1-\pi)K_{\text{seq}}$, with learned $\pi \in (0,1)$. At each round, fitting uses 150 AdamW steps (learning rate 0.1), with raw parameters bounded to $[-20, 20]$. The engineering surrogate also includes mutation-depth features, giving sampled scores $\mu_b + x^\top w + h(d)$, $d = |x|_1$. The depth-indexed search remains exact by adding $h(d)$ to each final depth list before comparing depths (Appendix A). We draw the augmented weights once per slot and exclude evaluated and pending variants, keeping the posterior fixed within each batch.

5.2.2 Results

We report the normalized regret, $r(m, b)$ (lower is better) over an optimization campaign for a method m at batch b, and cumulative regret, $cr(m)$, over the campaign:

$$r(m,b) = \frac{f(x^*) - f(x_{\leq b})}{f(x^*) - f(x_{wt})}, \qquad cr(m) = \sum_{b=1}^{B} r(m,b)$$

where x^* is the argmax for the additive oracles, and the best found variant in the frozen pooled search results for ProGen2. Its regret is therefore an *empirical pseudo-regret*, relative to the search results rather than a known optimum. $x_{\leq b}$ is the best found variant for the method m in batch b or earlier.

Table 1 shows the normalized regret in the final batch averaged across the three additive oracles, the 23 wild types, and the three seeds. For ProGen2, the table averages the normalized empirical pseudo-regret over the same panel after excluding CALM1.

BaLiSqO gets the lowest normalized final regret, followed by MLDE, DE, and finally Random which has substantially higher regret. Note that MLDE and BaLiSqO use the same surrogate. The same tendency can be seen on the cumulative regret (table 1). Hence, BaLiSqO not only ends up finding better variants, but also does so with fewer oracle evaluations. This is both true for the additive oracles as well as the epistatic oracle.

BaLiSqO improves over MLDE by reducing the average normalized final regret by 45 % on the additive oracles and 41 % on the epistatic oracle. Compared to DE, BaLiSqO improves the average normalized final regret by 72 % and 81 % on the additive and epistatic oracles respectively. To assess the contribution of posterior uncertainty to BaLiSqO, we did two ablations: (1) A greedy version (*BaLiSqO Zero Uncertainty*), where the uncertainty estimates were set to zero, (2) a version of BaLiSqO where we shuffle the covariance matrix (*BaLiSqO Shufflevar*). Both result in increased normalized final regret and cumulative regret. We further tested *BaLiSqO Half Shufflevar* where only a random half the covariance matrix was permuted and obtained a final regret between BaLiSqO and BaLiSqO Shufflevar (see Appendix Figure C3a). These ablations support the use of structured posterior uncertainty in BaLiSqO. They do not assess calibration.

Table 1: Final and cumulative normalized regret over the 6 batches. ProGen2 entries are normalized empirical pseudo-regret. Intervals are marginal 95% t intervals across wild types after averaging seeds and, for the additive column, oracles within wild types.

	Final		Cumulative	
Method	Additive	Epistatic	Additive	Epistatic
BaLiSqO	**0.064** [0.036, 0.093]	**0.079** [0.036, 0.122]	**2.057** [1.870, 2.244]	**2.168** [1.713, 2.622]
MLDE	0.117 [0.097, 0.136]	0.134 [0.105, 0.163]	2.845 [2.759, 2.931]	2.667 [2.499, 2.834]
DE	0.234 [0.207, 0.260]	0.426 [0.356, 0.496]	3.236 [3.131, 3.341]	3.783 [3.489, 4.077]
Random	0.721 [0.701, 0.740]	0.831 [0.795, 0.867]	4.580 [4.487, 4.674]	5.068 [4.885, 5.250]

Appendix Figure C2 shows regret curves, i.e. the normalized regret at each batch for multiple methods. The regret of DE and MLDE decrease almost linearly when mutations are added, while the regret for

BaLiSqO, which can explore the entire protein variant space from the beginning, is decreasing more rapidly in the first batches and reaches a lower regret.

In this benchmark we have given BaLiSqO the disadvantage that it can only do at most 6 mutations, to make it more comparable to MLDE and DE. In practice one may increase this number and potentially get a bigger improvement over directed evolution based methods. In Appendix Figure C4 we extend the campaign by another batch, allowing MLDE to add a 7th mutation. We compare this to two versions of BaLiSqO: one only allowed to do up to 6 mutations, and one allowed to do 7 mutations. We see that at the 6th batch MLDE has the worst performance of the 3, but the benefit of adding an additional mutation makes it beat the version of BaLiSqO which is restricted to 6 mutations. However, the BaLiSqO which is allowed to do 7 mutations beats both methods.

5.2.3 DIVERSITY

Generating diverse hits is important in protein engineering in practice: One might discover that one variant underperforms on a downstream task, and having diverse alternatives can minimize the risk of having to restart the optimization from scratch. We measure the diversity of a set of variants S by the performance of a subset of k candidates $S_k^* = \{\mathbf{x}_1^*, ..., \mathbf{x}_k^*\}$ chosen the following way (similar to Maus et al. (2022)):

$$\begin{aligned}\mathbf{x}_1^* &= \arg\max_{\mathbf{x}\in\mathcal{S}} f(\mathbf{x}) \\ \mathbf{x}_i^* &= \arg\max_{\mathbf{x}\in\mathcal{S}\setminus\{\mathbf{x}_1^*,\ldots,\mathbf{x}_{i-1}^*\}} f(\mathbf{x}) \text{ s.t. } \delta\left(\mathbf{x}, \mathbf{x}_j^*\right) \geq \tau \\ &\text{for } j = 1, \ldots, i-1\end{aligned}$$

where $\delta\left(\mathbf{x}_i^*, \mathbf{x}_j^*\right)$ is the hamming distance between two protein sequences. τ is thereby the minimum number of residues by which a pair of sequences in S_k^* may differ.

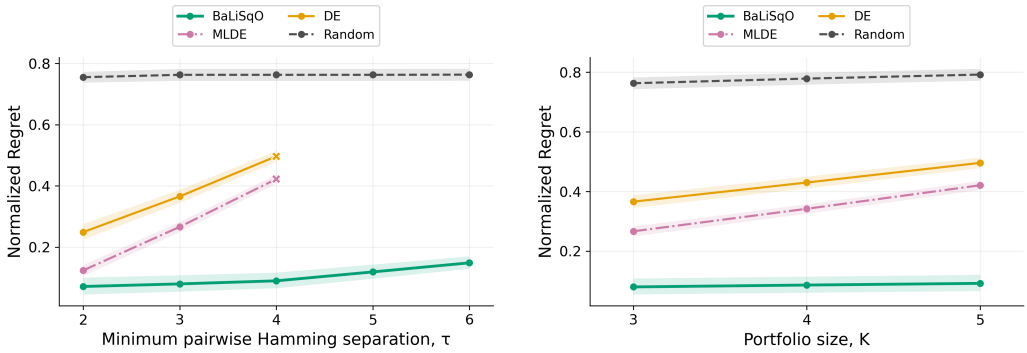
Because directed evolution based methods greedily add a new mutation to the reference variant in each round, they get very limited diversity in the generated candidates. Figure 3a shows the average normalized regret in the S_3^* sets for each method and different values of τ. For larger separations, the greedy selection did not return k variants in every run for MLDE and DE. On the contrary, the regret of BaLiSqO only increases slightly when τ increases. Similarly, when increasing k for fixed $\tau = 3$, the regret of MLDE and DE increase substantially faster than for BaLiSqO. The plots shows that BaLiSqO get multiple diverse variants all with low regret, while MLDE collects substantially worse variants when constructing a diverse set of candidates.

6 RELATED WORK

GP-based and NN methods for proteins. Kermut (Groth et al., 2024) achieves state-of-the-art supervised performance on ProteinGym (Notin et al., 2023) single-mutant benchmarks using a composite kernel that encodes sequence, structure, and evolutionary information. Greenman et al. (2025) benchmark GPs and Bayesian ridge regression (BRR) for protein fitness prediction and uncertainty quantification, but only is an isotropic Gaussian prior on the BRR. MULTI-evolve (Tran et al., 2026) combines protein-language-model-guided mutation selection with an MLP; ALDE (Yang et al., 2025) uses batched Bayesian optimization over selected mutational sites.

Additive models. MoCHI (Faure & Lehner, 2024) and GMMA (Johansson et al., 2023) estimate per-site mutation effects from relatively large training datasets, and push the additive sum through a sigmoid to account for global epistasis. BaLiSq is a Bayesian, kernel-prior generalization of this family: its linear posterior mean is additive. With a fixed sigmoid head, its MAP estimate approaches the matching unpenalized estimate under the additional curvature and boundedness assumptions in Appendix A.2. The key advantage over MoCHI/GMMA is that the kernel prior enables extrapolation to unseen mutations under data scarcity, while the additive structure retains interpretability and tractable acquisition.

Bayesian optimization in combinatorial and discrete spaces. COMBO (Oh et al., 2019) uses a GP with a graph kernel for combinatorial optimization but does not exploit the one-hot variant



(a) (b)

Figure 3: Diversity of BaLiSqO, MLDE, DE and Random on the additive landscapes. Lower is better. In (a) k is fixed at 3. In (b) τ is fixed at 3. Bands are pointwise 95% percentile wildtype-bootstrap intervals after averaging seeds and retained oracles within wild types. The SPTN1/FoldX-stability cell is excluded; SPTN1 remains in the other two oracles. A point is reported only if greedy selection returns k variants in every retained run, with no points after the first failure; crosses mark truncated endpoints.

feature structure that makes BaLiSq acquisition tractable. BOiLS (Grosnit et al., 2022) applies BO to logic synthesis, a different discrete domain. Baptista & Poloczek (2018) made a different version of Thompson sampling for combinatorial structures but not aimed at proteins. Moss et al. (2020) does BO over strings using string kernels.

7 Discussion and Limitations

We introduced BaLiSq, a Bayesian-linear-regression view of any positive-definite kernel on single-mutant protein variants. Two properties together set the framework apart from existing protein surrogates: it is additive on multi-mutants while preserving the full pairwise posterior covariance (Proposition 2); and it permits exact constrained Thompson sampling over the $\leq M$-mutant design space. Information gain about the weights can also be evaluated in closed form. Empirically, BaLiSq-Kermut inherits Kermut's strength in the very-low-data regime, and attains similar predictive performance to additive frequentist models in the data-rich regime; the ProteinGym multi-mutant evaluation traces this crossover quantitatively. On the simulated landscapes considered here, BaLiSqO finds better and more diverse variants at the same oracle-evaluation budget than machine learning guided directed evolution.

Bayesian optimization with BaLiSq rely on the quality of the kernel - both in terms of its predictive capabilities and its uncertainty estimates. Our ablations support the use of structured posterior uncertainty with Kermut, and we hypothesize that the development of novel protein kernels will improve the benefit of the proposed framework. Especially the calculation of expected information gain relies on accurate estimates of the entire covariance matrix, not just its diagonal. The core BaLiSq mean is additive; correlations between weight uncertainties do not model pairwise epistasis in that mean. The engineering depth offsets and an optional global sigmoid head allow restricted departures from additivity, but do not model arbitrary substitution-specific interactions, and a richer construction would be needed in regimes dominated by specific contact-mediated epistasis. We argue that in a practical setting in the low data regime it might be beneficial to discard the ambition of modeling epistasis and enjoy the plethora of advantages that comes with additive models.

AI USE STATEMENT

AI has been used to assist with coding, code reviewing and writing as well as polish the writing of the manuscript. AI was not used to generate the core research ideas, formulate the proposed method, or determine the scientific conclusions of the work. The authors take full responsibility for the content of this paper.

ETHICS STATEMENT

This work is entirely computational and uses publicly available datasets from ProteinGym and publicly available in silico protein-property predictors. It does not involve human subjects, animal experiments, biological sample collection, or newly generated wet-lab data. The work focuses on methodological development to make protein engineering more interpretable and efficient. While computational protein engineering methods may have broader dual-use potential, the outputs of this work require substantial downstream experimental validation before practical use. We encourage responsible and ethical use of the methods presented here. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

BaLiSq is defined in Section 3 and BaLiSqO in Algorithm 1; proofs of all theoretical results are in Appendix A. Section 5 specifies the experimental protocol: ProteinGym training-set sizes and held-out test sets, the kernel and prior mean, hyperparameter fitting (AdamW, 150 steps, learning rate 0.1), batch size, budget and mutation bound, the baselines, the regret definitions, and the interval construction. Appendix B lists every wild type in each panel, every exclusion and its reason, the depth-feature rule, and the ablation definitions. All data and oracles are public: ProteinGym (Notin et al., 2023), ProGen2 (Nijkamp et al., 2023), and FoldX Stability, FoldX SASA and RaSP through poli (González-Duque et al., 2024) (FoldX requires a free academic licence). We will release the full code under an open-source licence upon acceptance, including configurations, seeds, and scripts that regenerate every table and figure.

AUTHOR CONTRIBUTIONS

ACKNOWLEDGMENTS

REFERENCES

Ricardo Baptista and Matthias Poloczek. Bayesian optimization of combinatorial structures. In *International conference on machine learning*, pp. 462–471. PMLR, 2018.

Eli Bixby, Gino Brunner, Daniel Danciu, Richard Dela Rosa, Nicolas Deutschmann, Constance Ferragu, Franziska Geiger, Christian Holberg, Patrick Kidger, Arthur Lindoulsi, et al. What comes after de novo? automated lead optimization of proteins with cradle-1. *bioRxiv*, pp. 2026–03, 2026.

Andre J Faure and Ben Lehner. Mochi: neural networks to fit interpretable models and quantify energies, energetic couplings, epistasis, and allostery from deep mutational scanning data. *Genome Biology*, 25(1):303, 2024.

Peter Frazier, Warren Powell, and Savas Dayanik. The knowledge-gradient policy for correlated normal beliefs. *INFORMS journal on Computing*, 21(4):599–613, 2009.

Miguel González-Duque, Simon Bartels, and Richard Michael. poli: a library for discrete objective functions, January 2024. URL `https://github.com/MachineLearningLifeScience/poli`.

Kevin P Greenman, Ava P Amini, and Kevin K Yang. Benchmarking uncertainty quantification for protein engineering. *PLOS Computational Biology*, 21(1):e1012639, 2025.

Antoine Grosnit, Cedric Malherbe, Rasul Tutunov, Xingchen Wan, Jun Wang, and Haitham Bou Ammar. Boils: Bayesian optimisation for logic synthesis. In *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1193–1196. IEEE, 2022.

Peter Mørch Groth, Mads Herbert Kerrn, Lars Olsen, Jesper Salomon, and Wouter Boomsma. Kermut: Composite kernel regression for protein variant effects. *Advances in neural information processing systems*, 37:29514–29565, 2024.

José M Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in neural information processing systems*, 27, 2014.

Kristoffer E Johansson, Kresten Lindorff-Larsen, and Jakob R Winther. Global analysis of multi-mutants to improve protein function. *Journal of Molecular Biology*, 435(8):168034, 2023.

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

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

Natalie Maus, Kaiwen Wu, David Eriksson, and Jacob Gardner. Discovering many diverse solutions with bayesian optimization. *International Conference on Artificial Intelligence and Statistics*, 2022.

Henry Moss, David Leslie, Daniel Beck, Javier Gonzalez, and Paul Rayson. Boss: Bayesian optimization over string spaces. *Advances in neural information processing systems*, 33:15476–15486, 2020.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, et al. Proteingym: Large-scale benchmarks for protein fitness prediction and design. *Advances in neural information processing systems*, 36: 64331–64379, 2023.

Changyong Oh, Jakub Tomczak, Efstratios Gavves, and Max Welling. Combinatorial bayesian optimization using the graph cartesian product. *Advances in Neural Information Processing Systems*, 32, 2019.

Carl E Rasmussen and Christopher K I Williams. *Gaussian processes for machine learning*. Number 3. MIT press Cambridge, MA, 2006.

Philip A Romero and Frances H Arnold. Exploring protein fitness landscapes by directed evolution. *Nature reviews Molecular cell biology*, 10(12):866–876, 2009.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, and Zheng Wen. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–99, 2018.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 1015–1022, 2010.

Vincent Q Tran, Matthew Nemeth, Liam J Bartie, Sita S Chandrasekaran, Alison Fanton, Hyungseok C Moon, Brian L Hie, Silvana Konermann, and Patrick D Hsu. Rapid directed evolution guided by protein language models and epistatic interactions. *Science*, 392(6798):eaea1820, 2026. doi: 10.1126/science.aea1820.

Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient bayesian optimization. In *International conference on machine learning*, pp. 3627–3635. PMLR, 2017.

Jason Yang, Ravi G Lal, James C Bowden, Raul Astudillo, Mikhail A Hameedi, Sukhvinder Kaur, Matthew Hill, Yisong Yue, and Frances H Arnold. Active learning-assisted directed evolution. *Nature Communications*, 16(1):714, 2025.

A EXTENDED PROOFS FOR SECTION 4

Notation follows section 3: I is the set of single-mutant variants encoded as one-hot indicators in $\{0,1\}^p$ with $p = LA$, K is a fixed positive-definite kernel on I, the BaLiSq prior is $w \sim \mathcal{N}(\mu_w, K(I))$, and the likelihood has Gaussian noise σ^2.

Assume $\sigma^2 > 0$ and condition on the fitted bias and hyperparameters. The GP-equivalence, additivity, and information-gain identities allow $K(I) \succeq 0$. The ordinary Cholesky and inverse-prior arguments assume $K(I) \succ 0$; a singular Gaussian can instead be sampled using a covariance square root, with the same dense asymptotic upper bound.

A.1 PROOFS FOR PROPOSITIONS IN SECTION 4

Proof of proposition 1 (equivalence on single-mutant variants):

Proof. Let e_i denote the i-th standard basis vector in $\mathbb{R}^p$ and let $\iota : I \to \{1, \dots, p\}$ be the bijection sending each single-mutant variant to its one-hot index. For any single-mutant variants $x, x' \in I$ we have $x^\top K(I)\, x' = e_{\iota(x)}^\top K(I)\, e_{\iota(x')} = K(I)_{\iota(x),\iota(x')} = K(x, x')$, so $X_s K(I) X_s^\top = K(X_s)$ entrywise, and likewise for any cross block involving X_*. Substituting into the BLR predictive posterior reproduces the standard GP posterior mean and covariance with kernel K and noise σ^2. The prior means agree by construction since $f(x) = \mu_b + x^\top \mu_w$ evaluates to a single coordinate of μ_w plus the bias μ_b for any $x \in I$, which is exactly the GP prior mean parameterisation. □

Proof of proposition 2 (Additivity on multi-mutant variants):

Proof. The posterior predictive is the pushforward of $w|X, Y$ through the linear functional $w \mapsto \mu_b + x^\top w$, which is Gaussian with mean $\mu_b + x^\top \hat{\mu}_w$ and variance $x^\top \hat{\Sigma}_w\, x$. Because x is binary, $x^\top \hat{\mu}_w = \sum_{i \in \mathrm{supp}(x)} \hat{\mu}_{w,i}$, and expanding $x^\top \hat{\Sigma}_w\, x$ over the $\mathrm{supp}(x) \times \mathrm{supp}(x)$ block gives equation 4. □

Proof of proposition 3 (Thompson-sampling complexity):

Proof. *(i)* Naive dense joint sampling materialises the $N_M \times N_M$ posterior covariance. Storing it uses $O(N_M^2)$ memory, and $N_M \geq \binom{L}{M} A^M$. Structured or approximate GP samplers need not materialise this matrix.

(ii) For BaLiSq, the posterior $w|X, Y$ is a p-dimensional Gaussian, and a single Cholesky $\hat{\Sigma}_w = LL^\top$ at cost $O(p^3)$ yields independent samples at $O(p^2)$ each (the Cholesky is shared across all slots in a batch). Given a sample w, the optimization $\max_{x : |x|_1 \leq M,\ x \in \{0,1\}^p,\ \text{at most one entry per site}} \mu_b + w^\top x$ is solvable exactly because the objective is additive across sites and constraints are per-site. At each of the L sites, compute $b_s := \max_{a \neq \mathrm{wt}_s} w_{s,a}$ in $O(A)$. Discard sites with $b_s \leq 0$. Sort the surviving sites by b_s in $O(L \log L)$ and pick the top M: this is optimal because the objective is a sum of independent per-site choices. The total cost for finding the unconstrained maximum is $O(LA + L \log L)$. This proves the bound without exclusions. For Algorithm 1, we keep the same weight sample and use the exact ranked search proved in Appendix A.3. □

These are latent fitness variances; observation variances additionally include σ^2. The mean is additive in the mutation set, but the variance retains the full pairwise covariance structure inherited from K via $\hat{\Sigma}_w$. In particular, two mutations in close structural contact under K_{Kermut} are not treated as independent: their joint contribution to the predictive uncertainty is modulated by their off-diagonal entry $\hat{\Sigma}_{w,ij}$.

The same calculation gives the posterior *covariance* between two variants $x, x' \in \{0,1\}^p$ as

$$\mathrm{Cov}\big[f(x),\, f(x') \,\big|\, X, Y\big] \;=\; x^\top \hat{\Sigma}_w\, x' \;=\; \sum_{\substack{i \in \mathrm{supp}(x) \\ j \in \mathrm{supp}(x')}} \hat{\Sigma}_{w,ij},$$

i.e. the sum of the entries of $\hat{\Sigma}_w$ over the $\mathrm{supp}(x) \times \mathrm{supp}(x')$ block. In particular, the posterior is jointly Gaussian over any *set* of multi-mutant variants with covariance matrix $X\hat{\Sigma}_w X^\top$, computable from the $p \times p$ weight covariance alone, without ever materialising a kernel on multi-mutant sequences.

We record two consequences of section 4.

Equivalence of marginal likelihoods. The training function values and observations (on single mutant variants) have the same joint Gaussian under both representations (GP and BaLiSq) when means, noise, and hyperparameters agree. Consequently, the marginal likelihood $p(Y \mid X_s)$ and its hyperparameter objective agree. Different numerical optimization runs may still find different fitted hyperparameters.

Equivalence does not extend to multi-mutant variants. For a multi-mutant variant x with $|x|_1 = m \geq 2$, the BaLiSq prior covariance between $f(x)$ and $f(x')$ is $x^\top K(I)x'$, which is the *sum* of the entries of K restricted to the $\mathrm{supp}(x) \times \mathrm{supp}(x')$ block. This is not in general equal to any kernel evaluation $K(x, x')$ that one might extend to multi-mutants, and indeed for most domain kernels (Kermut included) the natural multi-mutant extension disagrees with this sum.

A.2 Conditional convergence of the MAP estimate

Let g be a fixed sigmoid head, μ_b a fixed bias, and $D_N(w)$ the negative log likelihood of the N observations under this head. The negative log posterior is

$$J_N(w) = D_N(w) + \tfrac{1}{2}(w - \mu_w)^\top K(I)^{-1}(w - \mu_w).$$

Proposition 4 (Conditional MAP convergence)**.** *Fix μ_w, a positive-definite $K(I)$, and the feature dimension. Suppose u_N and v_N are interior stationary points of J_N and D_N, respectively, in a common bounded convex region independent of N. Assume D_N is twice continuously differentiable there and $\nabla^2 D_N(w) \succeq cNI$ throughout that region for a constant $c > 0$ independent of N. Then $\|u_N - v_N\| = O(N^{-1})$. This applies to a MoCHI estimator only when the parameterization, loss, and fixed head agree with those of the unpenalized model.*

Proof. Write $Q = K(I)^{-1}$ and $\Delta = u_N - v_N$. Stationarity gives $\nabla D_N(u_N) = -Q(u_N - \mu_w)$ and $\nabla D_N(v_N) = 0$. Integrating the Hessian along the segment between the two points gives

$$cN\|\Delta\|^2 \leq \Delta^\top [\nabla D_N(u_N) - \nabla D_N(v_N)] \leq \|\Delta\| \, \|Q\| \, \|u_N - \mu_w\|.$$

If $\Delta \neq 0$, division yields the claimed rate since the region and prior are fixed. If $\Delta = 0$, the bound holds trivially. □

A.3 Proposition 3 (Thompson-sampling complexity)

Two clarifications are useful for implementation.

Exact search with exclusions. For fixed sampled weights, process sites in order and retain the top k partial variants at each mutation depth $d = 0, \ldots, M$. At each site, extend each retained partial variant by leaving the site unchanged or choosing one of its A substitutions, and retain the top k at each new depth. Merge the depth lists after the last site. Ties may be broken deterministically. Search with $k = 1, 2, 4, \ldots$, capped at $|E| + 1$, and return the highest-scoring variant outside E.

To see why retaining k partial variants is sufficient, consider a discarded partial variant at a fixed depth. There are k retained partials with scores at least as large. Any feasible completion of the discarded partial is also feasible for each of these retained partials and adds the same score. Thus it cannot be needed to obtain the top k completed scores. Induction over sites proves the ranked search exact. Provided $\mathcal{X}_M \setminus E$ is nonempty, at least one of the top $|E| + 1$ variants is outside E, so the search terminates and returns a constrained maximiser without resampling the weights.

If the best unexcluded variant has rank $r \leq |E| + 1$, the doubling search uses $O(LMAr \log(1 + Ar))$ score operations, since each depth update sorts at most $(A + 1)k$ candidates and the widths grow geometrically. This bound treats score and backpointer operations as constant time; explicitly copying

mutation lists, as in our implementation, adds representation cost. The per-site top-M bound alone therefore does not describe all slots of Algorithm 1.

Mutation-depth features in the engineering surrogate. The implementation samples jointly over substitution and depth weights. For a fixed sample its score is $\mu_b + x^\top w + h(d)$, with $d = |x|_1$. The offset $h(d)$ is constant within a depth, so adding it to each final depth list before merging leaves the ranked-search proof unchanged. Without exclusions, sort the site maxima, retain even negative maxima, and compare $\mu_b + h(d)$ plus the sum of the top d maxima for every $d = 0, \ldots, M$. Discarding negative contributions before comparing depths is not valid for this extension. Sampling costs use the augmented feature dimension instead of p.

A.4 Information-gain evaluation

In experimental design, we seek information about a parameter such as the model weights w Lindley (1956). For a GP information gain on the predictive posterior on the space of all possible protein variants within some hamming distance of a wild type would be infeasible to compute in anything lager than a few substitutions. For BaLiSq, information gain on the parameters reflect the information gain on the variant space and can be evaluated in closed form at a supplied candidate.

Proposition 5 (Information gain about the weights)**.** *For a Gaussian weight posterior with covariance $\hat{\Sigma}_w$ and independent Gaussian observation noise with $\sigma^2 > 0$,*

$$I(y; w \mid \mathcal{D}, x) = \tfrac{1}{2} \log(1 + \sigma^{-2} x^\top \hat{\Sigma}_w x).$$

Proof. The conditional predictive variance is $x^\top \hat{\Sigma}_w x + \sigma^2$. Conditioning additionally on w leaves variance σ^2. Subtracting the two Gaussian entropies gives the result. □

For a supplied batch with design matrix Z, the same argument gives $\frac{1}{2} \log\det(I + \sigma^{-2} Z \hat{\Sigma}_w Z^\top)$. These are evaluation formulas. Maximizing the single-candidate expression requires maximizing $x^\top \hat{\Sigma}_w x$. An exactly diagonal posterior is sufficient for per-site choices and top-M selection on the full design space without exclusions; a general posterior couples sites. A diagonal prior need not give a diagonal posterior after observing multi-mutant variants; approximate diagonality or low rank alone does not justify this optimization rule.

B Panels and reporting details

Protein engineering panels. The additive regret table, ablations, and per-oracle regret curves use the following wild types, with three seeds per wild type: `A0A247D711_LISMN`, `AMIE_PSEAE`, `CALM1_HUMAN`, `CCDB_ECOLI`, `DNJA1_HUMAN`, `DYR_ECOLI`, `ESTA_BACSU`, `HIS7_YEAST`, `NKX31_HUMAN`, `OBSCN_HUMAN`, `PHOT_CHLRE`, `PKN1_HUMAN`, `POLG_PESV`, `Q6WV12_9MAXI`, `RCD1_ARATH`, `RFAH_ECOLI`, `RL40A_YEAST`, `RPC1_BP434`, `RS15_GEOSE`, `SBI_STAAM`, `SPTN1_CHICK`, `UBC9_HUMAN`, `VRPI_BPT7`.

The normalized ProGen2 table and regret curve use this panel without `CALM1_HUMAN`, whose best-found-minus-wildtype denominator is zero. The additive diversity panels use the same wild types, excluding only `SPTN1_CHICK` with FoldX Stability for all seeds and methods because MLDE and DE not were able to find sufficient diverse sets, so including it would make it infeasible to include some of the points in the plot (the exclusion is in favor of MLDE and DE though); its FoldX SASA and RaSP runs remain included. The budget/depth comparison uses the full additive panel on FoldX Stability only.

Depth features. The engineering model omits depth features when all training variants have the same mutation depth. Otherwise, it uses indicators $\mathbf{1}\{d \geq t\}$ when more than two depths $t \geq 2$ each have at least 20 observations; with fewer qualifying depths it uses the scalar feature d. The depth weights have zero prior mean and an independent Gaussian prior block with fitted variance. They are sampled jointly with substitution weights from the augmented posterior. The nonlinear depth offset arises when the indicator features are active.

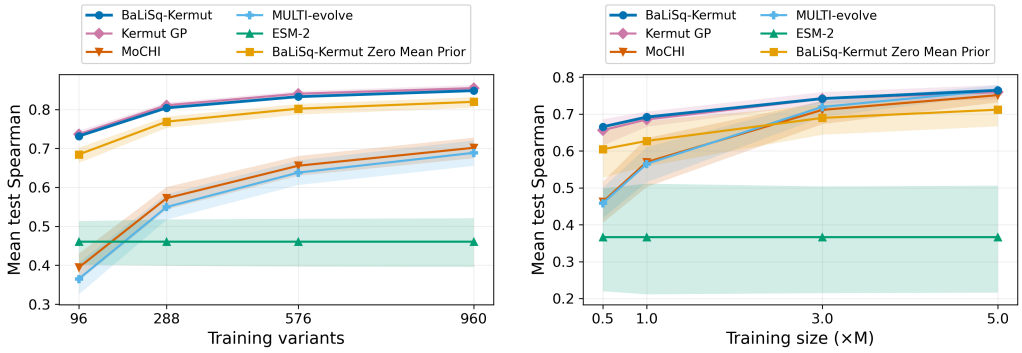
Intervals and comparisons. For protein engineering, seeds are averaged within wild types and, when pooling additive landscapes, retained oracles are averaged within wild types before averaging across proteins. Table 1 reports $\text{mean} \pm t_{0.975,n-1}\,\text{SEM}$ across the n wild types. Regret curves and ablation bars show ± 1 SEM. Diversity bands are percentile wildtype-bootstrap intervals. The budget/depth bands use paired wildtype-bootstrap differences from MLDE, shifted onto each arm's mean. ProteinGym bands similarly use paired dataset-bootstrap differences from BaLiSq, after averaging the available repeats within datasets. All bootstrap bands are pointwise 95% intervals, not simultaneous bands. An absent band on a reference method does not mean that its mean has no uncertainty. Marginal intervals are not paired tests of method differences.

ProteinGym panels. The candidate pool is the ProteinGym multi-mutant assays: all non-stability assays plus a sample of the Tsuboyama et al. (2023) stability assays, with `DMS_score` as the target. Splits are precomputed, so every model sees the same training and test variants, and test sets are capped at 10,000 variants. The fixed-size study (Fig. C1a) includes every assay in which all four nested training sets (96, 288, 576, 960 variants) fit within half the dataset. That gives 43 assays, each with three independent repeats. The relative-size study (Fig. C1b) includes every assay in which all four training sets (0.5, 1, 3, $5 \times M$) leave at least 1,000 test variants. That gives 17 assays, and we report one repeat. Seven of these 17 are omitted from the figure because exact Kermut did not complete at their largest training sets ($3 \times M$ and/or $5 \times M$, up to $\sim$17k training variants): D7PM05_CLYGR, GFP_AEQVI, HIS7_YEAST, PHOT_CHLRE, Q8WTC7_9CNID, and both RASK_HUMAN assays. They are the seven largest assays in the study. All other models completed on them. Panel membership was fixed before the results were examined, and no assay was excluded because of its results.

The fixed-size panel uses three repeats on these datasets: `A4_HUMAN_Seuma_2022`, `AMFR_HUMAN_Tsuboyama_2023_4G3O`, `BBC1_YEAST_Tsuboyama_2023_1TG0`, `CAPSD_AAV2S_Sinai_2021`, `CBPA2_HUMAN_Tsuboyama_2023_1O6X`, `CBX4_HUMAN_Tsuboyama_2023_2K28`, `CSN4_MOUSE_Tsuboyama_2023_1UFM`, `D7PM05_CLYGR_Somermeyer_2022`, `DLG4_HUMAN_Faure_2021`, `DNJA1_HUMAN_Tsuboyama_2023_2LO1`, `DOCK1_MOUSE_Tsuboyama_2023_2M0Y`, `EPHB2_HUMAN_Tsuboyama_2023_1F0M`, `F7YBW8_MESOW_Aakre_2015`, `F7YBW8_MESOW_Ding_2023`, `GCN4_YEAST_Staller_2018`, `GFP_AEQVI_Sarkisyan_2016`, `GRB2_HUMAN_Faure_2021`, `HECD1_HUMAN_Tsuboyama_2023_3DKM`, `HIS7_YEAST_Pokusaeva_2019`, `ISDH_STAAW_Tsuboyama_2023_2LHR`, `MBD11_ARATH_Tsuboyama_2023_6ACV`, `MYO3_YEAST_Tsuboyama_2023_2BTT`, `NKX31_HUMAN_Tsuboyama_2023_2L9R`, `NUSA_ECOLI_Tsuboyama_2023_1WCL`, `OBSCN_HUMAN_Tsuboyama_2023_1V1C`, `PABP_YEAST_Melamed_2013`, `PHOT_CHLRE_Chen_2023`, `POLG_PESV_Tsuboyama_2023_2MXD`, `PR40A_HUMAN_Tsuboyama_2023_1UZC`, `Q6WV12_9MAXI_Somermeyer_2022`, `Q8WTC7_9CNID_Somermeyer_2022`, `RASK_HUMAN_Weng_2022_abundance`, `RASK_HUMAN_Weng_2022_binding-DARPin_K55`, `RCRO_LAMBD_Tsuboyama_2023_1ORC`, `SDA_BACSU_Tsuboyama_2023_1PV0`, `SPA_STAAU_Tsuboyama_2023_1LP1`, `SPG1_STRSG_Olson_2014`, `SPG1_STRSG_Wu_2016`, `SPTN1_CHICK_Tsuboyama_2023_1TUD`, `UBE4B_HUMAN_Tsuboyama_2023_3L1X`, `VILI_CHICK_Tsuboyama_2023_1YU5`, `YAP1_HUMAN_Araya_2012`, `YNZC_BACSU_Tsuboyama_2023_2JVD`.

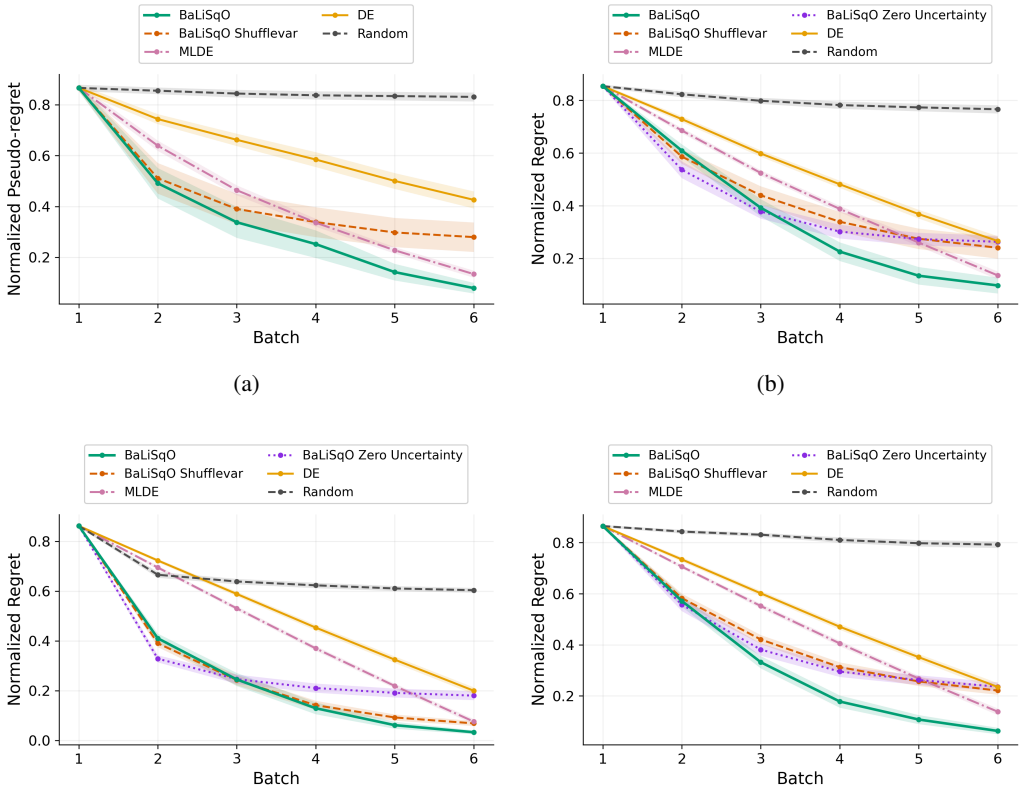
The relative-size panel uses one repeat on: `A4_HUMAN_Seuma_2022`, `CAPSD_AAV2S_Sinai_2021`, `F7YBW8_MESOW_Aakre_2015`, `F7YBW8_MESOW_Ding_2023`, `GRB2_HUMAN_Faure_2021`, `PABP_YEAST_Melamed_2013`, `Q6WV12_9MAXI_Somermeyer_2022`, `SPG1_STRSG_Olson_2014`, `SPG1_STRSG_Wu_2016`, `YAP1_HUMAN_Araya_2012`.

C Additional results and ablations



(a) (b)

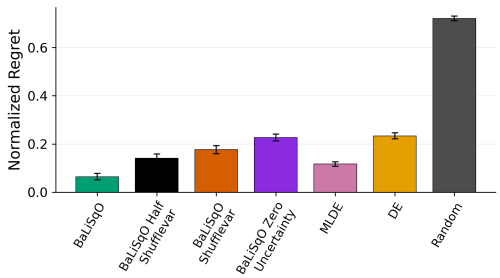
Figure C1: Average Spearman correlation across ProteinGym multi-mutant datasets. Shaded areas are pointwise 95% paired bootstrap intervals for differences from BaLiSq, shifted onto each method's mean curve. Repeats are averaged within datasets before resampling datasets 10,000 times; BaLiSq has no paired band. (a) $n \times 96$ training points ($n = 1, 3, 6, 10$), averaged over 43 datasets. (b) $n \times M$ training points ($n = 0.5, 1, 3, 5$), averaged over 10 datasets. Panel (a) uses three repeats and panel (b) one repeat. See Appendix B for the dataset lists. ESM-2 is LLR from the ESM-2 model.



(a) (b)

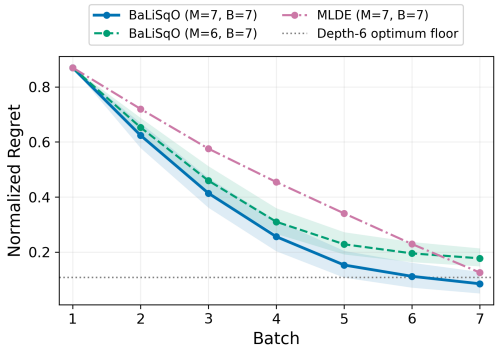
(c) (d)

Figure C2: Normalized regret averaged over the wild types and three seeds for each oracle. (a) ProGen2 normalized empirical pseudo-regret, excluding CALM1, (b) FoldX Stability, (c) FoldX SASA, (d) RaSP. Shaded areas indicate ± 1 SEM across wild types after averaging seeds. The additive panels retain all wild types.



(a)

Figure C3: Normalized final regret on the additive landscapes. Error bars indicate ± 1 SEM across wild types, after averaging seeds and oracles within wild types.



(a)

Figure C4: FoldX-stability budget/depth comparison on the additive panel. M is the maximum allowed number of mutations and B the number of batches. Regret is normalized against the depth-7 optimum for all arms; the depth-6 optimum floor is the best achievable regret under the depth-6 constraint on that scale. Bands are pointwise 95% paired wildtype-bootstrap intervals for differences from MLDE, shifted onto each arm's mean curve after averaging seeds. MLDE is the reference and has no paired band.