FACTORIZING LLM-GUIDED SEARCH FOR BLACK-BOX PROTEIN SEQUENCE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein engineering requires searching vast combinatorial sequence spaces under small experimental budgets. Large language models (LLMs) can propose protein mutations from previously measured variants, but direct LLM optimization conflates two distinct decisions: where to spend the next evaluation and which mutation to make. We introduce Factorized LLM Search (FLS), which separates these responsibilities: an explicit controller allocates evaluations across promising measured variants and feasible local edit components, while a frozen LLM proposes the concrete amino-acid substitution. Across four protein fitness landscapes and matched 500-evaluation budgets, FLS achieves the highest endpoint mean on three of four tasks among the end-to-end baseline methods evaluated. Controlled ablations show that LLM-generated mutations improve search over random legal mutations on two tasks. Analysis of search trajectories shows that direct LLM optimization can continue proposing novel substitutions while repeatedly sampling previously explored positional neighborhoods, whereas explicit allocation enables broader positional exploration. These results identify evaluation allocation and mutation generation as distinct components of LLM-guided protein optimization and provide an auditable framework for controlling each independently.

1 INTRODUCTION

Protein engineering requires identifying improved sequences from an enormous combinatorial space using only a small number of costly evaluations. Starting from a wild-type protein, an optimizer proposes variants, measures their fitness through an assay or predictive model, and uses those observations to choose subsequent evaluations. Because protein fitness landscapes are epistatic and labeled measurements are scarce (Fowler & Fields, 2014; Yang et al., 2019), the practical objective is not average predictive accuracy but the best sequence discovered within a fixed evaluation budget.

A central challenge is deciding how to spend that budget. Bayesian optimization (BO) makes this decision explicit through a surrogate model and acquisition function (Romero et al., 2013; Cheng et al., 2022; Groth et al., 2024; Yang et al., 2025), while protein language models paired with lightweight predictors provide another route to data-efficient optimization (Jiang et al., 2025; Xiong et al., 2026). More recently, general-purpose large language models (LLMs) have been used directly as black-box optimizers: measured sequences and fitness scores are placed in context and the LLM proposes the next variants to evaluate (Yang et al., 2024; Wang et al., 2025b; Liu et al., 2026). This is appealing because the same model can interpret scored examples and generate new protein sequences without fitting a task-specific surrogate. However, direct LLM optimization assigns two fundamentally different responsibilities to the same generation step: *where should the next evaluation be spent, and what mutation should be tested there?*

We find that these decisions can diverge. Even when explicitly instructed to balance exploration and exploitation, our full-history direct LLM baseline continues to generate previously unseen amino-acid substitutions while concentrating evaluations within previously sampled positional neighborhoods. On GB1, a well-characterized IgG-Fc binding landscape whose legal double-mutant space we can score exhaustively, 55.0% of accepted variants after query 200 contain a new amino-acid substitution, but only 10.5% test a previously unmeasured mutation-site pair (Figure 4). Thus generative novelty does not necessarily imply effective allocation of a limited evaluation budget. This is a whole-policy observation rather than evidence that in-context learning intrinsically prevents exploration; related

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.

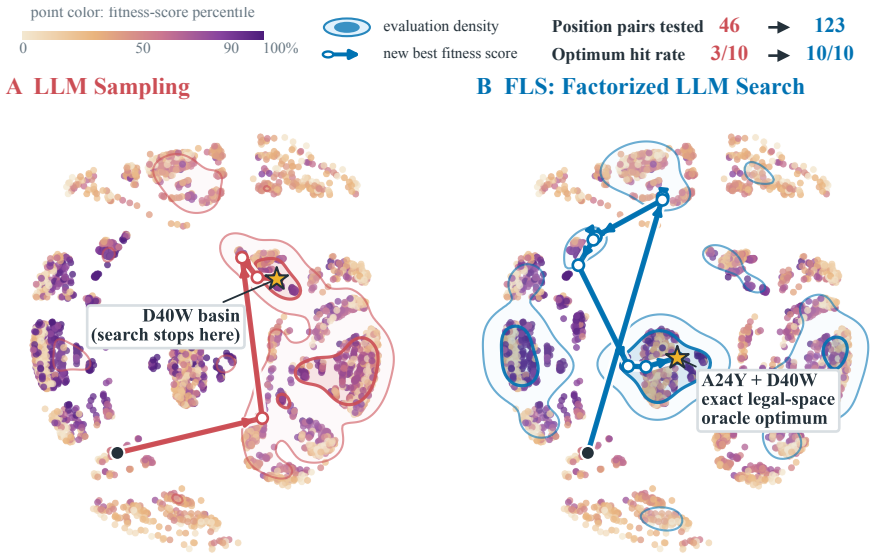


Figure 1: **On GB1, FLS tests 123 versus 46 distinct pairs of mutated sequence positions per run and finds the frozen-oracle optimum in 10/10 versus 3/10 runs.** Points are measured variants projected by Hamming-distance metric MDS; color denotes pooled fitness-score percentile, contours denote evaluation density, and arrows trace best-fitness updates in one representative matched run. A24Y/D40W is optimal for the frozen fitness ensemble over legal GB1 double mutants, not an experimentally established biological optimum. This is a whole-policy comparison with LLM Sampling; map construction and run selection are detailed in Appendix H.

failures of implicit exploration have also been observed for LLM agents in simpler bandit settings (Krishnamurthy et al., 2024).

A complementary target-blind probe asks whether an LLM can reliably realize a directed edit after an oracle-informed local cell has already been specified. Exact relocation remains difficult even within such selected cells (Figure 5). This supports the design rationale for separating allocation from generation: the controller makes the requested opportunity explicit, while the LLM remains an imperfect proposal mechanism.

We therefore introduce Factorized LLM Search (FLS), which separates *evaluation allocation* from *mutation generation*. Given previously measured variants, an explicit controller first selects a promising parent and a feasible local edit component; a frozen LLM then proposes the concrete amino-acid mutation within that request. We represent each local request as $z = (p, a, j, k)$, comprising a measured parent p, edit topology a, source position j, and coarse chemistry class k. Fitness directs evaluations toward promising parents, while the controller samples uniformly among source components that can execute the requested edit. The LLM is therefore responsible for proposing mutations within explicitly allocated opportunities rather than implicitly deciding how the entire evaluation budget should be distributed.

Across four protein fitness landscapes and matched 500-evaluation budgets, FLS achieves the highest endpoint mean on three of four tasks among the end-to-end baseline methods evaluated. Matched controls show that removing the local LLM proposal reduces performance on Ube4b and Bgl3. These results indicate that mutation generation and explicit evaluation allocation are distinct components of LLM-guided protein optimization.

Our contributions are:

- **Factorized LLM-guided protein optimization.** We separate explicit allocation of a limited evaluation budget from LLM mutation generation through a protein-edit interface over measured parents and feasible local components.

- **A diagnostic of direct LLM allocation.** Matched trajectories show that an LLM can continue generating novel amino-acid substitutions while repeatedly evaluating previously explored positional neighborhoods.
- **Controlled evaluation of generation.** Across four protein landscapes, a matched random-worker control tests whether LLM mutation proposals improve best-of-budget search under the same explicit allocation interface.

2 Problem Formulation and Design Prior

Let $x_0 \in \Sigma_{\text{aa}}^L$ be the wild type, $\mathcal{X} \subseteq \Sigma_{\text{aa}}^L$ the legal variant set, $f : \mathcal{X} \to \mathbb{R}$ the black-box fitness function, and B the evaluation budget. After observing $\mathcal{D}_t = \{(x_i, f(x_i))\}_{i<t}$, the objective is

$$\max_{\pi} \ \mathbb{E}_{\pi}\left[\max_{1\le b\le B} f(x_b)\right]. \tag{1}$$

We call a measured sequence selected as the source of an edit a *parent* p, and the resulting unevaluated sequence a *candidate variant* x'; "offspring" is reserved for evolutionary-algorithm baselines. The controller does not observe the fitness score of an unmeasured candidate variant. It chooses where to request a proposal, after which an LLM worker realizes the concrete edit.

The local action is either Refine, which changes the residue at an occupied position, or Relocate, which removes an occupied mutation and explores a new destination. We divide residues into three fixed classes: hydrophobic/aromatic (`AVILMFWY`), polar/flexible (`STNQCGP`), and charged/histidine (`DEKRH`). This partition is a weak local-smoothness prior, not an assumption that residues in one class have equal effects.

Design prior and claim boundary. The controller makes parent, action, source, and chemistry explicit before the worker proposes an edit. Evaluations create new parents and change the LLM's context, so these cells are not stationary bandit arms (Auer et al., 2002; Garivier & Moulines, 2011).

3 Factorized LLM Search

FLS separates search allocation from mutation generation. Its controller selects a measured parent and an executable local cell; a frozen LLM then realizes one concrete mutation within that cell. Refinement and relocation use the same interface. For each local query s, the fixed portfolio specifies an edit action a_s and parent regime g_s, while a persistent round-robin schedule specifies the requested chemistry class k_s. With $z = (p, a, j, \phi_t(p, a, j, k))$, the local proposal law decomposes into controller allocation and worker generation:

$$\pi_t^{\text{local}}(p, j, e \mid a, g, k) = \underbrace{\pi_{t,a,g}^{\text{par}}(p)}_{\text{parent allocation}} \ \underbrace{\pi_t^{\text{src}}(j \mid p, a, k)}_{\text{source allocation}} \ \underbrace{\pi_{\text{LLM}}(e \mid z, \mathcal{H}_t(z))}_{\text{edit proposal}}. \tag{2}$$

The action, parent regime, and requested chemistry class are scheduled rather than chosen by the LLM. The support map ϕ_t turns the requested class into an executable one. The compact context $\mathcal{H}_t(z)$ contains measurements relevant to that local request, not the full archive. Here π_{LLM} is the *effective* distribution of the first accepted edit after ranked LLM proposals and validation, not a token-probability model. We specify the two controller factors next, then give the operational sampling sequence; Appendix D derives the parent law and chemistry schedule.

3.1 Fitness score selects the parent

Let $g \in \mathcal{G} = \{\text{Exploit}, \text{Explore}\}$ index two parent-selection regimes with target effective sample sizes $\text{ESS}^{\star}_{\text{exploit}} < \text{ESS}^{\star}_{\text{explore}}$. For action a and regime g, let $\mathcal{P}_t(a)$ be measured parents with at least one executable local edit and let $\rho_P(p \mid a)$ be uniform on that set. We sample

$$\pi_{t,a,g}^{\text{par}}(p) \propto \rho_P(p \mid a) \exp\left(\frac{f(p)}{\tau_{a,g}}\right). \tag{3}$$

The temperature is chosen to reach a regime-specific effective sample size (ESS): the lower-ESS regime concentrates on the best measured parents, whereas the higher-ESS regime spreads probability

across several promising parents. The calibration derivation and numerical ESS values are given in Appendices E and G, respectively. Fitness score alone chooses the parent basin.

3.2 Uniform feasible-source selection

The controller cycles through the three chemistry classes in round-robin order. This simple stratification prevents a class from disappearing solely because of the LLM's proposal preference; it is not a learned class-value model. If the requested class is not executable for a source, a deterministic support rule maps it to an executable class or removes that source. The realized class, denoted $\widetilde{k}$, is used as the worker constraint.

After (p, a, k) is fixed, define $\mathcal{J}_t(p, a, k) = \{j : \phi_t(p, a, j, k) \neq \perp\}$. We then sample

$$\pi_t^{\mathrm{src}}(j \mid p, a, k) = \frac{\mathbf{1}\{j \in \mathcal{J}_t(p, a, k)\}}{|\mathcal{J}_t(p, a, k)|}. \tag{4}$$

Uniformity is conditional on the selected parent, action, requested chemistry, and executable support map; it is not uniform over all possible mutations.

3.3 The frozen LLM realizes the edit

The fixed action portfolio spans refinement and relocation under exploitative and broader parent distributions, together with an archive-only Recombine action. Recombination fits additive ridge regression to strictly earlier measurements and selects the best unseen legal combination in the measured-token closure; it cannot invent an unmeasured mutation. All matched controller and worker controls retain this action. We do not claim to isolate the contribution of every portfolio action. Exact batch allocation is given in Appendix C.

Once $(p, a, j, \widetilde{k})$ is selected, the frozen LLM receives the WT, parent, action, source component, allowed residues in class $\widetilde{k}$, legality constraints, and a compact summary of measured transitions matching the action, source, and chemistry class on the current or similar parents. The summary also identifies previously scored exact-parent edits to discourage repeats. The LLM does not receive the full archive. It returns an ordered list $L_s = (e_{s,1}, \ldots, e_{s,K})$ of $K = 3$ ranked edits; the validator scans L_s in order and evaluates the first legal, unseen, batch-unique candidate variant. Operationally, each local query samples the parent and source before requesting the concrete edit:

$$\begin{aligned} p_s &\sim \pi^{\mathrm{par}}_{t,a_s,g_s}(\cdot), \qquad j_s \sim \pi_t^{\mathrm{src}}(\cdot \mid p_s, a_s, k_s), \\ \widetilde{k}_s &= \phi_t(p_s, a_s, j_s, k_s), \qquad e_s \sim \pi_{\mathrm{LLM}}(\cdot \mid z_s, \mathcal{H}_t(z_s)), \end{aligned} \tag{5}$$

where $z_s = (p_s, a_s, j_s, \widetilde{k}_s)$. Archive-only recombination follows its separate measured-token rule rather than Equation 5. The exact portfolio, support rule, prompt fields, validation, and pseudocode are specified in Appendix C.

4 Experiments

Our evaluation first compares end-to-end search, then examines where direct LLM sampling spends its budget. A matched online control tests the LLM worker, while frozen-state probes delimit what this comparison can explain.

4.1 Setup and evaluation protocol

We optimize four frozen fitness-model landscapes: GB1 IgG-Fc binding (Olson et al., 2014), Ube4b ligase activity (Starita et al., 2013), Bgl3 activity/stability, and avGFP fluorescence (Sarkisyan et al., 2016). All scores come from task-specific frozen MLP ensembles used by the public RLXF workflow (Blalock et al., 2026; Romero Lab, 2025); no fitness model is updated during search. These are in-silico oracle evaluations, not wet-lab measurements. Appendix A documents the source DMS distributions, dataset splits, legal search spaces, and frozen reward-model configurations.

Every main arm uses ten runs, five candidates per synchronous batch, and 500 charged fitness evaluations. Matched FLS and LLM Sampling share the first five measured variants and block

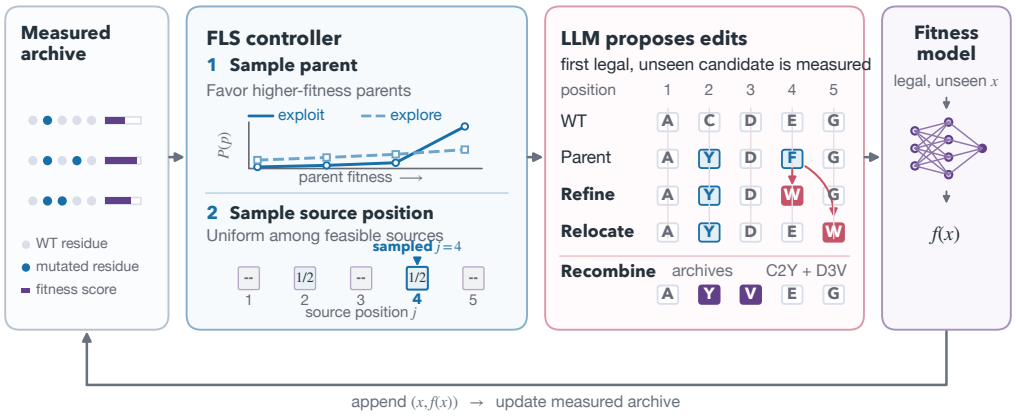


Figure 2: **FLS allocates edit opportunities; the frozen LLM supplies concrete mutations.** The controller chooses a parent by fitness score and samples uniformly among feasible source positions, then constrains a local edit; archive-only recombination is a separate action. Legal, unseen candidates are evaluated and returned to the measured archive. Gray/blue residues denote WT/mutated positions and purple bars denote fitness scores; all sequences and bars are schematic.

identities. GP, Kermut, ALDE, GA, Graph-GA, ESM-1v, and LLM-guided EA instead share ten DMS-supported initial measurements (Brooks et al., 2026); these are external, unpaired comparisons. Appendix G specifies the task spaces, initialization, model settings, and budget accounting.

We report best fitness score at query 500 for the end-to-end comparison and display the complete best-so-far trajectories. Paired bootstrap intervals for matched controls resample complete runs, not individual proposals.

4.2 Baselines and Causal Controls

LLM Sampling is the matched prompt-only diagnostic: it receives all measured variants and scores and proposes the next batch without an assigned parent, action, or component. Appendix B gives its complete batch and repair procedure. The w/o LLM proposal and Random Parent controls test the worker and parent-selection decisions while retaining the portfolio.

The external baselines span classical search, protein-model- and surrogate-guided optimization, and LLM-guided methods. GP and Kermut use the same charged ten-variant DMS starts and a 3,000-candidate DEAP/UCB loop; Kermut replaces the GP surrogate with its composite kernel, so it is an adapted rather than native Kermut search implementation. ALDE retains its DNN ensemble and Thompson-sampling components under the same candidate pool. GA, Graph-GA, ESM-1v, and LLM-guided EA retain their documented search or ranking rules with fixed-length sequence adapters and legal, unseen-candidate validation. Because these methods do not share the matched FLS initialization, their endpoint comparisons are descriptive.

4.3 End-to-End Best-of-Budget Performance

Table 1 reports the best fitness score after 500 evaluations. Among methods complete on the four main tasks, FLS has the highest endpoint mean on avGFP, GB1, and Bgl3; Kermut leads on Ube4b (8.070 versus 6.938). Figure 3 shows the corresponding trajectories. Rankings of unpaired means are descriptive, not paired superiority tests: external baselines and matched LLM arms start from different measured variants.

Table 1: **Best fitness score after 500 evaluations.** Mean ± sample SD over ten runs. Shading identifies the best two values per task. External DMS-initialized baselines are unpaired with FLS.

Method	avGFP	Ube4b	GB1	Bgl3
Classical search				
Random (2012)	3.715±.031	2.076±.357	1.908±.341	1.142±.172
GA (DEAP; 2012)	3.875±.024	4.777±1.029	2.995±.292	1.774±.234
Graph-GA (2019)	3.682±.052	2.336±1.532	2.571±.379	1.682±.140
Protein-model- and surrogate-guided methods				
GP (2013)	3.906±.024	5.557±.541	3.040±.151	1.809±.147
ESM-1v (2021)	3.563±.040	3.602±.000	1.164±.274	1.001±.214
Kermut (2024)	3.947±.026	**8.070±.223**	3.170±.404	2.008±.105
ALDE (2025)	3.835±.033	2.267±.896	1.685±.977	1.761±.224
GOLLuM (2026)	3.877±.019	4.352±1.406	2.867±.388	1.668±.045
LLM-guided methods				
LLM Sampling (ours)	3.966±.017	5.252±.879	2.835±.439	1.856±.286
BOPRO (2025)	3.919±.019	2.387±1.527	2.174±.689	1.224±.296
MOLLEO (2025a)	3.923±.017	4.463±1.607	2.930±.576	2.099±.308
LLM-guided EA (2025b)	3.978±.019	3.028±1.433	2.954±.572	1.687±.193
Expertise Evolution (2026)	3.863±.009	4.470±.431	2.081±.169	1.739±.108
PABLO (2026)	3.950±.015	4.777±.833	2.717±.514	1.898±.296
AlphaEvolve (2025)	3.903±.010	4.752±.942	2.241±.044	1.722±.196
FLS (ours)	**3.990±.013**	6.938±.400	**3.471±.000**	**2.110±.117**

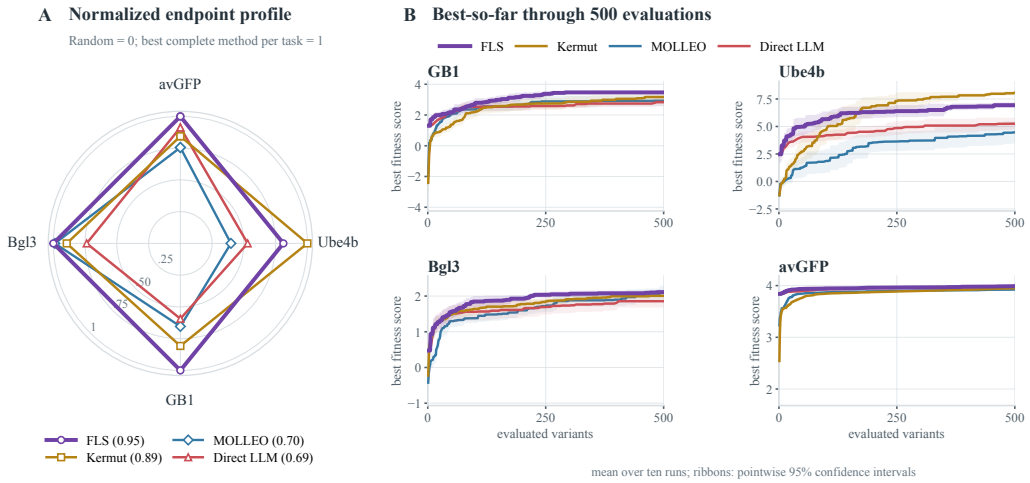


Figure 3: **FLS and the three strongest complete baselines have different strengths across four tasks.** Left: endpoints normalized between Random (0) and the best completed method (1) within each task; parentheses give equal-task macro-averages. Right: mean best-so-far fitness scores over ten runs with pointwise 95% confidence bands. FLS/Direct LLM Sampling share five-query prefixes, whereas Kermut and MOLLEO each use separate ten-query external prefixes; comparisons across these groups are unpaired.

4.4 Where does direct LLM sampling spend its budget?

Direct LLM Sampling sees every earlier measured GB1 variant and score, and its prompt explicitly instructs the model to *balance exploitation of high-fitness evidence with exploration of new mutation positions and amino-acid substitutions*. It is therefore a prompted, full-history control, not an unprompted LLM. Yet Figure 4 shows a split between two kinds of novelty. Over queries 201–500, 55.0% of its accepted candidates contain an exact amino-acid substitution not previously measured, but only 10.5% enter a previously unmeasured site pair. Under the same matched five-query prefixes, FLS reaches new site pairs in 18.9% of late evaluations while using the LLM inside controller-selected edit cells, but has fewer new exact substitutions (22.6%): the change is in *what* is explored, not a universal gain in novelty. Direct sampling spends 59.0% of late evaluations on site pairs measured at least five times whose previous best fitness is below the incumbent, versus 11.3% for FLS. These are accepted, evaluated variants rather than raw LLM completions. They diagnose the whole-policy budget allocation, not a causal effect of in-context examples or any single FLS component.

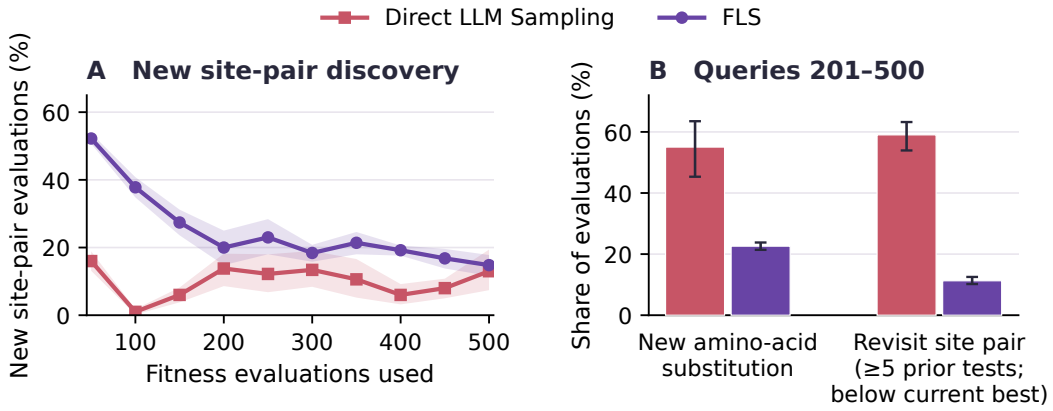


Figure 4: **On GB1, Direct LLM Sampling revisits well-tested, non-leading site pairs more often despite explicit explore/exploit prompting, while FLS discovers new pairs more often.** A: new site-pair evaluations per 50 queries. B: among queries 201–500, the share with a new exact substitution or revisiting a site pair with ≥ 5 earlier measurements whose prior best is below the incumbent; categories overlap. Bands and whiskers show 95% run-bootstrap intervals over ten matched runs using strict-prefix histories.

4.5 Does the controller or the LLM worker matter?

The w/o LLM proposal control replaces the local LLM proposals while retaining the controller-selected cells; Random Parent replaces the fitness-weighted parent law. In w/o position control, the controller retains parent and amino-acid-class control but does not select the source position. The LLM receives the complete measured history and proposes the concrete mutation, including its position and residue.

Table 2: **Matched online ablations at 500 fitness evaluations.** Values are q500 endpoint mean ± sample SD over ten runs with the same five measured initial variants; paired 95% intervals for w/o position control are reported in the text.

Method	avGFP	Ube4b	GB1	Bgl3
FLS	3.990±.013	6.938±.400	3.471±.000	2.110±.117
w/o position control	3.964±.014	6.436±1.024	2.744±.025	1.885±.127
w/o LLM proposal	3.992±.016	5.701±1.016	3.258±.371	1.873±.129
Random Parent	3.955±.012	6.427±1.005	2.771±.245	2.198±.180

The w/o LLM proposal control completed 40 audited q500 runs without adaptive LLM calls. Compared with that control, FLS improves endpoint fitness on Ube4b by $+1.237$ $[+0.650, +1.817]$ and Bgl3 by $+0.237$ $[+0.152, +0.332]$; the GB1 difference is $+0.213$ $[0.000, +0.443]$ and avGFP is unresolved at -0.002 $[-0.013, +0.008]$. These results support the conditional value of the local LLM worker on Ube4b and Bgl3. Removing position control lowers q500 on all four tasks. The paired differences relative to FLS are -0.0255 $[-0.0371, -0.0153]$ on avGFP, -0.502 $[-0.929, -0.089]$ on Ube4b, -0.727 $[-0.908, -0.454]$ on GB1, and -0.225 $[-0.355, -0.096]$ on Bgl3. Random Parent completes ten runs on all four tasks.

4.6 Can the interface enforce its edit request?

A matched frozen-state probe isolates proposal behavior: hard source enforcement increases valid-alternative source compliance by $+0.590$ $[+0.525, +0.655]$ without a resolved change in valid yield or one-step parent-relative gain. Source-scoped history has no supported gain over action-level history. Thus the interface controls proposal location, but the probe does not establish its independent contribution to sequential endpoint fitness (Appendix J).

Figure 5 motivates separating allocation from generation: directed edit realization remains difficult even after constraining (p, a, j, k).

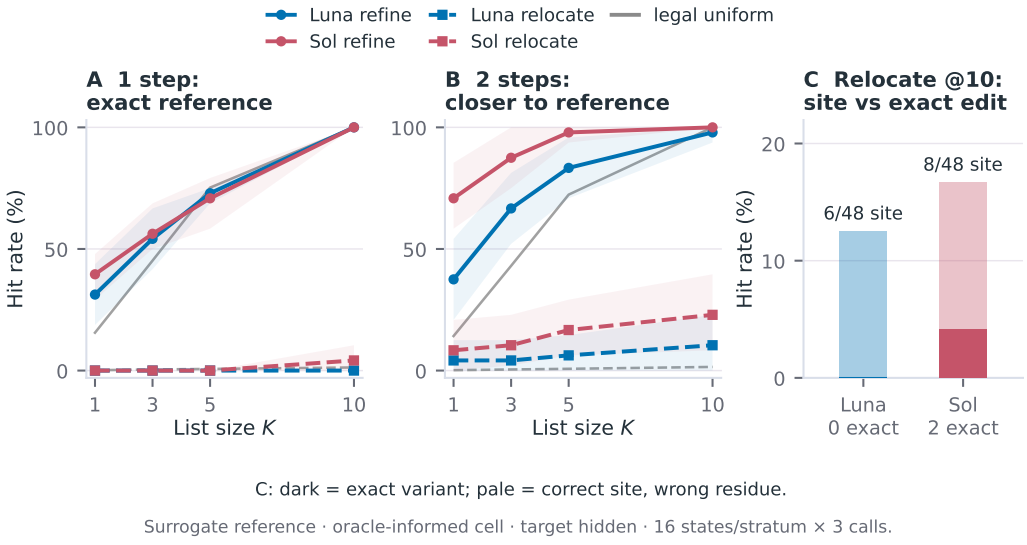


Figure 5: **A selected-cell probe exposes a distinct generation bottleneck.** Across four frozen-surrogate landscapes, 64 target-blind states span one- and two-step refinement and relocation; the frozen `gpt-5.6-luna` and `gpt-5.6-sol` workers each make three independent ranked-list calls per state. A: exact reference recovery from one step. B: distance reduction from two steps. C: one-step relocation at list size 10, separating the correct destination with the wrong residue from an exact hit. Gray lines show legal-uniform expectations, and ribbons show state-cluster bootstrap 95% intervals. Source and chemistry were selected with oracle information, while the destination, reference score, and measured history were hidden. Scores are from a frozen surrogate, not an assay; this is a proposal-only diagnostic rather than an online-search comparison.

At list size 10, the exact relocation reference is recovered in 0/48 and 2/48 calls, versus the correct destination in 6/48 and 8/48, showing that destination and residue choices can fail separately; refinement is less diagnostic because it has only 5–8 legal edits. FLS does not solve this: its controller fixes source j, while the worker still selects destination and residue. Rather, the probe motivates limiting LLM responsibility. Table 2 supplies complementary evidence: replacing the constrained worker with random legal edits hurts Ube4b and Bgl3. Together, the LLM is useful as a conditional proposer but too unreliable as an unconstrained allocation policy. This oracle-informed, frozen-surrogate probe does not establish an online-search advantage.

5 Related Work

Data-efficient protein optimization. Bayesian optimization, active learning, and model-guided directed evolution are established approaches for expensive protein evaluations (Romero et al., 2013; Wittmann et al., 2021; Cheng et al., 2022; Yang et al., 2025). Protein-specific methods include Kermut's uncertainty-aware kernel (Groth et al., 2024), ProteinNPT's non-parametric conditioning (Notin et al., 2023), and EVOLVEpro's protein-language-model embeddings with few-shot regression (Jiang et al., 2025). FLS instead considers a fixed general-purpose LLM as the mutation proposer and makes allocation of the fitness-evaluation budget explicit.

Language models inside search loops. LLMs can optimize from scored examples directly (Yang et al., 2024), serve as search operators within iterative selection and evaluation (Romera-Paredes et al., 2024; Ye et al., 2024), and propose protein variants from sequence histories and task descriptions (Wang et al., 2025b; Liu et al., 2026; Maus et al., 2026). These approaches motivate LLM-guided search but leave allocation of a limited protein-evaluation budget largely implicit.

Explicit search control. Recent methods increasingly separate generation from search control: BOPRO uses latent-space BO for example retrieval (Agarwal et al., 2025), LGBO incorporates LLM preferences into a surrogate (Yuan et al., 2026), LMABO selects among acquisition functions (Ngo

et al., 2026), and HOLLM partitions the search space before LLM sampling (Schwanke et al., 2026). Quality-diversity methods likewise allocate generation across predefined behavioral regions (Lim et al., 2024), while RosettaSearch controls structure-guided protein refinement (Kshirsagar et al., 2026). FLS instead allocates evaluations over feasible, parent-relative protein edit components while retaining the LLM as the mutation proposer, enabling allocation and generation to be tested separately under matched controls.

6 Discussion

Scope of the contribution. Our results support treating evaluation allocation and mutation generation as distinct problems in LLM-guided protein optimization. Direct LLM sampling can continue to generate novel amino-acid substitutions while concentrating evaluations within previously explored positional neighborhoods, showing that generative novelty alone does not ensure effective use of a limited evaluation budget. FLS makes this allocation decision explicit: the controller determines where to search, while the LLM determines which mutation to propose within the allocated opportunity. Across four protein fitness landscapes, FLS achieves the highest endpoint mean on three tasks among the end-to-end baselines in Table 1, and replacing the local LLM with random legal mutations reduces performance on Ube4b and Bgl3. Together, these results suggest that useful mutation generation and effective evaluation allocation are complementary rather than interchangeable capabilities.

Making allocation explicit does not imply that the current controller is universally optimal. The fitness-weighted parent law, uniform feasible-source rule, and stratified chemistry schedule are one auditable instantiation of the factorized interface. Alternative allocation policies can be studied or replaced without changing the mutation worker.

Limitations. Uniform feasible-source sampling is not calibrated uncertainty or an estimate of the value of unseen mutations, and the controller cannot recover edits outside the worker's proposal support. Additional frozen-state diagnostics of mutation generation within selected edit components are reported in Appendix I. The resulting policy has no guarantee of best-of-budget optimality under epistasis and history-dependent proposals. Our evaluations use frozen fitness models rather than wet-lab assays and match methods by fitness queries rather than total inference cost. Some external baselines use different initial measurements and are therefore compared descriptively rather than through paired tests. Finally, four protein landscapes and a single frozen LLM worker do not establish generalization to new proteins, experimental assays, or LLM backbones. These limitations motivate testing richer allocation policies and alternative mutation workers within the same factorized interface, ultimately under experimental feedback.

7 Conclusion

LLMs can propose useful protein mutations, but mutation generation and the allocation of scarce evaluations are separate optimization problems. FLS makes this distinction explicit by coupling a controller that allocates evaluations over feasible protein edits with a frozen LLM that realizes the mutations. Across four protein landscapes, FLS achieves the highest endpoint mean on three tasks among the end-to-end baselines in Table 1, while matched controls support a contribution from the LLM worker on Ube4b and Bgl3. Rather than establishing a universally superior allocation rule, these results provide an auditable framework in which where to search and what mutation to propose can be controlled and improved independently.

AI USE STATEMENT

Generative AI assisted with implementation and testing, log analysis, figure generation, literature search, and manuscript drafting. The method itself uses an LLM as specified above. The authors verified reported statistics against preserved histories and take responsibility for the manuscript.

REPRODUCIBILITY STATEMENT

Appendices A–G document the datasets and frozen reward models, complete algorithms, controller derivations and ESS calibration, canonical LLM templates, query accounting, validation, and exact-lazy recombination. The artifact preserves source-locked policies, run histories and proposals, trajectories, diagnostics, and figure scripts; every reported fitness score is linked to a full sequence and frozen-model checkpoint.

REFERENCES

Dhruv Agarwal, Manoj Ghuhan Arivazhagan, Rajarshi Das, Sandesh Swamy, Sopan Khosla, and Rashmi Gangadharaiah. Searching for optimal solutions with LLMs via bayesian optimization. In *International Conference on Learning Representations*, 2025.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002. doi: 10.1023/A:1013689704352.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012. URL `https://jmlr.org/papers/v13/bergstra12a.html`.

Nathaniel Blalock, Srinath Seshadri, Kensuke Nakamura, Agrim Babbar, Sarah A. Fahlberg, Ameya Kulkarni, and Philip A. Romero. Functional alignment of protein language models via reinforcement learning. *Nature Communications*, 2026. doi: 10.1038/s41467-026-77557-2.

Coban Brooks, Pascal Notin, and Philip A. Romero. Learning protein function through autonomous experimental interaction. *bioRxiv*, 2026. doi: 10.64898/2026.08.14.744985.

Lixue Cheng, Ziyi Yang, Changyu Hsieh, Benben Liao, and Shengyu Zhang. ODBO: Bayesian optimization with search space prescreening for directed protein evolution. *arXiv preprint arXiv:2205.09548*, 2022. doi: 10.48550/arXiv.2205.09548.

François-Michel De Rainville, Félix-Antoine Fortin, Marc-André Gardner, Marc Parizeau, and Christian Gagné. DEAP: A python framework for evolutionary algorithms. In *Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation*, pp. 85–92, 2012. doi: 10.1145/2330784.2330799.

Douglas M. Fowler and Stanley Fields. Deep mutational scanning: A new style of protein science. *Nature Methods*, 11(8):801–807, 2014. doi: 10.1038/nmeth.3027.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *Algorithmic Learning Theory*, pp. 174–188. Springer, 2011. doi: 10.1007/978-3-642-24412-4_16.

Peter M. Groth, Mads H. Kerrn, Lars Olsen, Jesper Salomon, and Wouter Boomsma. Kermut: Composite kernel regression for protein variant effects. In *Advances in Neural Information Processing Systems*, volume 37, pp. 29514–29565, 2024.

Jan H. Jensen. A graph-based genetic algorithm and generative model/monte carlo tree search for the exploration of chemical space. *Chemical Science*, 10(12):3567–3572, 2019. doi: 10.1039/C8SC05372C.

Kaiyi Jiang, Zhaoqing Yan, Matteo Di Bernardo, Samantha R. Sgrizzi, Lukas Villiger, Alisan Kayabolen, B. J. Kim, Josephine K. Carscadden, Masahiro Hiraizumi, Hiroshi Nishimasu, Jonathan S. Gootenberg, and Omar O. Abudayyeh. Rapid in silico directed evolution by a protein language model with EVOLVEpro. *Science*, 387(6732):eadr6006, 2025. doi: 10.1126/science.adr6006.

Akshay Krishnamurthy, Keegan Harris, Dylan J. Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? In *Advances in Neural Information Processing Systems*, volume 37, pp. 120124–120158, 2024. doi: 10.52202/079017-3818.

Meghana Kshirsagar, Allen Nie, Ching-An Cheng, Fanglei Xue, Rahul Dodhia, Juan M. Lavista Ferres, Kevin K. Yang, and Frank DiMaio. RosettaSearch: Multi-objective inference-time search for protein sequence design. *arXiv preprint arXiv:2604.17175*, 2026. doi: 10.48550/arXiv.2604.17175.

Bryan Lim, Manon Flageat, and Antoine Cully. Large language models as in-context AI generators for quality-diversity. *arXiv preprint arXiv:2404.15794*, 2024. doi: 10.48550/arXiv.2404.15794.

Xingyue Liu, Zijie Xing, Runze Wang, Luoming Hu, and Yanming Shen. Language-guided expertise evolution for protein optimization. In *ICLR 2026 Workshop on AI with Recursive Self-Improvement*, 2026.

Natalie Maus, Yimeng Zeng, Haydn Thomas Jones, Yining Huang, Gaurav Ng Goel, Alden Rose, Kyurae Kim, Hyun-Su Lee, Marcelo Der Torossian Torres, Fangping Wan, et al. Purely agent-driven black-box optimization for biological design. *arXiv preprint arXiv:2601.22382*, 2026. doi: 10.48550/arXiv.2601.22382.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29287–29303, 2021.

Giang Ngo, Dat Phan Trong, Dang Nguyen, Sunil Gupta, and Svetha Venkatesh. Adaptive acquisition selection for bayesian optimization with large language models. In *International Conference on Learning Representations*, 2026.

Pascal Notin, Ruben Weitzman, Debora Marks, and Yarin Gal. ProteinNPT: Improving protein property prediction and design with non-parametric transformers. In *Advances in Neural Information Processing Systems*, volume 36, pp. 33529–33563, 2023.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025. doi: 10.48550/arXiv.2506.13131.

C. Anders Olson, Nicholas C. Wu, and Ren Sun. A comprehensive biophysical description of pairwise epistasis throughout an entire protein domain. *Current Biology*, 24(22):2643–2651, 2014. doi: 10.1016/j.cub.2014.09.072.

Bojana Ranković, Ryan-Rhys Griffiths, and Philippe Schwaller. Large language models as uncertainty-calibrated optimizers for experimental discovery. *Nature Machine Intelligence*, 8:1466–1477, 2026. doi: 10.1038/s42256-026-01283-z.

Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, et al. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475, 2024. doi: 10.1038/s41586-023-06924-6.

Philip A. Romero, Andreas Krause, and Frances H. Arnold. 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.

Romero Lab. RLXF: Reinforcement learning from experimental feedback. GitHub repository, 2025. URL `https://github.com/RomeroLab/RLXF`. Accessed August 31, 2026.

Karen S. Sarkisyan, Dmitry A. Bolotin, Margarita V. Meer, Dinara R. Usmanova, Alexander S. Mishin, George V. Sharonov, Dmitry N. Ivankov, Nina G. Bozhanova, Mikhail S. Baranov, Onuralp Soylemez, et al. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603): 397–401, 2016. doi: 10.1038/nature17995.

Andrej Schwanke, Lyubomir Ivanov, David Salinas, Fabio Ferreira, Aaron Klein, Frank Hutter, and Arber Zela. Improving LLM-based global optimization with search space partitioning. In *International Conference on Learning Representations*, 2026.

Lea M. Starita, Jonathan N. Pruneda, Russell S. Lo, Douglas M. Fowler, Helen J. Kim, Joseph B. Hiatt, Jay Shendure, Peter S. Brzovic, Stanley Fields, and Rachel E. Klevit. Activity-enhancing mutations in an E3 ubiquitin ligase identified by high-throughput mutagenesis. *Proceedings of the National Academy of Sciences*, 110(14):E1263–E1272, 2013. doi: 10.1073/pnas.1303309110.

Haorui Wang, Marta Skreta, Cher-Tian Ser, Wenhao Gao, Lingkai Kong, Felix Strieth-Kalthoff, Chenru Duan, Yuchen Zhuang, Yue Yu, Yanqiao Zhu, et al. Efficient evolutionary search over chemical space with large language models. In *International Conference on Learning Representations*, 2025a.

Yinkai Wang, Jiaxing He, Yuanqi Du, Xiaohui Chen, Jianan Canal Li, Li-Ping Liu, Xiaolin Xu, and Soha Hassoun. Large language model is secretly a protein sequence optimizer. *arXiv preprint arXiv:2501.09274*, 2025b. doi: 10.48550/arXiv.2501.09274.

Bruce J. Wittmann, Yisong Yue, and Frances H. Arnold. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Systems*, 12(11):1026–1045.e7, 2021. doi: 10.1016/j.cels.2021.07.008.

Junhao Xiong, Ishan Gaur, Maria Lukarska, Hunter Nisonoff, Luke M. Oltrogge, David F. Savage, and Jennifer Listgarten. Property guidance for protein sequence generative models with ProteinGuide. *Nature Biotechnology*, 2026. doi: 10.1038/s41587-026-03207-z.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations*, 2024.

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. doi: 10.1038/s41467-025-55987-8.

Kevin K. Yang, Zachary Wu, and Frances H. Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694, 2019. doi: 10.1038/s41592-019-0496-6.

Haoran Ye, Jiarui Wang, Zhiguang Cao, Federico Berto, Chuanbo Hua, Haeyeon Kim, Jinkyoo Park, and Guojie Song. ReEvo: Large language models as hyper-heuristics with reflective evolution. In *Advances in Neural Information Processing Systems*, volume 37, pp. 43571–43608, 2024. doi: 10.52202/079017-1381.

Xinzhe Yuan, Zhuo Chen, Jianshu Zhang, Huan Xiong, Nanyang Ye, Yuqiang Li, and Qinying Gu. Unleashing LLMs in bayesian optimization: Preference-guided framework for scientific discovery. In *International Conference on Learning Representations*, 2026.

A Benchmark datasets and frozen fitness models

All four benchmarks use fixed-length wild-type sequences and task-specific deep mutational scanning (DMS) data distributed with the public RLXF workflow (Blalock et al., 2026; Romero Lab, 2025). Table 3 records the sequence length, wild-type score, ensemble size, legal WT-relative mutation range, and optimization objective. Scores are comparable only within a protein. The avGFP chromophore positions 65–67 remain fixed; the other tasks have no protected positions.

Table 3: **Optimization tasks and frozen fitness ensembles.**

Protein	Length	WT score	MLPs	Legal mutations	Objective
avGFP	238	3.4563	10	2–7	fluorescence brightness
Ube4b	102	−0.8546	10	2–4	ligase activity
GB1	56	−2.5904	3	exactly 2	IgG-Fc binding
Bgl3	501	−0.4778	10	1–2	enzyme activity / stability

The legal mutation ranges follow the central mass of each supplied source DMS distribution rather than a common radius across proteins. Table 4 reports the source sizes and the fraction of records inside each search range. Because mutation counts are discrete, the inclusive interval can contain more than exactly 90% of the source records.

Table 4: **Source DMS mass inside each legal search range.** Counts are transcribed from the mutation-count summaries supplied with the benchmarks.

Protein	Source variants	Search range	Fraction inside
avGFP	51,714	2–7	93.1%
Ube4b	98,297	2–4	96.7%
GB1	536,084	exactly 2	99.8%
Bgl3	26,653	1–2	97.0%

The frozen fitness checkpoints were produced with the public RLXF reward-model training implementation (Blalock et al., 2026). For avGFP and Ube4b, variants with one to four substitutions form the training regimes and five-substitution variants are split 75/25 into validation and test sets. For Bgl3, regimes one to four are used for training and the six five-substitution variants are split 75/25 for

validation and test. For GB1, single mutants are assigned to training; after randomly subsampling half of the double mutants, the retained double mutants are split 80/15/5. These split definitions are distinct from the legal search ranges in Table 3.

Every ensemble member is a one-hot sequence MLP with one 400-unit ReLU hidden layer, dropout 0.1, mean-squared-error loss, Adam with learning rate 10^{-6}, batch size 128, and at most 2,000 epochs. Table 5 summarizes the task-specific ensemble sizes and early-stopping settings. The checkpoints are frozen before optimization, and every method queries the same task-specific ensemble mean; no fitness-model fitting or updating occurs during a search trajectory.

Table 5: **Frozen fitness-model configurations.**

Setting	avGFP	Ube4b	GB1	Bgl3
Ensemble members	10	10	3	10
Hidden dimension	400	400	400	400
Dropout	0.1	0.1	0.1	0.1
Early-stop patience	400	400	200	200

B Direct LLM Sampling baseline

Algorithm 1 specifies the matched prompt-only baseline. It starts from the matched five-query history and asks one joint full-history prompt for five complete genotypes per round. The prompt specifies the global explore–exploit objective but no parent, action, position, chemistry class, or candidate role. In Algorithm 1, LLMGenerate encapsulates serialization of the full history, legality and novelty checks, and up to five repair calls that accumulate valid candidates. No search-policy fallback fills an incomplete batch.

Algorithm 1 Matched full-history Direct LLM Sampling

Require: legal space $\mathcal{X}$, budget B, initial history $\mathcal{D}_5$, batch size $b = 5$

$\mathcal{D} \leftarrow \mathcal{D}_5$
while $|\mathcal{D}| < B$ **do**
  $C \leftarrow \text{LLMGenerate}(\mathcal{X}, \mathcal{D}, b)$
  Evaluate C synchronously; append outcomes to $\mathcal{D}$
end while
return the best measured sequence

Candidates remain unmeasured until the complete batch is formed, so proposals within a batch never condition on one another's outcomes.

C Complete FLS algorithm

Algorithm 2 states the claimed method. The action portfolio is fixed, parent probabilities update after every measured batch, sources are sampled uniformly from the feasible set, and requested chemistry follows a round-robin schedule across the local queries. No action bandit, stagnation trigger, annealing schedule, or numbered-slot-specific proposal rule is active.

Algorithm 2 Factorized LLM Search (FLS)

Require: legal space $\mathcal{X}$, budget B, frozen LLM worker, slate size $K = 3$, ESS targets $\{\mathrm{ESS}_g^\star\}$

$\mathcal{D} \leftarrow$ five measured global proposals; $\ell \leftarrow 0$
while $|\mathcal{D}| < B$ **do**
  $C \leftarrow \varnothing$
  for $s = (g, a) \in \mathcal{G} \times \mathcal{A}$ **do**
    $k_s \leftarrow 1 + (\ell \bmod |\mathcal{K}|)$; $\ell \leftarrow \ell + 1$ ▷ chemistry schedule
    Calibrate $\tau_{a,g}$ to $\mathrm{ESS}_g^\star$; sample $p_s \sim \pi^{\mathrm{par}}_{t,a,g}$ (Eq. 3)
    Sample j_s uniformly among feasible sources; set $\widetilde{k}_s \leftarrow \phi_t(p_s, a, j_s, k_s)$ (Eq. 4)
    $L_s \leftarrow \mathrm{LLMRANKEDLIST}(p_s, a, j_s, \widetilde{k}_s, \mathcal{H}_t, K = 3)$
    $x_s \leftarrow \mathrm{FIRSTVALIDEDIT}(L_s)$; $C \leftarrow C \cup \{x_s\}$
  end for
  $C \leftarrow C \cup \{\mathrm{RIDGERECOMBINE}(\mathcal{D})\}$
  Evaluate C synchronously; append outcomes to $\mathcal{D}$
end while
return the best measured sequence

No candidate in a batch conditions on another candidate's unobserved fitness score. No fitness oracle score is used to rerank the three LLM proposals. Retries and unexecuted alternatives do not consume fitness-query budget. FIRSTVALIDEDIT scans the ordered list for the first legal, globally unmeasured, batch-unique variant and applies the fixed legal retry/fallback procedure only if none survives.

D Derivations and deterministic schedules for the factorized policy

Fitness-only parent law. For fixed (a, g), define

$$\mathcal{L}_P(q) = \mathbb{E}_{p\sim q}[f(p)] - \tau_{a,g} D_{\mathrm{KL}}(q \,\|\, \rho_P). \tag{6}$$

Let $Z_P = \sum_p \rho_P(p \mid a) \exp(f(p)/\tau_{a,g})$ and let $q^*(p) = \rho_P(p \mid a) \exp(f(p)/\tau_{a,g})/Z_P$. Then

$$\mathcal{L}_P(q) = \tau_{a,g} \log Z_P - \tau_{a,g} D_{\mathrm{KL}}(q \,\|\, q^*). \tag{7}$$

Therefore Equation 3 is the unique optimizer on the support of ρ_P.

Deterministic chemistry schedule. Let $\mathcal{K} = \{1, \ldots, C\}$ index the fixed chemistry classes, in the order specified by the implementation, and index local proposal requests by a persistent counter ℓ that is not reset at batch boundaries. The requested class is simply

$$k_\ell^{\mathrm{rr}} = 1 + (\ell \bmod C), \qquad \ell = 0, 1, \ldots. \tag{8}$$

This is a schedule, not the solution of an allocation objective. If $N_L(k)$ denotes the number of requests assigned to class k among the first L local proposals, then the cyclic schedule gives

$$\max_{k,k'} |N_L(k) - N_L(k')| \le 1 \qquad \text{for every prefix } L. \tag{9}$$

Thus chemistry coverage is balanced over the query stream without a learned class value, posterior, or optimization problem. This guarantee is for requested classes. The realized classes can differ through the deterministic support map

$$\phi_t(p, a, j, \cdot) : \mathcal{K} \to \mathcal{K} \cup \{\perp\}, \qquad \widetilde{k} = \phi_t(p, a, j, k), \tag{10}$$

where $\perp$ denotes an infeasible source. The map retains k when executable, returns another executable class when inherited support repair is required, and returns $\perp$ when no legal class is available.

E ESS calibration and limiting cases

The fitness-Boltzmann parent law is

$$q_\tau(i) = \frac{\exp(y_i/\tau)}{\sum_j \exp(y_j/\tau)}, \tag{11}$$

where $y_i = f(p_i)$. Let $\beta = 1/\tau$ and $Z(\beta) = \sum_i \exp(\beta y_i)$. Then

$$\mathrm{ESS}(q_\beta) = \frac{Z(\beta)^2}{Z(2\beta)} \tag{12}$$

and

$$\frac{d}{d\beta} \log \mathrm{ESS}(q_\beta) = 2\mathbb{E}_{q_\beta}[y] - 2\mathbb{E}_{q_{2\beta}}[y] \leq 0. \tag{13}$$

Thus ESS is non-decreasing in temperature. Under $y'_i = ay_i + b$ with $a > 0$, choosing $\tau' = a\tau$ preserves the normalized law.

In FLS, the numerical solver calibrates the fitness-only law in Equation 3 and records requested, reachable, and achieved ESS. Neither chemistry feasibility cardinality nor source selection changes the parent mass.

The parent policy contains useful limiting cases:

- $\tau_{a,g} \to 0$: concentration on highest-fitness executable parents;
- $\tau_{a,g} \to \infty$: the uniform executable-parent reference;

F CANONICAL LLM PROMPT TEMPLATES

Figures 6 and 7 show canonicalized templates for the two LLM interfaces used in the matched comparison. Angle-braced fields are populated at runtime from the task configuration, measured history, or controller assignment. Whitespace and field names are normalized for readability; candidates are unmeasured when either prompt is issued, and both interfaces require structured output that is checked by deterministic code. The shared five-query initialization uses the task and legality fields in Figure 6 with an empty measured history and requests five globally diverse legal variants.

Prompt for Direct LLM Sampling

```
SYSTEM:
You are a protein engineer optimizing <PROTEIN>. Propose exactly
five distinct legal variants. Use only observed fitness scores;
the variants requested in this call are unmeasured.

USER:
wild_type_sequence: <WT_SEQUENCE>
mutation_range: <MIN, MAX>
protected_positions: <POSITIONS OR EMPTY>
measured_history: <ALL EARLIER VARIANTS AND SCORES>
objective: maximize the best observed fitness within the budget
search_guidance:
  balance exploitation of high-fitness evidence with exploration
  of new mutation positions and amino-acid substitutions
rules:
  - fixed length; canonical substitutions only
  - OldAA and one-based positions must match the wild type
  - obey the mutation range and protected positions
  - exclude measured and within-batch duplicate variants
output:
  variants: [{mutations: [...], rationale: ...}, ...]
```

Figure 6: Canonical full-history prompt for Direct LLM Sampling. The prompt assigns no parent, action, source position, chemistry class, or candidate role.

Prompt for the FLS local worker

```
SYSTEM:
You are a local protein-mutation worker. Return three ranked,
distinct parent-relative edits that satisfy the assigned contract.
Do not change the parent, action, source, or chemistry assignment,
and do not invent a fitness score for an unmeasured candidate.

USER:
wild_type_sequence: <WT_SEQUENCE>
parent_mutations: <MEASURED PARENT>
legal_mutation_range: <MIN, MAX>
protected_positions: <POSITIONS OR EMPTY>
contract:
  action: <REFINE OR RELOCATE>
  source_component: <SELECTED SOURCE>
  chemistry_class: <REALIZED EXECUTABLE CLASS>
cell_history:
  promising_directions: <AT MOST 2>
  discouraged_directions: <AT MOST 3>
  scored_parent_relative_deltas: <AT MOST 12>
  recent_unmeasured_destinations: <RELOCATION ONLY>
rules:
  - obey the assigned topology, source, and chemistry class
  - exclude protected, measured, and batch-reserved variants
output:
  ranked_edits: [{parent_relative_edit: ..., rationale: ...}, ...]
```

Figure 7: Canonical cell-scoped prompt for the FLS local worker. If no ranked alternative passes validation, the worker is shown the validation errors and asked only for the remaining alternatives; deterministic fallback uses the same hard contract.

The controller reconstructs each candidate from WT and the assigned parent, checks source residues and the canonical amino-acid alphabet, enforces protected positions, mutation-count limits, action topology, source component, and chemistry class, and rejects measured or batch-reserved variants. The first valid ranked alternative is used. If none survives, only that worker is retried with its validation errors; deterministic fallback enumerates legal unseen edits under the same contract.

G Protocol and accounting

Task spaces and worker. The legal wild-type-relative substitution counts are exactly two for GB1, two to four for Ube4b, one to two for Bgl3, and two to seven for avGFP; task-specific protected positions and fitness-model checkpoints are fixed by the released run manifests. The frozen LLM worker is `gpt-5.6-luna`; it uses the scoped prompt contract in Appendix F and is not updated from search outcomes.

Table 6: **Fixed hyperparameters of the evaluated FLS policy.** Regime names and symbols are used in the main text; numerical settings are collected here.

Quantity	Symbol	Value
Exploit parent target ESS	$\mathrm{ESS}^{\star}_{\mathrm{exploit}}$	1
Explore parent target ESS	$\mathrm{ESS}^{\star}_{\mathrm{explore}}$	4
Source sampling	π^{src}	Uniform among feasible sources
Parent regimes	$\mathcal{G}$	{Exploit, Explore}
Local edit topologies	$\mathcal{A}$	{Refine, Relocate}
Ranked edits per LLM call	K	3
Recombination candidates per batch	—	1
Batch size	—	5
Total fitness-query budget	B	500

Budget and batches. Each trajectory contains 500 unique finite fitness evaluations: five global LLM proposals followed by 99 synchronous batches of five. There are ten runs per protein and no free post-search oracle query.

Parent and action laws. Every complete adaptive batch instantiates the Cartesian product $\mathcal{G} \times \mathcal{A}$ and adds one recombination candidate. For each local action, the exploit and explore channels target $\mathrm{ESS}^{\star}_{\mathrm{exploit}}$ and $\mathrm{ESS}^{\star}_{\mathrm{explore}}$, respectively, under Equation 3. Channels are shuffled in presentation order; no numbered candidate has a distinct policy. Table 6 reports the numerical targets and remaining fixed settings.

Chemistry classes. The three classes are hydrophobic/aromatic (`AVILMFWY`), polar/flexible (`STNQCGP`), and charged/histidine (`DEKRH`). Requested classes follow the persistent cyclic schedule in Equation 8; the cycle continues across batch boundaries. This provides uniform long-run class coverage without fitting or optimizing a chemistry-allocation policy. For each candidate source, the unchanged support function retains the requested class when executable and otherwise projects to an executable class. Across the forty completed FLS runs, the map changes the requested class in 262 of 15,840 local queries (1.65%).

Uniform feasible-source rule. Given each fixed parent, action, and requested class, FLS evaluates executable support for every source, records the realized class $\widetilde{k}(j)$, and samples the source uniformly from Equation 4. After selecting j, FLS sends the realized class $\widetilde{k}(j)$ to its worker; local history is indexed by that class and current within-batch reservations remain excluded.

Recombination. Additive ridge fits all strictly earlier measurements,

$$\widehat{f}(M) = \widehat{\beta}_0 + \sum_{c \in M} \widehat{\beta}_c, \tag{14}$$

where c is an exact measured mutation token. The implementation retains one measured representative residue per position and returns the highest-ranked unseen legal subset. An exact lazy heap preserves additive ranking and deterministic tie breaking without materializing the full closure.

Compute accounting. The primary budget is fitness evaluations, not LLM tokens. Final reporting must include requests, input/output tokens, retries, wall time, and fitness evaluations separately for every method.

H VISUALIZATION PROTOCOL

Figure 1 projects all 2,849 unique GB1 variants measured by FLS or LLM Sampling across ten matched 500-query runs. A fixed-seed, two-dimensional metric MDS fit uses normalized pairwise Hamming distances between aligned 56-residue sequences; the point color is each variant's fitness-score percentile within the pooled measured set. Gaussian-kernel contours show query density, not fitness: the outer and inner levels contain approximately 82% and 42% of each method's 5,000 evaluations. The arrowed path shows only incumbent updates in the matched run whose paired trajectory difference is closest to the median. The displayed site-pair counts are rounded means; the unrounded mean $\pm$ sample SD is 46.1 ± 8.1 for LLM Sampling and 123.0 ± 7.7 for FLS. A24Y/D40W maximizes the frozen three-model fitness ensemble over all 555,940 legal GB1 double mutants. The map is descriptive: its contours are not fitness contours, and a 2D projection does not imply that the full landscape is two-dimensional.

In Figure 2, archive rows and the five-position edit grid are schematic; a sequence may contain multiple mutations. The highlighted source $j = 4$ is one stochastic draw. Bar height and darkness encode more distinct evaluated directions at a source, but do not affect the uniform source probabilities.

Figure 4 uses the same ten matched GB1 histories as Figure 1. It analyzes accepted, paid variants, not rejected LLM outputs. At the start of each five-query batch, a site pair is new if no strictly earlier measured variant has the same two mutated positions; an exact substitution is new if its WT-relative residue change has not been measured. A heavily revisited non-incumbent pair has at least five earlier measurements and its previous best fitness is below the global incumbent at that batch start. Panel A groups new-site-pair indicators into 50-query bins over the full budget; panel B uses queries 201–500. Its two indicators are not disjoint. Intervals bootstrap ten matched runs 20,000 times. This is a whole-policy trajectory comparison, not a randomized test of whether in-context demonstrations cause repeated site-pair choices.

I FROZEN-STATE INTERFACE PROPOSAL AUDIT

To separate source enforcement from context scoping, we preregistered a compact three-arm proposal probe before any additional call. At twenty fixed q200 states, the probe reuses eighty local parent/action/chemistry assignments and compares free source choice with action-level context, a uniformly sampled hard source with the same action-level context, and the same hard source with source-scoped context. All 240 calls returned without retry, and no proposal entered an online history. Paired state-clustered bootstrap effects are summarized below:

Contrast	Source compliance	Valid yield	Parent-relative gain
Hard − free	$+0.590$ $[+0.525, +0.655]$	$+0.042$ $[-0.025, +0.113]$	$+0.115$ $[-0.133, +0.364]$
Scoped − action-level	0 by construction	-0.033 $[-0.096, +0.025]$	-0.125 $[-0.453, +0.226]$

For scoped minus action-level context, the exact-repeat-rate effect is $+0.046$ $[-0.013, +0.100]$. The probe therefore supports hard source control as a behavioral mechanism but does not support scoped context as a general anti-repetition or fitness-score mechanism. Frozen-oracle gains are one-step diagnostics and do not identify q500 value.

J SCOPE OF THE MATCHED CONTROLS

The w/o LLM proposal control replaces the four adaptive local LLM calls with uniformly shuffled legal edits inside the selected cells while holding the remaining search policy fixed. Its forty runs have 500 unique finite evaluations and zero new LLM SDK requests. The paired endpoint effects for FLS minus the random worker are $+0.213$ $[0.000, +0.443]$ on GB1, $+1.237$ $[+0.650, +1.817]$ on Ube4b, $+0.237$ $[+0.152, +0.332]$ on Bgl3, and -0.002 $[-0.013, +0.008]$ on avGFP. These are task-specific effects of the local worker.

The w/o position control ablation retains parent and amino-acid-class control but removes the controller-selected source position. Its LLM receives the complete measured history and proposes the concrete mutation, including its position and residue.

The hard-source and history-scoping contrasts are frozen-state proposal probes, not online q500 ablations. They establish that a hard source constraint changes proposal compliance but do not attribute endpoint gains to prompt scoping. The complete run is the inferential unit for the online comparisons, and paired intervals require matched initialization and block identity.

K LIMITS OF THE THEORETICAL INTERPRETATION

The reward-only parent Gibbs identity, low-discrepancy chemistry schedule, deterministic support map, and fitness-only ESS properties are exact for their respective stated controllers. Uniform feasible-source selection, parent ESS targets, the fixed action portfolio, and the chemistry partition are design choices. The portfolio, including archive recombination, is held constant in the matched worker control; the present evidence does not establish the necessity of each action separately.

The reward-only parent objective formalizes fitness-directed basin selection. The source controller samples uniformly among feasible sources. The chemistry rule is instead the minimum-discrepancy integer realization of a uniform requested-class reference. The support map can change the realized class, so uniformity is claimed for requested allocations rather than executed chemistry. These identities prove the unique optimizers of the stated surrogate problems; they do not prove that the resulting sequential policy is Bayes-optimal or regret-optimal for protein search. Finally, Equation 6 does not solve the sequential extreme-value objective in Equation 1. A performance theorem would require assumptions about epistasis, the history-dependent LLM worker, delayed descendants, the support map, and candidate support. We do not impose those assumptions. The mathematics specifies and explains the policy; prospective experiments determine whether it improves best-of-budget search.