Rethinking Directed-Evolution Benchmarks: Top-Variant Recovery Under a Fixed Budget

Anonymous authors
Paper under double-blind review

Abstract

Machine learning-guided directed evolution improves a protein over successive rounds of experimental screening. When variants must be assayed individually, as is common for catalytic activity, each round tests only tens of variants. Existing benchmarks such as ProteinGym reward ordering an entire mutational landscape rather than recovering the best variants under such a budget. We introduce **EvolveGym**, a benchmark of measured deep mutational scanning data that scores top-variant recovery under a fixed small screening budget, over three tasks drawn from practice: identifying beneficial single mutations, assembling higher-order multi-mutants, and searching a site-saturated library. It spans 487 single-substitution assays, 96 of them functional rather than stability assays, 44 multi-mutant landscapes, and 15 combinatorially complete enzyme-activity and binding landscapes. We further argue that variant selection is better posed as ranking variants against one another than as regressing their measured values, which are noisy and condition-dependent, and propose **EvolveRank**: it turns n measurements into all $n(n-1)/2$ pairwise comparisons and aggregates a learned preference classifier into a global ranking. Benchmarking 17 methods at a matched budget of 80 measurements, EvolveRank places 77.5% of its 16 nominations in the true top 5% of each single-mutation assay, against 41.0% for the strongest few-shot supervised baseline; with representation and budget held fixed, ranking adds nine points of precision over regression. On multi-mutant landscapes the winner is predictable from the landscape itself: additive models lead under near-additivity, EvolveRank under strong epistasis, and an ensemble of the two needs no prior knowledge of the regime.

1 Introduction

Protein engineering aims to improve a measurable property—catalytic activity, binding affinity or stability—by mutating a protein's sequence. The number of candidate variants grows exponentially with sequence length and mutational depth (number of mutations), so only a vanishing fraction can ever be measured. Directed evolution proceeds by iteration: each round screens a small library and carries the best variants forward into the next round of mutagenesis (Romero and Arnold, 2009; Arnold, 2019).

Machine learning promises to cut the experimental cost by predicting which variants are worth measuring (Wittmann et al., 2021); two families of methods have emerged. Zero-shot methods score variants from large-scale unlabeled sequence data alone; protein language models, whose sequence likelihoods correlate with fitness, are the leading example (Meier et al., 2021; Hie et al., 2024). Supervised methods instead learn from measurements of the target protein itself, either from a fixed set of labeled variants or by choosing which variants to measure next (active learning), which is necessary because evolutionary likelihood is an imperfect proxy for engineering utility: it reflects natural selection rather than the property being optimized (Jiang et al., 2024; Hou et al., 2026; Gordon et al., 2024; Ding and Steinhardt, 2024). A campaign makes one decision, over and over: which handful of variants to build next. But existing benchmarks do not score methods on that decision, and existing methods are not trained to make it. These are the two limitations we address.

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.

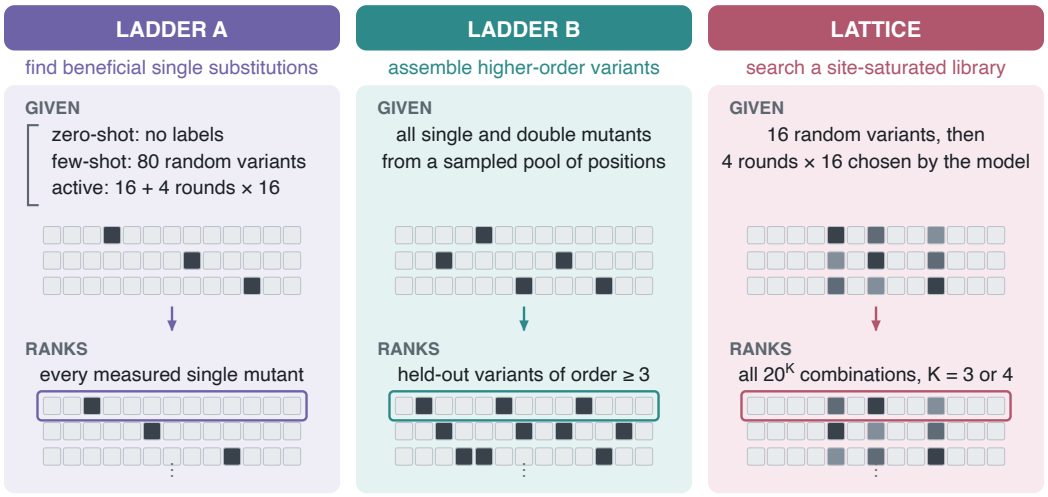

Figure 1: **The three tasks of EvolveGym.** Each row of squares is one variant of optimized protein, with dark squares marking mutated positions; in Lattice the mutations fall at the same positions in every variant, the shade standing for a different amino acid. Every column states what a method is *given* and what it must *rank*. In every task the method ranks all candidate variants and is scored on the 16 it nominates, outlined in red, using the metrics of Table 1. Ladder A compares three regimes at a matched budget of 80 measured variants, Ladder B receives all single and double mutants up front with no acquisition, and Lattice runs a closed loop over a combinatorially complete library.

The first is that a practitioner cannot tell from the literature which method to use. Established benchmarks score how well a model orders an entire mutational landscape (Notin et al., 2023), so a method can rank well overall yet place the true best variants just below the cutoff, where they are never tested; their supervised tracks also train on far more labels than a campaign has. The papers that do evaluate under campaign-like budgets each use their own handful of landscapes, so their results do not compose (Li et al., 2025; Yang et al., 2025; Ibarraran et al., 2026).

The second is that nearly all of these methods fit a regressor to measured property values and then select the top-scoring variants. Yet an assay often returns a different number for the same variant on a different day, plate or laboratory, while the relative ordering of variants is far more stable (Faure et al., 2020; Kroll et al., 2023; Markin et al., 2021). This is most prominent for functional readouts on arbitrary scales. We return to this in Section 5, where the gain from ranking is largest in exactly the functional categories. Since selection consumes only the ordering, and the ordering is what survives the assay, learning it directly is the better matched objective.

Contributions. To address these two limitations, this work makes three contributions. First, we introduce **EvolveGym**, a benchmark of measured deep mutational scanning data that scores the recovery of the best variants under a small, fixed budget across three tasks drawn from practice: finding beneficial single mutations (Ladder A), assembling higher-order multi-mutants from them (Ladder B), and searching a site-saturated library (Lattice), described in Section 3. Second, we propose **EvolveRank**, which learns a pairwise preference classifier from the comparisons implied by the measured variants and aggregates them into a global ranking with a Bradley–Terry model (Section 4). Third, we benchmark 17 methods under a matched budget of 80 measurements: EvolveRank finds the single best variant in 37.0% of assays against 18.7% for the strongest supervised model, and improves over its regression-based baseline on every metric at a matched representation (top-5% precision 71.6% vs. 62.5%).

2 Related Work

Benchmarks. The most widely used benchmark for protein fitness prediction is ProteinGym (Notin et al., 2023), which aggregates 217 substitution DMS assays into a zero-shot

and a supervised track. Both score a model over an entire assay, by Spearman correlation, NDCG, AUC, MCC and Top-K recall; even the most selective of these, Top-K recall at the top decile, rewards hundreds to thousands of variants, far more than a campaign can build, and the supervised track trains on roughly 80% of each assay under five-fold cross-validation. FLIP and FLIP2 (Dallago et al., 2021; Didi et al., 2026), TAPE (Rao et al., 2019), PEER (Xu et al., 2022) and CombinGym (Chen et al., 2026) likewise report whole-set rank correlation. FLEXS (Sinai et al., 2020) evaluates closed-loop exploration under budget and round constraints, but almost all of its landscapes are *in-silico* oracles rather than measurements. Conversely, Li et al. (2025) does adopt a campaign-style budget-limited metric on real measurements, but only over its own 16 predominantly TrpB libraries, all small site-saturated combinatorial landscapes, so it never tests identification of beneficial single mutations or the assembly of multi-mutants across a protein. Appendix A assesses each of these benchmarks against the three things a directed-evolution comparison needs — real measurements, coverage of all three mutational regimes, and a budget-limited top-variant metric — and none satisfies all three (Table A.1).

Selecting variants to measure. Methods differ mainly in which part of sequence space they search. *Scan-and-combine* approaches first rank single mutations, by measurement, by a zero-shot model, or both, and build only combinations of the top-ranked ones: from greedy accumulation of hits (Cui et al., 2021) and language-model-guided antibody maturation (Hie et al., 2024) to EVOLVEpro, which iteratively trains a top-layer regressor on language-model embeddings of the few measured variants and re-ranks the remainder (Jiang et al., 2024). MULTI-evolve extends this by measuring all pairwise combinations of beneficial single mutations and modeling their interactions to extrapolate to higher order (Tran et al., 2026), although a reanalysis reports that an additive model correlates almost perfectly with MULTI-evolve's multi-mutant predictions (Visani et al., 2026). *Site-saturation* approaches instead fix a few positions, usually near the active site, and search all amino-acid combinations there (Reetz and Carballeira, 2007): MLDE measures a sample of the library and trains a model to rank the remainder (Wu et al., 2019), refined by zero-shot-guided sampling (Wittmann et al., 2021), by library co-optimization (Ding et al., 2024), and by rounds of active learning (Yang et al., 2025). Across both protocols the model is fit to absolute measured values and the top-scoring variants are then selected.

Ranking objectives. Ranking is not a new idea in protein fitness modeling, but it has so far been used to fine-tune large pretrained models rather than to rank candidate variants directly. FSFP combines meta-transfer learning, a learning-to-rank loss and parameter-efficient fine-tuning to adapt a language model from tens of labeled mutants (Zhou et al., 2024), and (Lee et al., 2023) fine-tune language models with a fitness-ranking loss; both yield a pointwise score that then enters the usual predict-then-select loop. Closest to our position is ERA (Ibarraran et al., 2026), which post-trains a generative language model on all pairwise preferences among the measured variants using an energy-rank-alignment objective (Chennakesavalu et al., 2025) related to DPO (Rafailov et al., 2023), and iterates over four rounds. ERA shares our premise that experimental rankings are the dependable signal, but differs in three respects: it updates a 1.4B-parameter model by gradient descent, it requires the mutated positions to be fixed in advance, and it was developed at 96 measurements per round and 384 in total. We evaluate ERA and DPO on Lattice, the only one of our three tasks their formulation addresses, and both fall well short of the active-learning baselines in the low-data regime of 16 measurements per round.

Rank aggregation from pairwise comparisons. Recovering an order from noisy pairwise comparisons is itself a mature problem, from the Bradley–Terry model (Bradley and Terry, 1952) and its Plackett–Luce generalization to spectral estimators such as Rank Centrality (Negahban et al., 2015), and to active schemes that allocate comparisons so as to identify the top-K items rather than the full order (Mohajer et al., 2017). That last setting is directed evolution almost exactly: comparisons are expensive, and only the top of the order is ever used.

Landscape structure. Finally, how hard a landscape is to search depends on its structure. GraphFLA (Huang et al., 2025) characterizes measured fitness landscapes by ruggedness statistics, among them the fraction of reciprocal-sign epistatic motifs, pairs of mutations

whose individual effects reverse sign in each other's presence (Wu et al., 2016). We use this measure to stratify our multi-mutant results.

3 EvolveGym: Benchmark for ML-guided directed evolution

No experimental assay offers both broad positional coverage and complete measurement of the combinations those positions allow, so EvolveGym spans the trade-off with three tasks (Figure 1). Ladder A asks which single substitutions are worth making, Ladder B how to combine them, and Lattice how to search a few positions exhaustively. All three are scored on the same question: how many of the best variants a method nominates within a fixed budget.

3.1 Tasks and evaluation protocols

Ladder A: beneficial single substitutions. The first task is the one that opens a campaign: deciding which single substitutions to measure. A method ranks every single-point variant measured in the assay and nominates its top 16. We compare three regimes: zero-shot methods use no labels; supervised few-shot methods are trained on 80 randomly drawn variants, of which we also report the first 16; and active learning begins from those same 16 and acquires a further 64 over four rounds of 16. A method may nominate variants it has already measured, so that an active learner is credited for a top variant it finds in an early round. Zero-shot scoring is deterministic; the other two regimes are averaged over 10 independent simulations per dataset. Ladder A is evaluated on 487 deep mutational scanning assays.

Ladder B: assembling higher-order variants. The second task is the step that follows: given measurements of single and double mutants, decide which higher-order combinations are worth building. A method is trained on every measured single mutant in a sampled pool of single mutations together with every double mutant both of whose constituents lie in that pool, and must rank the held-out variants of mutational order three and above, again nominating 16. Unlike the other two tasks, Ladder B has no measurement rounds: every method sees the same training data and never chooses what to measure next, so it tests generalization from single and double mutants to higher-order combinations alone. Each landscape is split 10 times independently, with pools grown until at least 100 variants of order three or more have all of their constituent single mutations inside the pool. Ladder B is evaluated on 44 multi-mutant landscapes.

Lattice: searching a site-saturated library. The third task is the site-saturation campaign: a few positions are chosen in advance and every amino-acid combination at them has been measured, so a method can be run as a closed loop rather than on a fixed split. A campaign starts from 16 variants drawn uniformly at random and runs for four rounds, each revealing the true fitness of the method's top 16 not-yet-measured predictions, for 80 measurements in total; each campaign is repeated 10 times. Because at least 93% of each landscape is measured, almost every proposed variant has a known fitness, so performance reflects how well a method searches rather than whether its picks happen to be measured. Lattice is evaluated on 15 combinatorially complete landscapes, 12 over three positions ($20^3 = 8{,}000$ variants) and three over four ($20^4 = 160{,}000$).

3.2 Metrics

Every method produces the same output, a ranked list from which the top 16 are taken. We set the nomination budget to 16 because it is small enough to reflect one round of low-throughput screening, yet large enough that per-dataset precision and recall are not decided by a single pick. Table 1 defines the four metrics. Together they ask three different questions of the same 16 nominations: how many are exceptional (Precision@16, FoundTop1), how much of the true peak was captured (Recall@16), and whether the hits are spread across enough positions to be recombined into multi-mutants (UniqPos) — because a set of hits concentrated at one position, however strong, cannot be combined.

Table 1: **Metrics.** Higher is better for all four. Each is computed on the 16 variants a method nominates, then averaged across datasets with 95% bootstrap confidence intervals over 1,000 resamples. Precision@16 is measured against percentile bands of the landscape, whereas Recall@16 is measured against the dataset's own 16 best variants.

Metric	Definition	Tasks
Precision@16	Fraction of the 16 nominations in the top 10%, 5% or 1%	All
Recall@16	Fraction of the dataset's true top-16 variants nominated	All
FoundTop1	Whether the single best variant was nominated	All
UniqPos	Distinct positions among the nominations that fall in the true top 10%	Ladder A

3.3 Data sources and curation

All three tasks are assembled from previously published measurements; the per-category composition is given in Appendix Table B.1. Ladder A combines ProteinGym (Notin et al., 2023), a MaveDB-derived collection (Rubin et al., 2025), and the complete PhiX174 proteome mutagenesis of Wei et al. (2026). After deduplication by wild-type sequence, coverage and length filtering, and MMseqs2 (Steinegger and Söding, 2017) clustering at 30% identity within each functional category, 487 assays remain, 96 of which measure function rather than stability. Ladder B's 44 landscapes are mined from SSMuLA (Li et al., 2025), CombinGym (Chen et al., 2026), the GraphFLA catalog (Huang et al., 2025), and combinatorial assays in ProteinGym, MULTI-evolve and MaveDB, keeping only those with a feasible low-N split in every replicate. Lattice's 15 landscapes come from SSMuLA collection (Li et al., 2025). Appendix B details the filtering criteria, the alignments built for the alignment-based baselines, and the Ladder B split sampler.

4 EvolveRank: ranking mutations by pair-wise comparison

EvolveRank is an ML-guided directed evolution method for identifying the very best variants of a target protein based on a small number of wet-lab measurements. Given a pool of N candidate variants, including n variants measured in the wet lab, and a protein language model, it returns a ranking of the pool. The top k not-yet-measured variants are then measured in the wet lab, added to the measured set, and the procedure repeats in the next round. The evaluation is performed after the last round on the top predicted candidates.

The EvolveRank approach (Figure 2) addresses three hard challenges. First, protein activity measurements are highly sensitive to experimental and environmental conditions, rendering regression to values from different measurements unreliable. Rather than directly predicting values, EvolveRank learns from pairs of measured variants to predict rankings, a property that is more robust to measurement variability. Second, the number of measured variants n is typically small (ranging from 16 to 80 in our experiments), which is far too low to fit a high-capacity model and orders of magnitude too low to update a language model by gradient descent. EvolveRank considers every pair of measured variants a training example, which increases the size of the training set from n to $\binom{n}{2}$, and the protein language model serves only as a frozen feature extractor, so EvolveRank fits just a small predictor rather than a fine-tuned network. Third, reasoning about pairs is expensive, since comparing every pair requires $\binom{N}{2}$ comparisons (3.2×10^7 for a three-site library). Instead, EvolveRank uses a Swiss-system tournament with a logarithmic number of passes to compute weak ranking, followed by a Bradley–Terry model to compute strict ranking of all N variants.

Pairwise preference model training. Each variant v_i is represented by a mean-pooled protein-language-model embedding (ESM-2 15B by default),

$$x_i \;=\; \tfrac{1}{L_i} \sum_{l=1}^{L_i} h_l^{(i)} \;\in\; \mathbb{R}^d, \tag{1}$$

where L_i is the number of residues, $h_l^{(i)} \in \mathbb{R}^d$ is the last-layer representation of residue l in the protein language model, and d is the width of that representation; the mean-pooling yields a vector of the same length across all variants, regardless of the number of residues.

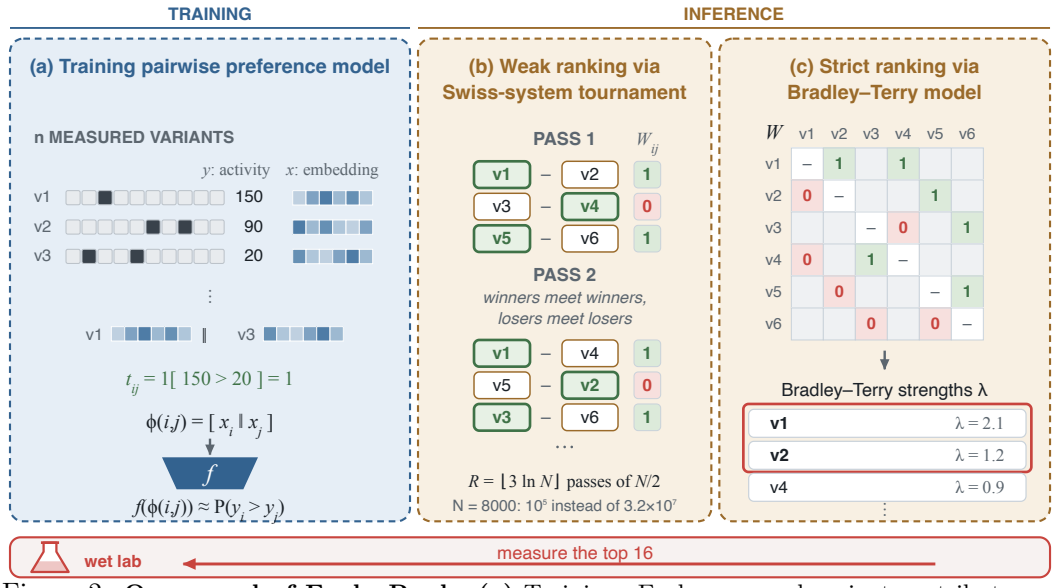

Figure 2: **One round of EvolveRank. (a)** Training. Each measured variant contributes an activity value y and a frozen language-model embedding x; every pair of measured variants becomes one training example, labeled by which member is more active, and a random forest f is fitted to predict that label from the concatenated embeddings $\phi(i, j)$. **(b)** Inference. Rather than compare all $\binom{N}{2}$ pairs, candidates play a Swiss-system tournament: in each pass every candidate is compared with one opponent, and candidates are re-sorted by their running points between passes, so that winners meet winners and losers meet losers. **(c)** The outcomes fill a win matrix W (green where the row variant won, red where it lost, gray where the pair was never compared) from which a Bradley–Terry model recovers a rating λ for every candidate. The top 16 are measured in the wet lab and added to the measured set, closing the loop.

To construct a training set for the model, we form all $\binom{n}{2}$ pairs of the n measured variants, and order the variants in each pair randomly. Each pair is then encoded as a concatenation of the embeddings,

$$\phi(i, j) = [\, x_i \| x_j \,] \in \mathbb{R}^{2d}, \qquad t_{ij} = \mathbf{1}[\, y_i > y_j \,], \tag{2}$$

where $\|$ denotes concatenation, y_i is the measured property value of variant v_i, and t_{ij} is 1 when $y_i > y_j$ and 0 otherwise. The model is trained on these encodings to predict which variant in each pair has the higher measured value. It thus never sees the measured values y, only their pairwise ordering t. Formally, a random forest $f : \mathbb{R}^{2d} \to [0, 1]$ with 100 trees is trained on $\{(\phi(i, j), t_{ij})\}$ so that $f(\phi(i, j))$ estimates the probability $\mathbb{P}(y_i > y_j)$.

Weak ranking via Swiss-system tournament. Comparing all $\binom{N}{2}$ pairs of the variants by the pairwise preference model would be computationally expensive. Instead, we run a logarithmic number of passes of the Swiss-system tournament. In each pass, the N variants are coupled into $\lfloor N/2 \rfloor$ pairs, and the model compares the variants in each pair. The results are used to pair similarly strong variants in subsequent passes, gradually leading to the identification of the best variants. We use $R = \lfloor 3 \ln N \rfloor$ passes, which corresponds to approximately two additional passes each time the number of variants doubles. For $N = 8{,}000$ (a three-site library) this amounts to 26 passes and about 1.0×10^5 comparisons, against 3.2×10^7 for an exhaustive, all-pairs round-robin. The pairing rule, handling of odd pool sizes, two-directional evaluation of each comparison, and exact formula for R are detailed in Appendix C.

Strict ranking via Bradley–Terry model. The Swiss-system tournament provides a weak ranking of the N variants. To obtain a strict ranking, we use the Bradley–Terry model (Bradley and Terry, 1952). It assigns each variant v_i a strength $\lambda_i \geq 0$ such that $\mathbb{P}(i \succ j) = \lambda_i/(\lambda_i + \lambda_j)$, where $\mathbb{P}(i \succ j)$ denotes the probability of variant v_i being ranked above variant v_j by the pairwise preference model. The Swiss-system tournament results in an incomplete pairwise comparison matrix W, where W_{ij} counts the number of comparisons

variant v_i won against variant v_j. The strengths are then obtained as the fixed point of the standard minorization–maximization iteration (Hunter, 2004),

$$\lambda_i \leftarrow \frac{W_i}{\sum_{j \neq i} n_{ij}/(\lambda_i + \lambda_j)}, \qquad \lambda \leftarrow \lambda/\bar{\lambda}, \tag{3}$$

where $W_i = \sum_j W_{ij}$ is the total number of comparisons where variant v_i won, $n_{ij} = W_{ij} + W_{ji}$ is the number of comparisons between variants v_i and v_j, and $\bar{\lambda}$ is the mean strength, by which the strengths λ are rescaled after each update. Top k not-yet-measured variants ($k = 16$ in our experiments) are then measured in wet-lab and added to the measured set ($n \leftarrow n + k$), thereby completing one round of the ML-guided directed evolution. The method is evaluated based on the top 16 variants with the highest strengths λ_i after the last round.

5 Experiments

We benchmark 17 methods across the three tasks of EvolveGym, under the budgets defined in Section 3: zero-shot methods, which use no labels from the target protein; supervised methods, trained on a fixed set of labeled variants; and active-learning methods, which start from 16 randomly drawn variants and acquire a further 64 over four rounds of 16 (Appendix D). Each subsection below reports one task and discusses the key findings.

5.1 Finding beneficial single substitutions (Ladder A)

Actively acquired labels outperform random ones. The three regimes show a clear progression as access to experimental data increases (Table 2). Zero-shot language models achieve up to 19.6% top-5% precision, with SaProt-650M the strongest. Adding 80 labeled variants approximately doubles this, with Kermut strongest within the supervised regime at 41.0%. Active learning yields a further substantial gain under the same labeling budget: EvolveRank with SaProt-1.3B embeddings reaches 77.5% top-5% precision and 37.0% FoundTop1, the strongest configuration overall. The same active > supervised > zero-shot ordering holds across all six property categories (Figure E.1), so the trend is not driven by the stability-dominated composition of the benchmark. EvolveRank leads in five categories, with actively acquired Kermut leading only on the nine-dataset abundance set, and the largest gains occur on activity, binding, expression and organismal fitness – the categories where zero-shot priors are least informative and where measured values are least comparable between experiments. UniqPos follows the same ordering, so the actively acquired nominations are also the most recombinable — which is what Ladder B goes on to test.

Ranking outperforms regression. With the ESM-2 15B backbone fixed, EvolveRank consistently outperforms its regression-based predecessor EVOLVEpro: top-5% precision (71.6% vs. 62.5%, $p = 9.4 \times 10^{-45}$), top-1% precision (21.5% vs. 17.6%) and FoundTop1 (28.5% vs. 23.2%, $p = 1.8 \times 10^{-7}$), under two-sided paired Wilcoxon signed-rank tests with Bonferroni correction. To isolate the effect of pairwise ranking from iterative acquisition, we also evaluated Kermut in the same active-learning loop: active acquisition improves Kermut but does not close the gap to EvolveRank (67.5% vs. 71.6% top-5% precision). The full comparison is reported in Appendix E.1.

5.2 Assembling higher-order variants (Ladder B)

Epistasis determines which method wins. Across all 44 landscapes no learned method clearly outperforms the others: Kermut, additive regression, MULTI-evolve and EvolveRank all reach 39.8–40.6% top-10% Precision@16 with overlapping bootstrap intervals, and the tie extends to Recall@16 and FoundTop1, while EVOLVEpro trails on every metric (37.2%; Table 3). This parity reflects how little a method can learn about higher-order epistasis from first- and second-order variants alone. Stratifying the landscapes by the fraction of reciprocal-sign epistatic motifs, however, reveals a systematic dependence: absolute performance falls sharply from the additive to the epistatic half, with top-10% precision dropping from roughly 52% to 30%, and the ordering of methods reverses. On the more additive half Kermut

Table 2: **Benchmark results on the beneficial single-point mutation identification task (Ladder A)**, across 487 deep mutational scanning datasets, stratified by methodological regime. For each dataset a method nominates a set of 16 candidate variants; reported values are means across datasets, with 95% bootstrap confidence intervals (1,000 resamples) given in brackets. Boldface denotes the best value within each regime. The four EvolveRank rows differ only in the frozen embedding; ESM-2 15B is the default.

Method	N	Precision@16 ↑			Recall@16 ↑	FoundTop1 ↑	UniqPos ↑
		top 10%	top 5%	top 1%			
Zero-shot (no labeled variants)							
VespaG	487	20.2 [19.0, 21.3]	9.3 [8.4, 10.2]	1.7 [1.3, 2.3]	1.9 [1.6, 2.2]	2.1 [0.8, 3.5]	2.7 [2.5, 2.8]
MULTI-evolve ensemble†	481	25.6 [24.3, 26.9]	14.5 [13.4, 15.5]	3.7 [3.2, 4.3]	4.2 [3.8, 4.7]	6.7 [4.6, 8.9]	**4.0 [3.8, 4.2]**
ESM-2 15B	487	27.0 [25.2, 28.7]	15.6 [14.2, 16.9]	4.5 [3.8, 5.4]	5.6 [4.8, 6.4]	6.8 [4.7, 9.0]	2.9 [2.7, 3.1]
LFB (ESM-2 15B)	487	27.9 [26.3, 29.5]	15.9 [14.6, 17.1]	4.3 [3.6, 5.0]	5.2 [4.4, 5.9]	7.4 [5.1, 9.4]	3.2 [3.0, 3.4]
ESM-2 3B	487	28.0 [26.4, 29.7]	16.3 [15.0, 17.6]	4.7 [4.0, 5.5]	5.9 [5.2, 6.7]	7.8 [5.3, 10.3]	3.0 [2.8, 3.2]
LFB (ESM-2 650M)	487	28.5 [26.9, 30.0]	16.6 [15.3, 17.9]	4.6 [3.9, 5.3]	5.6 [4.8, 6.4]	7.6 [5.3, 10.1]	3.2 [3.0, 3.3]
ESM-1v	487	27.9 [26.2, 29.6]	16.7 [15.3, 18.2]	5.2 [4.5, 6.0]	6.3 [5.5, 7.1]	6.4 [4.3, 8.6]	2.8 [2.6, 2.9]
ESMC-6B	487	28.8 [27.0, 30.4]	17.4 [16.0, 18.8]	5.3 [4.5, 6.2]	6.5 [5.7, 7.3]	8.0 [5.7, 10.5]	3.1 [2.9, 3.2]
ESM-2 650M	487	30.0 [28.3, 31.8]	18.0 [16.6, 19.4]	5.3 [4.5, 6.2]	6.7 [5.9, 7.5]	7.8 [5.5, 10.3]	3.0 [2.9, 3.2]
VenusREM	487	30.3 [28.5, 32.1]	18.6 [17.2, 20.1]	**6.2 [5.4, 7.3]**	7.6 [6.6, 8.6]	8.6 [6.4, 11.1]	3.4 [3.2, 3.5]
SaProt-650M	487	**32.2 [30.4, 33.9]**	**19.6 [18.1, 21.1]**	6.2 [5.4, 7.1]	**7.7 [6.8, 8.6]**	**8.8 [6.6, 11.3]**	3.2 [3.0, 3.3]
Supervised few-shot (sees N labeled variants, full-pool ranking)							
Hsu n=16	487	29.9 [29.0, 30.8]	17.1 [16.4, 17.9]	4.2 [3.8, 4.8]	5.2 [4.8, 5.5]	5.2 [4.5, 6.0]	1.7 [1.6, 1.7]
Kermut n=16	487	35.6 [34.3, 36.9]	21.7 [20.6, 22.8]	5.9 [5.3, 6.5]	7.6 [6.9, 8.3]	8.2 [6.8, 9.5]	3.1 [3.0, 3.2]
FSFP n=16	487	36.1 [34.7, 37.5]	22.1 [20.8, 23.2]	6.2 [5.5, 6.9]	8.0 [7.3, 8.7]	9.4 [7.8, 11.1]	3.0 [2.9, 3.1]
Hsu n=80	487	56.0 [55.1, 56.9]	33.7 [32.8, 34.6]	8.5 [8.0, 9.2]	10.9 [10.3, 11.7]	11.3 [10.3, 12.4]	**5.7 [5.6, 5.8]**
FSFP n=80	487	54.4 [52.4, 56.1]	38.4 [36.7, 39.9]	11.6 [10.8, 12.4]	15.4 [14.4, 16.5]	17.0 [15.0, 19.2]	4.3 [4.2, 4.4]
Kermut n=80	487	**56.9 [55.2, 58.6]**	**41.0 [39.3, 42.5]**	**12.7 [11.9, 13.6]**	**16.5 [15.4, 17.6]**	**18.7 [16.8, 20.8]**	4.7 [4.6, 4.9]
Active learning (16 initial + 4 rounds × 16 = 80 variants total)							
EVOLVEpro (ESM-2 15B)	487	85.6 [84.4, 86.8]	62.5 [60.8, 64.3]	17.6 [16.5, 18.5]	23.3 [21.8, 24.8]	23.2 [21.0, 25.7]	5.4 [5.3, 5.5]
Kermut	487	84.7 [83.2, 86.1]	67.5 [65.4, 69.5]	21.0 [19.8, 22.5]	28.1 [26.3, 29.9]	28.2 [25.3, 30.8]	5.3 [5.2, 5.4]
EvolveRank (ESM-2 15B)	487	91.0 [90.0, 91.8]	71.6 [69.8, 73.2]	21.5 [20.3, 22.6]	28.2 [26.5, 29.9]	28.5 [25.7, 31.4]	5.9 [5.7, 6.0]
EvolveRank (ESM-2 650M)	487	90.4 [89.4, 91.3]	71.7 [70.0, 73.5]	21.5 [20.3, 22.7]	28.5 [26.7, 30.2]	28.8 [25.9, 31.8]	6.4 [6.2, 6.5]
EvolveRank (SaProt-650M)	487	92.3 [91.4, 93.1]	77.1 [75.3, 78.8]	25.4 [24.0, 26.8]	33.5 [31.5, 35.5]	34.1 [31.1, 37.4]	6.6 [6.5, 6.8]
EvolveRank (SaProt-1.3B)	487	**92.6 [91.7, 93.5]**	**77.5 [75.6, 79.2]**	**25.6 [24.2, 27.0]**	**34.2 [32.1, 36.2]**	**37.0 [33.8, 40.3]**	**6.7 [6.6, 6.9]**

†MULTI-evolve is evaluated on 481/487 datasets; six are incompatible with its zero-shot ensemble

and additive regression lead, with Kermut recovering 53.0% of the true top-16 variants and finding the global optimum in 70.3% of landscapes. On the more epistatic half EvolveRank is the best individual method on all five metrics, including 33.1% Recall@16 and 52.3% FoundTop1.

An ensemble removes the need to know the regime. Because the regime can be measured in advance but a method should not have to know it, we combine Kermut and EvolveRank with a minimum-rank ensemble. It closely matches the stronger specialist in each half, reaching 52.5% against 52.9% on the additive and 30.5% against 30.8% on the epistatic, while achieving the highest overall top-10% precision (41.5%) and Recall@16 (42.6%). The same regime-specific pattern and ensemble ordering persist when the 44 landscapes are pooled into their 31 protein families (Table E.2 in Appendix).

5.3 Lattice

The full comparison across the 15 landscapes, averaged over their six protein families, is reported in Appendix Table E.4. Key findings are below.

Methods differ only at the very top. On combinatorially complete landscapes every trained method except the two generative baselines (discussed below) recovers most of the top 10% of the landscape (Precision@16 above 82%), so that measure no longer separates them. Methods differ only on the stricter ones: precision within the top 1%, recall of the true top-16 variants, and whether the single best variant is found. This is evidence that metrics for this problem should be defined at the very top of the landscape rather than at its top 10%. Among methods that use no zero-shot prior, the minimum-rank EvolveRank ensembles lead both of the strictest metrics, reaching up to 32.2% Recall@16 and 40.3% FoundTop1, though no single pairing leads on both.

The zero-shot prior matters more than the choice of method. Kermut with the zero-shot prior reaches 93.3%/91.8%/80.9% Precision@16, but the edge comes from the zero-shot signal rather than from the model: removing it drops Kermut to 82.6%. The same signal is worth more still when spent on the first batch — seeding EvolveRank with the zero-shot top 16 instead of a random batch raises top-1% precision from 68% to 93%. We

Table 3: **Constructing higher-order variants (Ladder B).** Each method is trained on single and double mutants and must rank held-out variants of mutational order ≥ 3 (44 landscapes; values are per-dataset means over 10 replicate splits with 95% bootstrap CIs). The landscapes are split at the median into an additive half and an epistatic half, using the fraction of reciprocal-sign epistatic motifs (a GraphFLA ruggedness measure). The final row of each block is a minimum-rank ensemble of Kermut and EvolveRank. Boldface marks the best point estimate per column within each block (within-block differences fall inside overlapping CIs).

Method	N	Precision@16 ↑			Recall@16 ↑	FoundTop1 ↑
		top 10%	top 5%	top 1%		
All landscapes (reciprocal-sign epistasis 0.00–0.34)						
Additive regression	44	40.0 [34.3, 46.5]	25.3 [21.2, 30.1]	8.0 [6.6, 9.6]	41.2 [35.3, 47.3]	58.0 [48.6, 68.4]
MULTI-evolve	44	39.9 [34.1, 46.5]	24.9 [20.8, 29.9]	7.9 [6.4, 9.7]	41.2 [35.3, 47.4]	58.9 [49.3, 68.9]
Kermut	44	40.6 [33.9, 47.4]	25.6 [21.0, 30.6]	8.0 [6.6, 9.6]	41.7 [35.3, 48.2]	**59.2 [50.4, 68.7]**
EVOLVEpro (ESM-2 15B)	44	37.2 [31.8, 43.6]	23.3 [19.6, 27.9]	7.2 [5.8, 8.9]	37.3 [32.1, 42.6]	54.3 [45.7, 63.6]
EvolveRank (ESM-2 15B)	44	39.8 [34.5, 46.2]	24.8 [20.9, 29.6]	7.7 [6.3, 9.5]	40.5 [35.0, 46.5]	56.8 [47.9, 66.6]
Kermut + EvolveRank	44	**41.5 [35.3, 47.9]**	**26.4 [22.0, 31.1]**	**8.2 [6.8, 9.8]**	**42.6 [36.8, 48.9]**	59.0 [50.0, 68.7]
More additive half (reciprocal-sign epistasis 0.00–0.19)						
Additive regression	22	52.4 [45.9, 58.9]	33.0 [28.4, 38.6]	**10.0 [8.5, 11.9]**	51.8 [44.0, 59.7]	69.5 [59.1, 80.0]
MULTI-evolve	22	51.8 [45.6, 58.9]	32.2 [27.6, 38.2]	9.9 [8.2, 12.0]	51.4 [42.6, 59.9]	70.0 [58.2, 80.5]
Kermut	22	**52.9 [45.3, 61.8]**	**33.8 [28.0, 40.9]**	9.9 [8.3, 11.9]	**53.0 [43.9, 61.6]**	**70.3 [59.3, 80.0]**
EVOLVEpro (ESM-2 15B)	22	45.9 [39.0, 53.4]	28.6 [23.8, 34.3]	8.4 [6.6, 10.7]	44.0 [36.8, 51.3]	59.5 [49.5, 68.6]
EvolveRank (ESM-2 15B)	22	48.8 [42.2, 56.5]	30.2 [25.4, 36.2]	9.0 [7.2, 11.3]	47.9 [39.9, 56.2]	61.4 [48.6, 72.7]
Kermut + EvolveRank	22	52.5 [46.2, 60.3]	33.3 [28.6, 39.3]	10.0 [8.5, 11.8]	52.4 [44.0, 60.9]	69.8 [59.4, 80.3]
More epistatic half (reciprocal-sign epistasis 0.20–0.34)						
Additive regression	22	27.6 [20.9, 34.8]	17.6 [12.7, 22.9]	5.9 [4.0, 8.0]	30.5 [23.8, 37.4]	46.4 [32.7, 60.0]
MULTI-evolve	22	28.0 [21.1, 35.1]	17.5 [12.6, 22.8]	5.9 [4.0, 7.9]	31.1 [24.3, 38.3]	47.7 [33.6, 61.8]
Kermut	22	28.2 [21.1, 36.0]	17.5 [12.5, 22.7]	6.1 [4.3, 8.2]	30.4 [23.5, 37.2]	48.2 [35.0, 61.8]
EVOLVEpro (ESM-2 15B)	22	28.6 [22.0, 35.5]	18.0 [13.3, 23.0]	6.0 [4.1, 8.2]	30.5 [24.2, 36.7]	49.1 [35.5, 62.7]
EvolveRank (ESM-2 15B)	22	**30.8 [23.8, 38.4]**	19.3 [14.3, 24.6]	**6.4 [4.5, 8.6]**	**33.1 [26.4, 39.7]**	**52.3 [39.1, 65.5]**
Kermut + EvolveRank	22	30.5 [23.0, 38.6]	**19.5 [14.3, 25.1]**	6.3 [4.3, 8.7]	32.7 [25.6, 39.7]	48.2 [35.0, 61.8]

report this as a caution rather than a recommendation: the standalone zero-shot scorer varies enormously across these six protein families (29.5% top-1% precision, 95% CI [7.3, 54.5]), so the gain may not transfer to a new target.

Preference-optimization baselines do not transfer to this measurement budget. ERA and DPO fall well short here, at 52.0% top-10% precision and near the random floor, but both were developed and evaluated at 384 measured variants in rounds of 96, so the 16-per-round budget of this benchmark lies outside the regime for which they were tuned.

6 Conclusions

We report the following key findings. (1) Active learning substantially improves the identification of high-fitness single-point mutations, with EvolveRank nearly doubling top-5% precision over the strongest supervised model. (2) Under matched backbones and budgets, ranking consistently outperforms regression, with further gains from structure-aware representations. (3) On the multi-point settings no method dominates: in building higher-order variants from measured singles and doubles, a Gaussian process and an additive model lead where mutational effects add up, EvolveRank leads under reciprocal-sign epistasis where mutations interact strongly, and an ensemble of the Gaussian process and EvolveRank matches whichever is better without needing to know which case applies. (4) On fully site-saturated landscapes every method recovers almost the entire top 10%, so methods separate only at the very top: there a zero-shot prior drives precision, and among methods starting from a random batch EvolveRank ensembles find the single best variant most often.

Two key limitations remain. First, the multi-point tasks rest on 44 and 15 landscapes, several of them measuring the same protein; more measured landscapes will be needed to sharpen these conclusions and both tasks state the criteria a new landscape must meet. Second, the large gains from the zero-shot model rest on a signal whose quality varies from one target to the next, so they may not carry over to a new protein.

By showing what is achievable when only a few variants can be measured in each round, this work moves ML-guided directed evolution closer to the many proteins that have so far been engineered by hand.

AI use statement

In this work, we used generative AI tools (Claude and Claude Code, Anthropic). The EvolveGym benchmark and the EvolveRank method are our own design; we implemented them with the help of Claude Code from our specification. We used generative AI tools for writing and editing code and for generating figures and plots from our results. We did not use generative AI tools to generate or modify the underlying data: all measurements come from previously published deep mutational scanning studies, and where AI-written code assembled, cleaned and reformatted the datasets we specified the criteria and verified the resulting composition ourselves. Generative AI was not used to propose or refine our hypotheses, to design the benchmark tasks, metrics or experimental protocol, to interpret the results, or to generate synthetic data, and the work contains no AI-assisted mathematical claims or proofs. Additionally, we used generative AI tools for literature search and for drafting, editing, and reorganizing the manuscript text. We have reviewed all AI-assisted work: we read and tested the code before running it, chose the datasets and protocols and ran the experiments ourselves, and verified the reported results and citations against the underlying data and the original sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility Statement

EvolveGym is built entirely from previously published deep mutational scanning data. The tasks, budgets, splits, and metrics are defined in Section 3, with the dataset curation, filters, and split sampler in Appendix B; the statistical protocol (95% bootstrap confidence intervals and the paired Wilcoxon tests with Bonferroni correction) is described in Section 5 and Appendix E. EvolveRank's implementation and random seeds are given in Appendix C; all baselines and embeddings are described in Appendix D and Table D.1. Code, datasets, the ten fixed replicate splits, embeddings, and predictions are anonymously available at `https://anonymous.4open.science/r/de-benchmark-FD8E/`, with a README that reproduces the reported tables; all resources will be publicly released under an open license upon acceptance.

References

Philip A. Romero and Frances H. Arnold. Exploring protein fitness landscapes by directed evolution. *Nature reviews. Molecular cell biology*, 10(12):866–876, December 2009. ISSN 1471-0072. doi: 10.1038/nrm2805. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC2997618/.

Frances H. Arnold. Innovation by Evolution: Bringing New Chemistry to Life (Nobel Lecture). *Angewandte Chemie International Edition*, 58(41):14420–14426, 2019. ISSN 1521-3773. doi: 10.1002/anie.201907729. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/anie.201907729.

Bruce J Wittmann, Kadina E Johnston, Zachary Wu, and Frances H Arnold. Advances in machine learning for directed evolution. *Current Opinion in Structural Biology*, 69: 11–18, August 2021. ISSN 0959-440X. doi: 10.1016/j.sbi.2021.01.008. URL https://www.sciencedirect.com/science/article/pii/S0959440X21000154.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in neural information processing systems*, volume 34, pages 29287–29303. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f51338d736f95dd42427296047067694-Paper.pdf.

Brian L. Hie, Varun R. Shanker, Duo Xu, Theodora U. J. Bruun, Payton A. Weidenbacher, Shaogeng Tang, Wesley Wu, John E. Pak, and Peter S. Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2): 275–283, February 2024. ISSN 1546-1696. doi: 10.1038/s41587-023-01763-2. URL https://www.nature.com/articles/s41587-023-01763-2.

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, November 2024. doi: 10.1126/science.adr6006. URL https://www.science.org/doi/10.1126/science.adr6006.

Chao Hou, Di Liu, Aziz Zafar, and Yufeng Shen. Understanding language model scaling for protein fitness prediction. *Nature Computational Science*, 6(7):778–788, July 2026. ISSN 2662-8457. doi: 10.1038/s43588-026-01010-z. URL https://www.nature.com/articles/s43588-026-01010-z.

Cade Gordon, Amy X. Lu, and Pieter Abbeel. Protein Language Model Fitness Is a Matter of Preference, October 2024. URL http://biorxiv.org/lookup/doi/10.1101/2024.10.03.616542.

Frances Ding and Jacob Steinhardt. Protein language models are biased by unequal sequence sampling across the tree of life, March 2024. URL http://biorxiv.org/lookup/doi/10.1101/2024.03.07.584001.

Pascal Notin, Aaron W. Kollasch, Daniel Ritter, Lood Van Niekerk, Steffanie Paul, Hansen Spinner, Nathan Rollins, Ada Shaw, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Rose Orenbuch, Yarin Gal, and Debora S. Marks. ProteinGym: Large-Scale Benchmarks for Protein Design and Fitness Prediction, December 2023. URL http://biorxiv.org/lookup/doi/10.1101/2023.12.07.570727.

Francesca-Zhoufan Li, Jason Yang, Kadina E. Johnston, Emre Gürsoy, Yisong Yue, and Frances H. Arnold. Evaluation of machine learning-assisted directed evolution across diverse combinatorial landscapes. *Cell Systems*, 16(9):101387, September 2025. ISSN 24054712. doi: 10.1016/j.cels.2025.101387. URL https://linkinghub.elsevier.com/retrieve/pii/S2405471225002200.

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, January 2025. ISSN 2041-1723. doi: 10.1038/s41467-025-55987-8. URL https://www.nature.com/articles/s41467-025-55987-8.

Sebastian Ibarraran, Shriram Chennakesavalu, Frank Hu, and Grant M. Rotskoff. Efficient, Few-Shot Directed Evolution with Energy Rank Alignment. *Journal of Chemical Information and Modeling*, 66(17):10609–10621, September 2026. ISSN 1549-9596. doi: 10.1021/acs.jcim.6c01333. URL https://doi.org/10.1021/acs.jcim.6c01333.

Andre J. Faure, Jörn M. Schmiedel, Pablo Baeza-Centurion, and Ben Lehner. DiMSum: an error model and pipeline for analyzing deep mutational scanning data and diagnosing common experimental pathologies. *Genome Biology*, 21(1):207, December 2020. ISSN 1474-760X. doi: 10.1186/s13059-020-02091-3. URL https://genomebiology.biomedcentral.com/articles/10.1186/s13059-020-02091-3.

Alexander Kroll, Yvan Rousset, Xiao-Pan Hu, Nina A. Liebrand, and Martin J. Lercher. Turnover number predictions for kinetically uncharacterized enzymes using machine and deep learning. *Nature Communications*, 14(1):4139, July 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-39840-4. URL https://www.nature.com/articles/s41467-023-39840-4.

C. J. Markin, D. A. Mokhtari, F. Sunden, M. J. Appel, E. Akiva, S. A. Longwell, C. Sabatti, D. Herschlag, and P. M. Fordyce. Revealing enzyme functional architecture via high-throughput microfluidic enzyme kinetics. *Science*, 373(6553):eabf8761, July 2021. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.abf8761. URL https://www.science.org/doi/10.1126/science.abf8761.

Christian Dallago, Jody Mou, Kadina E. Johnston, Bruce J. Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K. Yang. FLIP: Benchmark tasks in fitness landscape inference for proteins, November 2021. URL http://biorxiv.org/lookup/doi/10.1101/2021.11.09.467890.

Kieran Didi, Sarah Alamdari, Alex X. Lu, Bruce Wittmann, Kadina E. Johnston, Ava P. Amini, Ali Madani, Maya Czeneszew, Christian Dallago, and Kevin K. Yang. FLIP2: Expanding Protein Fitness Landscape Benchmarks for Real-World Machine Learning Applications, February 2026. URL http://biorxiv.org/lookup/doi/10.64898/2026.02.23.707496.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating Protein Transfer Learning with TAPE, June 2019. URL http://arxiv.org/abs/1906.08230. arXiv:1906.08230 [cs.LG].

Minghao Xu, Zuobai Zhang, Jiarui Lu, Zhaocheng Zhu, Yangtian Zhang, Chang Ma, Runcheng Liu, and Jian Tang. PEER: A Comprehensive and Multi-Task Benchmark for Protein Sequence Understanding, September 2022. URL http://arxiv.org/abs/2206.02096.

Yongcan Chen, Lihao Fu, Xuchao Lu, Wenzhuo Li, Yuan Gao, Yibo Wang, Zhicheng Ruan, and Tong Si. CombinGym: a benchmark platform for machine learning-assisted design of combinatorial protein variants, March 2026. URL http://biorxiv.org/lookup/doi/10.64898/2026.03.24.714074.

Sam Sinai, Richard Wang, Alexander Whatley, Stewart Slocum, Elina Locane, and Eric D. Kelsic. AdaLead: A simple and robust adaptive greedy search algorithm for sequence design, October 2020. URL http://arxiv.org/abs/2010.02141.

Yinglu Cui, Yanchun Chen, Xinyue Liu, Saijun Dong, Yue Tian, Yuxin Qiao, Ruchira Mitra, Jing Han, Chunli Li, Xu Han, Weidong Liu, Quan Chen, Wangqing Wei, Xin Wang, Wenbin Du, Shuangyan Tang, Hua Xiang, Haiyan Liu, Yong Liang, Kendall N. Houk, and Bian Wu. Computational Redesign of a PETase for Plastic Biodegradation under Ambient Condition by the GRAPE Strategy. *ACS Catalysis*, 11(3):1340–1350,

February 2021. ISSN 2155-5435, 2155-5435. doi: 10.1021/acscatal.0c05126. URL https://pubs.acs.org/doi/10.1021/acscatal.0c05126.

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, May 2026. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.aea1820. URL https://www.science.org/doi/10.1126/science.aea1820.

Gian Marco Visani, Aayush Verma, and William S. DeWitt. Additive baselines furnish no evidence for epistasis learning by MULTI-evolve, April 2026. URL http://biorxiv.org/lookup/doi/10.64898/2026.04.23.719915.

Manfred T Reetz and José Daniel Carballeira. Iterative saturation mutagenesis (ISM) for rapid directed evolution of functional enzymes. *Nature Protocols*, 2(4):891–903, April 2007. ISSN 1754-2189, 1750-2799. doi: 10.1038/nprot.2007.72. URL https://www.nature.com/articles/nprot.2007.72.

Zachary Wu, S. B. Jennifer Kan, Russell D. Lewis, Bruce J. Wittmann, and Frances H. Arnold. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, 116(18):8852–8858, April 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1901979116. URL https://pnas.org/doi/full/10.1073/pnas.1901979116.

Kerr Ding, Michael Chin, Yunlong Zhao, Wei Huang, Binh Khanh Mai, Huanan Wang, Peng Liu, Yang Yang, and Yunan Luo. Machine learning-guided co-optimization of fitness and diversity facilitates combinatorial library design in enzyme engineering. *Nature Communications*, 15(1):6392, July 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-50698-y. URL https://www.nature.com/articles/s41467-024-50698-y.

Ziyi Zhou, Liang Zhang, Yuanxi Yu, Banghao Wu, Mingchen Li, Liang Hong, and Pan Tan. Enhancing efficiency of protein language models with minimal wet-lab data through few-shot learning. *Nature Communications*, 15(1):5566, July 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-49798-6. URL https://www.nature.com/articles/s41467-024-49798-6.

Minji Lee, Kyungmin Lee, and Jinwoo Shin. Fine-tuning protein Language Models by ranking protein fitness. In *NeurIPS 2023 generative AI and biology (GenBio) workshop*, 2023. URL https://openreview.net/forum?id=DUjUJCqqA7.

Shriram Chennakesavalu, Frank Hu, Sebastian Ibarraran, and Grant M. Rotskoff. Aligning Transformers with Continuous Feedback via Energy Rank Alignment, October 2025. URL http://arxiv.org/abs/2405.12961.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems 36*, pages 53728–53741, New Orleans, Louisiana, USA, 2023. Neural Information Processing Systems Foundation, Inc. (NeurIPS). ISBN 978-1-7138-9911-2. doi: 10.52202/075280-2338. URL http://www.proceedings.com/075280-2338.html.

Ralph Allan Bradley and Milton E. Terry. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324, December 1952. ISSN 00063444. doi: 10.2307/2334029. URL https://www.jstor.org/stable/2334029?origin=crossref.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank Centrality: Ranking from Pair-wise Comparisons, November 2015. URL http://arxiv.org/abs/1209.1688.

Soheil Mohajer, Changho Suh, and Adel Elmahdy. Active learning for top-K rank aggregation from noisy comparisons. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th international conference on machine learning*, volume 70 of *Proceedings of machine learning research*, pages 2488–2497. PMLR, August 2017. URL https://proceedings.mlr.press/v70/mohajer17a.html.

Mingyu Huang, Shasha Zhou, and Ke Li. Augmenting Biological Fitness Prediction Benchmarks with Landscapes Features from GraphFLA, October 2025. URL http://arxiv.org/abs/2510.24826.

Nicholas C Wu, Lei Dai, C Anders Olson, James O Lloyd-Smith, and Ren Sun. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife*, 5:e16965, July 2016. ISSN 2050-084X. doi: 10.7554/eLife.16965. URL https://doi.org/10.7554/eLife.16965.

Alan F. Rubin, Jeremy Stone, Aisha Haley Bianchi, Benjamin J. Capodanno, Estelle Y. Da, Mafalda Dias, Daniel Esposito, Jonathan Frazer, Yunfan Fu, Sally B. Grindstaff, Matthew R. Harrington, Iris Li, Abbye E. McEwen, Joseph K. Min, Nick Moore, Olivia G. Moscatelli, Jesslyn Ong, Polina V. Polunina, Joshua E. Rollins, Nathan J. Rollins, Ashley E. Snyder, Amy Tam, Matthew J. Wakefield, Shenyi Sunny Ye, Lea M. Starita, Vanessa L. Bryant, Debora S. Marks, and Douglas M. Fowler. MaveDB 2024: a curated community database with over seven million variant effects from multiplexed functional assays. *Genome Biology*, 26(1):13, January 2025. ISSN 1474-760X. doi: 10.1186/s13059-025-03476-y. URL https://doi.org/10.1186/s13059-025-03476-y.

Huijin Wei, Xianghua Li, and Ben Lehner. Complete Mutagenesis of the Genome and Proteome of X174, July 2026. URL http://biorxiv.org/lookup/doi/10.64898/2026.07.25.740675.

Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, November 2017. ISSN 1546-1696. doi: 10.1038/nbt.3988. URL https://www.nature.com/articles/nbt.3988.

David R. Hunter. MM algorithms for generalized Bradley-Terry models. *The Annals of Statistics*, 32(1), February 2004. ISSN 0090-5364. doi: 10.1214/aos/1079120141. URL https://projecteuclid.org/journals/annals-of-statistics/volume-32/issue-1/MM-algorithms-for-generalized-Bradley-Terry-models/10.1214/aos/1079120141.full.

L Steven Johnson, Sean R Eddy, and Elon Portugaly. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics*, 11(1):431, December 2010. ISSN 1471-2105. doi: 10.1186/1471-2105-11-431. URL https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-431.

Baris E. Suzek, Yuqi Wang, Hongzhan Huang, Peter B. McGarvey, Cathy H. Wu, and the UniProt Consortium. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, March 2015. ISSN 1367-4811, 1367-4803. doi: 10.1093/bioinformatics/btu739. URL https://academic.oup.com/bioinformatics/article/31/6/926/214968.

Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K. Min, Kelly Brock, Yarin Gal, and Debora S. Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, November 2021. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-021-04043-8. URL https://www.nature.com/articles/s41586-021-04043-8.

Andrei Papkou, Lucia Garcia-Pastor, José Antonio Escudero, and Andreas Wagner. A rugged yet easily navigable fitness landscape. *Science*, 382(6673):eadh3860, November 2023. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.adh3860. URL https://www.science.org/doi/10.1126/science.adh3860.

Thuy-Lan V Lite, Robert A Grant, Isabel Nocedal, Megan L Littlehale, Monica S Guo, and Michael T Laub. Uncovering the basis of protein-protein interaction specificity with a combinatorially complete library. *eLife*, 9:e60924, October 2020. ISSN 2050-084X. doi: 10.7554/eLife.60924. URL https://elifesciences.org/articles/60924.

Kadina E. Johnston, Patrick J. Almhjell, Ella J. Watkins-Dulaney, Grace Liu, Nicholas J. Porter, Jason Yang, and Frances H. Arnold. A combinatorially complete epistatic fitness

landscape in an enzyme active site. *Proceedings of the National Academy of Sciences*, 121 (32):e2400439121, August 2024. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2400439121. URL https://pnas.org/doi/10.1073/pnas.2400439121.

Boqiang Tu, Vikram Sundar, and Kevin M. Esvelt. An ultra-high-throughput method for measuring biomolecular activities, March 2022. URL http://biorxiv.org/lookup/doi/10.1101/2022.03.09.483646.

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. URL https://www.science.org/doi/abs/10.1126/science.ade2574.

Salvatore Candido, Thomas Hayes, Alexander Derry, Roshan Rao, Zeming Lin, Robert Verkuil, Bryan Z. Wu, Jin Sub Lee, Elise S. Bruguera, Jehan A. Keval, Mykhailo Kopylov, John E. Pak, Wesley Wu, Neil Thomas, Samson Mataraso, Alvin Hsu, Ashton C. Trotman-Grant, Kilian Fatras, Allan dos Santos Costa, Rohil Badkundri, Halil Akn, Deniz Oktay, Jonathan Deaton, Elizabeth Montabana, Hrishita Sitwala, Yue Yu, Marius Wiggert, Dylan Alexander Carlin, Anthony W. Goering, Tomasz Blazejewski, McCullen Sandora, Michael Hla, Tina Z. Jia, Leon H. Kloker, Nicholas J. Sofroniew, Masatoshi Uehara, Jassi Pannu, Sharrol Bachas, Daniel S. Liu, Tom Sercu, and Alexander Rives. Language Modeling Materializes a World Model of Protein Biology, June 2026. URL https://www.biorxiv.org/content/10.64898/2026.06.03.729735v1. ISSN: 2692-8205 Pages: 2026.06.03.729735 Section: New Results.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. SaProt: Protein language modeling with structure-aware vocabulary. In *The twelfth international conference on learning representations*, 2024. URL https://openreview.net/forum?id=6MRm3G4NiU.

Yang Tan, Ruilin Wang, Banghao Wu, Liang Hong, and Bingxin Zhou. From high-throughput evaluation to wet-lab studies: advancing mutation effect prediction with a retrieval-enhanced model. *Bioinformatics*, 41(Supplement_1):i401–i409, July 2025. ISSN 1367-4803, 1367-4811. doi: 10.1093/bioinformatics/btaf189. URL https://academic.oup.com/bioinformatics/article/41/Supplement_1/i401/8199374.

Charles W. J. Pugh, Paulina G. Nuñez-Valencia, Mafalda Dias, and Jonathan Frazer. From Likelihood to Fitness: Improving Variant Effect Prediction in Protein and Genome Language Models, May 2025. URL http://biorxiv.org/lookup/doi/10.1101/2025.05.20.655154.

Céline Marquet, Julius Schlensok, Marina Abakarova, Burkhard Rost, and Elodie Laine. Expert-guided protein language models enable accurate and blazingly fast fitness prediction. *Bioinformatics*, 40(11):btae621, November 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae621. URL https://doi.org/10.1093/bioinformatics/btae621.

Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 40(7): 1114–1122, July 2022. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-021-01146-5. URL https://www.nature.com/articles/s41587-021-01146-5.

Peter Mørch Groth, Mads Herbert Kerrn, Lars Olsen, Jesper Salomon, and Wouter Boomsma. Kermut: Composite kernel regression for protein variant effects, October 2024. URL http://arxiv.org/abs/2407.00002.

Appendix

A Comparison to existing benchmarks **16**

B Benchmark construction **17**

B.1 Ladder A: assay selection 17

B.2 Multiple sequence alignments 18

B.3 Ladder B: split construction 18

B.4 Lattice: landscape roster 18

C EvolveRank implementation details **19**

D Baseline methods **19**

E Extended results **20**

E.1 Ladder A 20

E.2 Ladder B 20

E.3 Lattice 24

A Comparison to existing benchmarks

To compare methods for ML-guided directed evolution, a benchmark has to satisfy three requirements at once. Its data must be wet-lab measurements rather than *in-silico* oracles, since a method that exploits the quirks of an *in-silico* landscape tells us nothing about a real campaign. Its regime coverage must include both single substitutions under broad positional coverage, where beneficial mutations are discovered, and multi-point variants, where they are combined; no single assay provides both, so a benchmark has to draw on several. And its metric must reward recovering the few best variants within a fixed screening budget, because that is the only quantity a campaign consumes. Table A.1 assesses existing benchmarks against these requirements.

The established fitness-prediction suites meet the first requirement and part of the second, but not the third. ProteinGym (Notin et al., 2023) aggregates 217 measured substitution DMS assays, and FLIP, FLIP2, TAPE and PEER (Dallago et al., 2021; Didi et al., 2026; Rao et al., 2019; Xu et al., 2022) likewise use measured data, but each scores a model over an entire assay by rank correlation or a related global statistic. Even ProteinGym's Top-K recall, the most selective of its metrics, is evaluated at the top decile, hundreds to thousands of variants; and its supervised track trains on roughly 80% of each assay under five-fold cross-validation, an order of magnitude more labels than a campaign has. FLEXS (Sinai et al., 2020) is the one widely used closed-loop framework and does evaluate along budget and round axes, but almost all of its landscapes are *in-silico* oracles, so it does not meet the first requirement instead. The combinatorial collections sit on the other side of the trade-off: CombinGym (Chen et al., 2026) curates 14 measured multi-mutant landscapes across 9 proteins, and Li et al. (2025) evaluate ML-guided directed evolution across 16 combinatorial landscapes under campaign-style, budget-limited metrics, but both are confined to the combinatorial regime and to a small number of target proteins, predominantly TrpB in the latter case.

No existing benchmark therefore satisfies all three requirements simultaneously. EvolveGym does not contribute new measurements; every landscape it uses has been published before. Its contribution is to unify the single-substitution and combinatorial regimes under one

budget-limited top-variant protocol, and to do so at a scale — 487 assays alongside 44 multi-mutant and 15 combinatorially complete landscapes — at which a conclusion is not an artifact of which handful of proteins a paper happened to pick.

Benchmark	# sets	Prot.	Measured	Regime			Top-variant
				Single	Multi	Site-sat.	
ProteinGym	217	187	✓	✓	(✓)	×	(✓)
FLIP / FLIP2	12 + 16	9	✓	×	✓	✓	×
TAPE	1	1	✓	(✓)	(✓)	×	×
PEER	2	2	✓	✓	(✓)	×	×
FLEXS	3	3	×	×	✓	×	✓
CombinGym	14	9	✓	×	✓	×	×
Li et al. (2025)	16	7	✓	×	✓	✓	✓
EvolveGym (ours)	487 + 44 + 15	497	✓	✓	✓	✓	✓

Table A.1: **Existing benchmarks for protein fitness prediction and ML-guided directed evolution.** Rows, in order: ProteinGym (Notin et al., 2023), FLIP and FLIP2 (Dallago et al., 2021; Didi et al., 2026), TAPE (Rao et al., 2019), PEER (Xu et al., 2022), FLEXS (Sinai et al., 2020), CombinGym (Chen et al., 2026), and Li et al. (2025) whose landscapes are predominantly TrpB libraries and are released as the SSMuLA collection that supplies 15 of our Lattice and 15 of our Ladder B landscapes. ✓ = supported, (✓) = partially supported, × = not supported; *# sets* counts assays or landscapes as each benchmark defines them; for EvolveGym, 487 single-substitution assays, 44 multi-mutant landscapes and 15 combinatorially complete landscapes. *Prot.* counts distinct target proteins, which is what limits how far a result generalizes. *Measured*: scores are wet-lab measurements rather than *in-silico* oracles. *Regime*: single substitutions under broad positional coverage; higher-order multi-mutants; combinatorially complete site-saturated libraries. *Top-variant*: methods are scored on recovery of the few best variants under a fixed screening budget rather than on ordering the whole landscape. ProteinGym's Top-K recall is top-focused but set at the top decile, and its supervised track trains on ∼80% of each assay under five-fold cross-validation, far more labels than a campaign has. EvolveGym is the first to combine measured data, all three regimes, and a budget-limited top-variant metric.

B Benchmark construction

Table B.1 summarizes the composition of the three tasks. Ladder A is broken down by the property each assay measures, because that composition is inherited from the public deep mutational scanning record rather than chosen: stability assays make up 80% of it, and the 96 assays measuring activity, binding, expression, abundance or organismal fitness are the ones where evolutionary priors are least informative. The subsections below give the filters, alignments and split construction behind each task.

B.1 Ladder A: assay selection

We integrated deep mutational scanning substitution assays from three public sources: ProteinGym (Notin et al., 2023), the MaveDB-derived collection (Rubin et al., 2025), and the complete PhiX174 genome and proteome mutagenesis dataset of Wei et al. (2026). Datasets were deduplicated by target (wild-type) sequence, retaining the version with higher single-mutant coverage at overlapping sites.

We then filtered on single-mutant coverage. Stability datasets were required to have at least 90% coverage, while the functional categories (Activity, Binding, Abundance, Expression and Organismal Fitness) required at least 70%; the looser threshold retains functional assays, which are scarcer and less completely measured than stability assays. Sequences shorter than 30 amino acids were excluded.

To avoid inflated performance estimates arising from homologous datasets, we clustered with MMseqs2 (Steinegger and Söding, 2017) at 30% sequence identity and 80% coverage, retaining for each cluster the dataset with the highest single-mutant coverage. Clustering was performed independently within each functional category, so that datasets measuring different properties of the same protein are preserved. All ten PhiX174 proteome assays

Table B.1: **Composition of EvolveGym.** Ladder A is broken down by the property each assay measures; Ladder B and Lattice are reported per task. *Variants* and *Length* are medians per dataset. Stability assays dominate Ladder A (80%), which reflects the composition of the public deep mutational scanning record rather than a design choice; we therefore report every metric stratified by property (Figure E.1) so that the functional categories are not hidden inside the average.

Task / property	# sets	Variants	Length	Assay measures
Ladder A: single substitutions				
Stability	391	1,059	56	Thermostability or proteolysis
Organismal fitness	50	5,576	317	Growth- or enrichment-based selection
Activity	17	6,227	402	Specific molecular activity
Binding	10	2,882	157	Binding affinity or interaction
Expression	10	5,220	298	Surface expression or insertion
Abundance	9	5,843	313	Intracellular protein abundance
Subtotal	487	1,158	62	
Ladder B: higher-order multi-mutants				
All landscapes	44	8,692	242	Enzyme activity, binding, fluorescence...
Lattice: site-saturated libraries				
Three positions	12	7,886	390	Enzyme activity and binding
Four positions	3	159,128	236	Enzyme activity and binding

passed the coverage and length filters ($\geq$ 98% coverage) and formed singleton clusters, confirming that they are non-redundant with the existing datasets. The final collection comprises 487 datasets across six functional categories (Table B.1).

B.2 Multiple sequence alignments

Alignment-based predictors require evolutionary context, which we assembled as follows. For datasets originating from ProteinGym we used the official ProteinGym alignments (Notin et al., 2023), generated with jackhmmer against UniRef100 using a per-dataset optimized inclusion bit-score. For the MaveDB-derived datasets and the PhiX174 assays, for which no curated alignment is available, we generated alignments de novo with jackhmmer (Johnson et al., 2010) (5 iterations, fixed inclusion threshold of 0.5 bits per residue) against UniRef100 (Suzek et al., 2015), downloaded April 2026, followed by the EVE post-processing filters (Frazer et al., 2021): sequences with more than 50% gaps and columns with less than 70% residue occupancy were removed.

B.3 Ladder B: split construction

The 44 landscapes are mined from SSMuLA (Li et al., 2025) (15), CombinGym (Chen et al., 2026) (7), the GraphFLA catalog (Huang et al., 2025) (3), and combinatorial assays curated in ProteinGym, MULTI-evolve and MaveDB (19), deduplicated by target sequence. A dataset was kept only if a low-N split was feasible, meaning that at least 100 variants of order ≥ 3 whose constituent single mutations are all measured were reachable in every replicate.

We generate 10 replicate splits per dataset. Each split is defined by a pool of single mutations, built by a connected sampler that shuffles the order-≥ 3 variants and adds their constituent single mutations one variant at a time, until the pool covers at least 100 order-≥ 3 variants; a variant is *covered* when all of its constituent single mutations are in the pool. The training set is then every single mutant in the pool together with every double mutant whose two constituents are both in the pool, and the test set is every order-≥ 3 variant that the pool covers.

B.4 Lattice: landscape roster

The 15 site-saturated landscapes are taken from SSMuLA (Li et al., 2025), in which all 20 amino acids are measured at each of K sites. Twelve are three-site landscapes ($20^3 = 8{,}000$ variants): DHFR (Papkou et al., 2023), ParD2 and ParD3 (Lite et al., 2020), and TrpB3A–

TrpB3I (Johnston et al., 2024). Three are four-site landscapes ($20^4 = 160{,}000$ variants): GB1 (Wu et al., 2016), TEV (Tu et al., 2022) and TrpB4 (Johnston et al., 2024). Measured coverage is at least 0.93 in every case, and each landscape was verified against its primary deposit.

C EvolveRank implementation details

Number of passes. The number of Swiss passes is

$$R = \max\big(1,\ \min(N-1,\ \lfloor 3\ln N \rfloor)\big), \tag{C.1}$$

where N is the size of the candidate pool and the two guards ensure at least one pass and never more than the $N-1$ passes a full round-robin would need. Both bind only for pools of fewer than about ten candidates and are therefore inactive in every experiment reported here, which is why the main text quotes the unguarded $R = \lfloor 3\ln N \rfloor$.

Pairing within a pass. In each pass, candidates are sorted by their running points, with one point awarded per win and a free point awarded for a bye — the candidate left unpaired when N is odd — and each candidate is then paired with the nearest opponent in that order that it has not yet played. Sorting before pairing is what concentrates comparisons near the top of the order: after the first pass, candidates that have won tend to be matched against other winners, so the tournament spends its budget separating strong candidates from one another rather than re-establishing that strong candidates beat weak ones. Byes are excluded from the win matrix, so a candidate is never credited with a win it did not play for.

Comparison rule and tie-breaking. A random forest classifier is not antisymmetric. Trained on pairs presented in a random orientation, it does not guarantee $f(\phi(i,j)) = 1 - f(\phi(j,i))$, so the winner of a comparison could otherwise depend on which variant happens to be written first. We therefore evaluate every comparison in both directions and declare that i beats j when $f(\phi(i,j)) > f(\phi(j,i))$. Equivalently, i beats j when the averaged preference $\frac{1}{2}\big(f(\phi(i,j)) + 1 - f(\phi(j,i))\big)$ exceeds $\frac{1}{2}$, so the rule is the natural symmetrization of the classifier rather than an arbitrary choice of orientation. Exact ties, which are negligible for real-valued scores, are awarded to j. Evaluating both directions doubles the number of classifier calls per comparison, about 2.1×10^5 rather than 1.0×10^5 for a pool of $N = 8000$, but leaves the $\mathcal{O}(N \ln N)$ scaling of the tournament unchanged.

Fitting. The pairwise classifier is a scikit-learn random forest of 100 trees, using the Gini criterion, bootstrap sampling and `max_features=None` rather than the library default, so that every feature is considered at each split; all remaining settings are at library defaults. The Bradley–Terry fixed point is iterated until the L_2 norm of the change in λ between successive iterations falls below 10^{-8}, or for at most 100 iterations, whichever comes first, with a term $\varepsilon = 10^{-12}$ added to each $(\lambda_i + \lambda_j)$ to avoid division by zero. A candidate that won no comparison receives $\lambda_i = 0$; such candidates tie at the bottom of the ranking and are never selected, since only the top B are measured. Training-pair orientations are reseeded per replicate, the ten replicate splits are fixed and stored, and both the classifier and the tournament ranker are seeded by the replicate index. The same procedure is used for all three tasks, and only the candidate pool differs: every measured single mutant for Ladder A, the held-out variants of order ≥ 3 for Ladder B, and the full combinatorial library for Lattice.

D Baseline methods

Each task draws on its own roster, chosen to span the main methodological families while including a state-of-the-art representative of each.

Single substitutions (Ladder A). The *zero-shot* category comprises PLM likelihood scorers spanning a range of scales and architectures (ESM-1v (Meier et al., 2021); ESM-2 at 650M, 3B and 15B (Lin et al., 2023); ESMC-6B (Candido et al., 2026)), a structure-aware model (SaProt-650M (Su et al., 2024)), a retrieval-augmented model (VenusREM (Tan

et al., 2025)), MSA-aware scoring (LFB (Pugh et al., 2025)), a distilled supervised proxy (VespaG (Marquet et al., 2024)), and a zero-shot ensemble (MULTI-evolve (Tran et al., 2026)). The *supervised few-shot* category trains on a fixed set of labeled variants (n=16 and n=80) and includes a low-capacity ridge baseline (Hsu OneHot Ridge (Hsu et al., 2022)), a Gaussian-process model with a structure-aware kernel (Kermut (Groth et al., 2024)), and a parameter-efficient few-shot fine-tuning method (FSFP (Zhou et al., 2024)). The *active learning* category starts from 16 randomly drawn variants and acquires a further 64 over four rounds of 16 (80 total), and comprises EVOLVEpro (Jiang et al., 2024) and our ranking variant EvolveRank. Full descriptions and per-method rationale are given in Table D.1

Higher-order construction (Ladder B). Every method is trained on the single and double mutants of a landscape and must rank its held-out variants of order ≥ 3. We compare an *additive regression* baseline that predicts each variant from independent per-site effects fitted on the training mutants, the multi-mutant engineering framework MULTI-evolve, the structure-aware Gaussian-process model Kermut, EVOLVEpro, and our ranking variant EvolveRank (all on ESM-2 15B embeddings), together with a minimum-rank ensemble of Kermut and EvolveRank.

Site-saturated libraries (Lattice). Every method operates in closed loop, starting from 16 randomly drawn variants and acquiring a further 64 over four rounds of 16. Alongside a *random* acquisition baseline and a static *zero-shot* scorer (ESM-2 650M), we evaluate two reward- and preference-based generative fine-tuning methods from the directed-evolution literature (ERA and DPO (Ibarraran et al., 2026; Rafailov et al., 2023)), a Bayesian-optimization directed-evolution method (ALDE, GP-UCB (Yang et al., 2025)), Kermut, EVOLVEpro, and our ranking variant EvolveRank across several embeddings (ESM-2 650M and 15B, SaProt-650M, ESMC-6B). To probe the effect of the zero-shot signal, we additionally run Kermut with the ESM-2 650M zero-shot scores as its GP mean and EvolveRank warm-started from the zero-shot top-16 batch. Finally, we report minimum-rank ensembles that merge two of these rankings.

E Extended results

E.1 Ladder A

Figure E.1 breaks the Ladder A results down by the property each assay measures, showing that the active > supervised > zero-shot ordering holds within each of the six categories rather than only in the aggregate.

We ablate the two levers of the active-learning loop—which variants are acquired (random vs. active selection) and which surrogate ranks the library (EvolveRank's pairwise ranker vs. Kermut's Gaussian process)—under a fixed budget of 80 measurements, with every method re-ranking the entire library at the final round (Table E.1). Pairwise comparisons use two-sided Wilcoxon signed-rank tests across the 487 datasets (per-dataset scores averaged over 10 seeds), with Bonferroni correction over the reported comparisons. Acquisition is the dominant factor: for the GP surrogate, random selection is markedly worse than informative picks on every metric ($p < 10^{-17}$). The surrogate, by contrast, matters only for broad enrichment: EvolveRank surpasses the GP on Precision@16 at the 10% and 5% thresholds (+6.3 and +4.0 points, $p < 10^{-8}$). Notably, this advantage is confined to broad enrichment—on the strictest metrics (Precision@16 at 1%, Recall@16, and FoundTop1) the two differ by less than half a point and are statistically indistinguishable ($p > 0.1$), consistent with our broader finding that no single method dominates across regimes.

E.2 Ladder B

Table E.2 repeats the Ladder B comparison with each protein family contributing once rather than each landscape, and Table E.3 varies the EvolveRank backbone; neither changes the additive/epistatic crossover reported in Section 5.

Table D.1: Methods evaluated in the Ladder A experimental setting, grouped by methodology regime, with rationale for inclusion.

Method	Category	Rationale for inclusion
Zero-shot (no labeled variants)		
ESM-1v (Meier et al., 2021)	Zero-shot PLM	Early widely-used PLM baseline; reference point for newer ESM-2 models.
ESM-2 (650M, 3B, 15B) (Lin et al., 2023)	Zero-shot PLM	The ESM-2 family evaluated at three scales, tracing the scale-vs-performance curve; the 15B model is also the frozen embedding backbone used by EVOLVEpro and EvolveRank.
ESMC-6B (Candido et al., 2026)	Zero-shot PLM	Latest-generation ESM Cambrian model at its largest open (6B) scale; tests whether the newer architecture and training recipe improve zero-shot fitness prediction over the ESM-2 family.
SaProt-650M (Su et al., 2024)	Structure-aware PLM	Sequence PLM augmented with Foldseek structural tokens; strongest zero-shot scorer in our Ladder A benchmark and included as the SOTA structure-aware representative. Also used as a frozen embedding backbone for EvolveRank.
VespaG (Marquet et al., 2024)	Supervised proxy model	Trained on millions of variants annotated using GEMME-derived scores; tests whether a small distilled model can rival expensive direct PLM scoring.
VenusREM (Tan et al., 2025)	Retrieval-augmented PLM	Among the top-performing retrieval-enhanced methods on ProteinGym benchmarks; included as SOTA representative of this family.
Likelihood Fitness Bridging (LFB) (Pugh et al., 2025)	MSA-aware PLM	Combines large PLMs with MSA-derived sequence context (applied here at two scales, ESM-2 650M and ESM-2 15B); tests the value of explicit evolutionary information on top of PLM.
MULTI-evolve (Tran et al., 2026)	Zero-shot ensemble	Combines several zero-shot PLM and structural signals; originally designed for combinatorial mutant proposals, included to assess applicability on single-mutant DMS data.
Passive supervised (n=16, n=80 labels)		
Hsu OneHot Ridge (Hsu et al., 2022)	Passive supervised baseline	Classical low-capacity baseline from Hsu et al.; isolates the marginal benefit of active learning at matched budgets.
Kermut (Groth et al., 2024)	Supervised GP	Gaussian-process regressor with a structure-aware mutation kernel; current SOTA on the ProteinGym supervised benchmark. Included as a strong supervised baseline at matched n=16/n=80 budgets.
Few-Shot Fitness Prediction (FSFP) (Zhou et al., 2024)	Few-shot fine-tuning	Recent few-shot fitness predictor with parameter-efficient fine-tuning (LoRA + meta-learning); tested at the same budgets as Hsu and EVOLVEpro for direct comparison.
Active learning (5 rounds, n=80 labels in total)		
EVOLVEpro (Jiang et al., 2024)	Active-learning method	Closest prior active-learning method; 5 rounds of acquisition over ESM-2 15B embeddings with Random Forest regression; quantifies lift of active learning over passive supervised baselines.
EvolveRank (Ours)	Active-learning method	Proposed method: ranks candidates by pairwise comparison rather than fitness regression, using a random forest preference classifier over frozen PLM embeddings with Bradley–Terry aggregation.

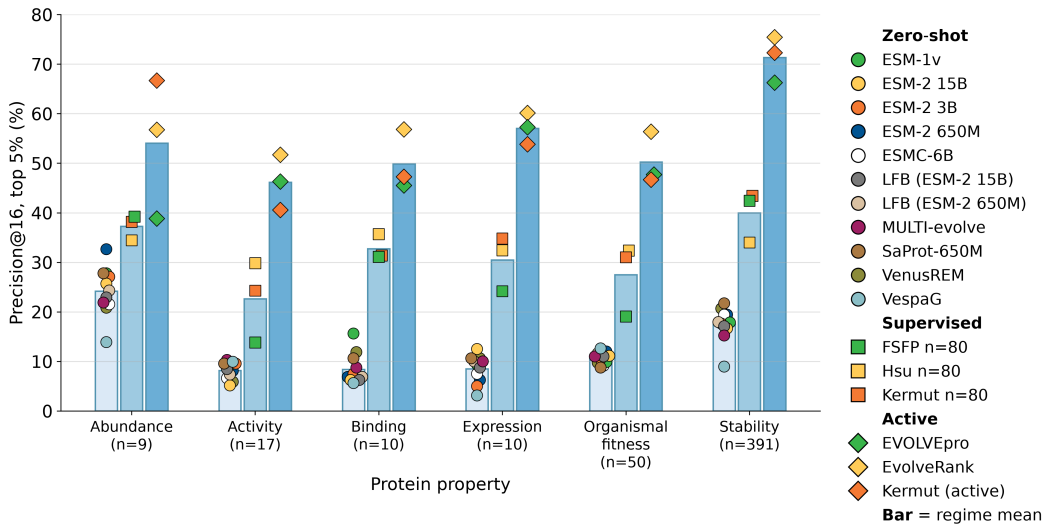

Figure E.1: **Precision@16 by protein property on the beneficial single-point mutation identification task.** Percentage of the 16 nominated variants falling in the true top 5% of fitness, across six property categories of the 487-dataset benchmark (n = datasets per category). Bars are the mean over the methods of each regime (Zero-shot, Supervised, Active; shades of blue); points are individual methods (color = method; marker = regime), each a mean over the category's datasets. Active learning outperforms all baselines across all property types.

Table E.1: **Active-learning ablation on the single-point benchmark (n=487).** Each surrogate (EvolveRank's pairwise ranker; Kermut's GP) is paired with a random or an active 80-variant training set; Kermut/EvolveRank-picks feeds the GP EvolveRank's measured set. All arms re-rank the full pool after 80 measurements; both EvolveRank arms use the same Swiss-tournament ranker. Values are means across 487 datasets over 10 seeds, 95% bootstrap CIs (1,000 resamples) in brackets. Best per column in bold (random baselines excluded). Acquisition dominates (random vs. active: all $p < 10^{-17}$); the ranker beats the GP on broad enrichment ($p < 10^{-8}$) but ties it at the very top ($p > 0.1$; two-sided Wilcoxon over datasets).

Method	P@16-10%	P@16-5%	P@16-1%	Recall@16	FoundTop1
Kermut / random (n=80)	54.8 [53.0, 56.6]	39.2 [37.5, 40.8]	12.3 [11.4, 13.1]	16.1 [15.0, 17.3]	18.2 [16.2, 20.2]
Kermut / EvolveRank-picks	70.7 [68.9, 72.5]	58.1 [55.9, 60.0]	21.4 [20.1, 22.7]	27.9 [26.2, 29.7]	**29.8 [27.0, 32.8]**
Kermut-AL (self)	84.7 [83.1, 86.1]	67.5 [65.3, 69.6]	21.0 [19.7, 22.4]	28.0 [26.1, 30.0]	28.2 [25.1, 31.3]
EvolveRank / random (n=80)	69.6 [68.6, 70.5]	43.1 [42.1, 44.1]	10.9 [10.4, 11.5]	14.2 [13.4, 15.0]	15.4 [13.7, 17.1]
EvolveRank / active (AL)	**91.0 [90.0, 91.8]**	**71.6 [69.8, 73.2]**	**21.5 [20.3, 22.6]**	**28.1 [26.5, 29.8]**	28.5 [25.7, 31.4]

Table E.2: **Ladder B: results averaged over protein families rather than individual datasets.** Each protein family contributes once (the 31 protein families; the correlated TrpB libraries are pooled into a single family), which prevents any one heavily sampled family from dominating. Values are means over protein families with 95% bootstrap CIs; the median split of Table 3 is applied to each family's mean epistasis. Ensembles merge their two constituent rankings by minimum rank. The additive/epistatic crossover is unchanged. Boldface marks the best point estimate per column within each block.

Method	N	Precision@16 ↑			Recall@16 ↑	FoundTop1 ↑
		top 10%	top 5%	top 1%		
All protein families (reciprocal-sign epistasis 0.00–0.31)						
MULTI-evolve	31	42.7 [36.4, 49.4]	26.6 [22.1, 31.4]	8.7 [7.1, 10.6]	43.8 [36.6, 50.9]	64.9 [54.3, 75.4]
Kermut	31	44.0 [37.2, 50.8]	27.5 [22.7, 32.7]	8.9 [7.2, 10.6]	45.0 [37.3, 52.5]	**64.9 [54.2, 75.6]**
Additive regression	31	43.0 [36.7, 49.3]	27.1 [22.5, 31.6]	8.9 [7.3, 10.7]	43.9 [36.8, 50.7]	63.7 [52.7, 74.0]
EVOLVEpro (ESM-2 15B)	31	39.1 [33.7, 45.2]	24.3 [20.3, 28.7]	7.7 [6.0, 9.5]	38.8 [32.8, 44.9]	57.0 [47.5, 66.5]
EvolveRank (ESM-2 650M)	31	41.1 [35.5, 46.9]	25.5 [21.3, 29.9]	7.9 [6.3, 9.8]	41.2 [34.9, 47.1]	58.3 [47.3, 68.5]
EvolveRank (ESM-2 15B)	31	42.4 [36.5, 48.8]	26.2 [21.9, 31.1]	8.3 [6.6, 10.3]	42.7 [36.0, 49.6]	59.6 [49.0, 70.1]
EvolveRank (SaProt-650M AF2)	31	42.8 [36.7, 49.2]	26.7 [22.2, 31.5]	8.6 [6.9, 10.3]	42.5 [36.1, 48.8]	61.5 [50.4, 71.7]
EvolveRank (SaProt-1.3B AF2)	31	41.0 [35.0, 47.6]	25.7 [21.2, 31.0]	8.3 [6.6, 10.1]	41.8 [35.3, 47.7]	60.0 [48.5, 70.2]
EvolveRank (ESMC-6B)	31	39.7 [34.2, 45.8]	24.7 [20.4, 29.6]	8.2 [6.6, 10.0]	40.4 [33.8, 46.8]	59.3 [48.5, 69.9]
EVOLVEpro (ESM-2 15B) + EvolveRank (ESM-2 15B)	31	41.8 [36.1, 48.2]	26.0 [21.8, 30.8]	8.4 [6.8, 10.3]	42.0 [35.7, 48.4]	60.6 [50.3, 71.3]
Kermut (ESM-2 650M) + EvolveRank (ESM-2 15B)	31	**44.6 [38.2, 51.2]**	**28.1 [23.7, 32.9]**	**9.0 [7.4, 10.8]**	**45.5 [38.3, 52.5]**	64.5 [54.0, 74.7]
More additive families (reciprocal-sign epistasis 0.00–0.19)						
MULTI-evolve	19	49.9 [43.6, 57.3]	31.2 [26.2, 36.9]	9.8 [8.0, 12.3]	50.7 [41.8, 59.1]	69.5 [57.4, 80.0]
Kermut	19	50.2 [42.5, 58.4]	31.8 [25.9, 38.1]	9.8 [8.0, 11.8]	**52.2 [42.5, 60.8]**	69.8 [57.9, 80.3]
Additive regression	19	50.5 [43.7, 57.7]	31.7 [26.7, 37.1]	**10.0 [8.2, 12.0]**	51.1 [42.8, 58.8]	68.9 [57.9, 79.0]
EVOLVEpro (ESM-2 15B)	19	42.9 [36.2, 49.9]	26.6 [21.7, 32.1]	8.0 [6.0, 10.4]	42.8 [35.0, 50.5]	57.6 [48.2, 67.1]
EvolveRank (ESM-2 650M)	19	45.8 [39.0, 52.7]	28.0 [22.9, 33.7]	8.2 [6.1, 10.7]	45.7 [37.7, 53.6]	59.5 [46.3, 72.6]
EvolveRank (ESM-2 15B)	19	47.4 [40.6, 55.1]	29.2 [24.1, 35.1]	8.7 [6.6, 11.3]	47.3 [38.6, 55.8]	60.3 [47.9, 71.6]
EvolveRank (SaProt-650M AF2)	19	47.3 [40.5, 54.7]	29.3 [24.5, 34.8]	9.0 [7.4, 11.0]	47.2 [38.3, 55.4]	64.5 [50.5, 77.4]
EvolveRank (SaProt-1.3B AF2)	19	45.9 [39.4, 53.0]	28.5 [23.5, 34.0]	8.5 [6.8, 10.3]	46.1 [37.7, 54.2]	62.9 [49.5, 76.3]
EvolveRank (ESMC-6B)	19	43.6 [36.9, 51.0]	27.1 [22.3, 32.6]	8.4 [6.8, 10.3]	44.5 [36.1, 53.0]	60.5 [46.3, 73.7]
EVOLVEpro (ESM-2 15B) + EvolveRank (ESM-2 15B)	19	46.2 [39.7, 53.6]	28.7 [23.8, 34.2]	8.9 [6.9, 11.3]	46.3 [38.2, 54.5]	63.4 [52.4, 74.2]
Kermut (ESM-2 650M) + EvolveRank (ESM-2 15B)	19	**50.6 [44.3, 58.0]**	**31.9 [27.1, 37.1]**	9.8 [8.1, 11.7]	52.0 [43.0, 60.3]	**69.8 [58.4, 80.0]**
More epistatic families (reciprocal-sign epistasis 0.20–0.31)						
MULTI-evolve	12	31.2 [25.1, 38.1]	19.3 [14.6, 25.2]	7.0 [4.7, 9.7]	32.9 [24.8, 42.2]	57.6 [37.6, 79.2]
Kermut	12	34.1 [27.4, 41.1]	20.8 [16.0, 26.0]	7.4 [5.0, 10.2]	33.7 [26.1, 42.1]	57.2 [38.7, 75.6]
Additive regression	12	31.2 [24.9, 38.4]	19.7 [14.9, 25.8]	7.2 [4.8, 10.2]	32.6 [24.9, 41.5]	55.4 [35.8, 76.2]
EVOLVEpro (ESM-2 15B)	12	33.2 [26.9, 40.3]	20.5 [15.8, 26.1]	7.2 [4.9, 10.0]	32.6 [25.5, 40.3]	55.9 [38.3, 73.5]
EvolveRank (ESM-2 650M)	12	33.8 [28.4, 39.5]	21.6 [17.0, 26.7]	7.5 [5.1, 10.0]	33.9 [27.0, 41.0]	56.5 [39.5, 73.8]
EvolveRank (ESM-2 15B)	12	34.4 [27.7, 42.4]	21.4 [16.2, 28.2]	7.7 [5.2, 10.8]	**35.4 [27.2, 44.0]**	**58.7 [40.3, 76.7]**
EvolveRank (SaProt-650M AF2)	12	**35.7 [28.0, 44.4]**	**22.5 [16.7, 29.7]**	7.8 [5.2, 11.2]	35.2 [28.1, 42.4]	56.8 [40.2, 74.3]
EvolveRank (SaProt-1.3B AF2)	12	33.3 [25.0, 42.8]	21.4 [14.5, 29.7]	**8.0 [5.1, 11.3]**	34.9 [27.6, 43.2]	55.5 [38.8, 73.0]
EvolveRank (ESMC-6B)	12	33.5 [26.5, 41.8]	20.9 [14.9, 28.4]	7.8 [5.2, 11.2]	33.8 [26.2, 42.5]	57.4 [42.0, 75.3]
EVOLVEpro (ESM-2 15B) + EvolveRank (ESM-2 15B)	12	34.9 [28.1, 42.9]	21.7 [16.4, 28.3]	7.7 [5.2, 10.9]	35.3 [27.3, 43.8]	56.2 [37.9, 75.0]
Kermut (ESM-2 650M) + EvolveRank (ESM-2 15B)	12	34.9 [27.8, 43.4]	22.2 [16.9, 29.0]	7.7 [5.2, 10.8]	35.3 [27.1, 44.5]	56.0 [37.1, 75.6]

Table E.3: **Ladder B: additional embedding backbones and ensembles, averaged over datasets.** Extended version of Table 3 with EvolveRank run on five embedding backbones and two minimum-rank ensembles. Values are per-dataset means over 10 replicate splits with 95% bootstrap CIs; landscapes are split at the median fraction of reciprocal-sign epistatic motifs. Boldface marks the best point estimate per column within each block.

Method	N	Precision@16 ↑			Recall@16 ↑	FoundTop1 ↑
		top 10%	top 5%	top 1%		
All landscapes (reciprocal-sign epistasis 0.00–0.34)						
MULTI-evolve	44	39.9 [34.1, 46.5]	24.9 [20.8, 29.9]	7.9 [6.4, 9.7]	41.2 [35.3, 47.4]	58.9 [49.3, 68.9]
Kermut	44	40.6 [33.9, 47.4]	25.6 [21.0, 30.6]	8.0 [6.6, 9.6]	41.7 [35.3, 48.2]	**59.2 [50.4, 68.7]**
Additive regression	44	40.0 [34.3, 46.5]	25.3 [21.2, 30.1]	8.0 [6.6, 9.6]	41.2 [35.3, 47.3]	58.0 [48.6, 68.4]
EVOLVEpro (ESM-2 15B)	44	37.2 [31.8, 43.6]	23.3 [19.6, 27.9]	7.2 [5.8, 8.9]	37.3 [32.1, 42.6]	54.3 [45.7, 63.6]
EvolveRank (ESM-2 650M)	44	38.6 [33.2, 44.8]	24.2 [20.0, 29.0]	7.4 [6.0, 9.1]	38.8 [33.7, 44.3]	55.2 [46.4, 64.6]
EvolveRank (ESM-2 15B)	44	39.8 [34.5, 46.2]	24.8 [20.9, 29.6]	7.7 [6.3, 9.5]	40.5 [35.0, 46.5]	56.8 [47.9, 66.6]
EvolveRank (SaProt-650M AF2)	44	40.0 [34.0, 46.8]	25.2 [20.8, 30.4]	8.0 [6.5, 9.8]	40.3 [35.0, 46.1]	58.0 [48.9, 67.0]
EvolveRank (SaProt-1.3B AF2)	44	38.6 [32.5, 45.2]	24.5 [19.9, 29.9]	7.7 [6.1, 9.6]	39.4 [34.1, 45.3]	55.2 [45.5, 65.2]
EvolveRank (ESMC-6B)	44	37.3 [31.9, 43.2]	23.4 [19.4, 28.1]	7.6 [6.2, 9.3]	38.2 [32.8, 43.7]	55.9 [46.6, 66.4]
EVOLVEpro (ESM-2 15B) + EvolveRank (ESM-2 15B)	44	39.5 [34.0, 45.7]	24.8 [20.8, 29.6]	7.8 [6.3, 9.6]	40.0 [34.6, 45.7]	56.8 [47.7, 66.6]
Kermut (ESM-2 650M) + EvolveRank (ESM-2 15B)	44	**41.5 [35.3, 47.9]**	**26.4 [22.0, 31.1]**	**8.2 [6.8, 9.8]**	**42.6 [36.8, 48.9]**	59.0 [50.0, 68.7]
More additive half (reciprocal-sign epistasis 0.00–0.19)						
MULTI-evolve	22	51.8 [45.6, 58.9]	32.2 [27.6, 38.2]	9.9 [8.2, 12.0]	51.4 [42.6, 59.9]	70.0 [58.2, 80.5]
Kermut	22	**52.9 [45.3, 61.8]**	**33.8 [28.0, 40.9]**	9.9 [8.3, 11.9]	**53.0 [43.9, 61.6]**	**70.3 [59.3, 80.0]**
Additive regression	22	52.4 [45.9, 58.9]	33.0 [28.4, 38.6]	**10.0 [8.5, 11.9]**	51.8 [44.0, 59.7]	69.5 [59.1, 80.0]
EVOLVEpro (ESM-2 15B)	22	45.9 [39.0, 53.4]	28.6 [23.8, 34.3]	8.4 [6.6, 10.7]	44.0 [36.8, 51.3]	59.5 [49.5, 68.6]
EvolveRank (ESM-2 650M)	22	47.8 [41.3, 55.5]	29.9 [24.7, 36.6]	8.8 [6.8, 11.2]	46.4 [39.6, 53.2]	62.3 [50.0, 74.1]
EvolveRank (ESM-2 15B)	22	48.8 [42.2, 56.5]	30.2 [25.4, 36.2]	9.0 [7.2, 11.3]	47.9 [39.9, 56.2]	61.4 [48.6, 72.7]
EvolveRank (SaProt-650M AF2)	22	50.1 [42.8, 58.4]	31.8 [26.2, 38.8]	9.9 [8.0, 12.2]	48.6 [40.9, 56.4]	68.2 [55.0, 79.6]
EvolveRank (SaProt-1.3B AF2)	22	49.2 [42.1, 57.6]	31.4 [25.3, 38.8]	9.6 [7.4, 12.3]	48.0 [39.9, 55.7]	66.4 [53.2, 78.2]
EvolveRank (ESMC-6B)	22	46.2 [39.6, 53.5]	29.3 [24.4, 35.1]	9.2 [7.4, 11.4]	45.9 [38.3, 53.4]	64.1 [50.9, 76.4]
EVOLVEpro (ESM-2 15B) + EvolveRank (ESM-2 15B)	22	48.6 [42.1, 56.1]	30.5 [25.6, 36.6]	9.3 [7.6, 11.6]	47.5 [39.7, 55.3]	65.0 [54.1, 75.5]
Kermut (ESM-2 650M) + EvolveRank (ESM-2 15B)	22	52.5 [46.2, 60.3]	33.3 [28.6, 39.3]	10.0 [8.5, 11.8]	52.4 [44.0, 60.9]	69.8 [59.4, 80.3]
More epistatic half (reciprocal-sign epistasis 0.20–0.34)						
MULTI-evolve	22	28.0 [21.1, 35.1]	17.5 [12.6, 22.8]	5.9 [4.0, 7.9]	31.1 [24.3, 38.3]	47.7 [33.6, 61.8]
Kermut	22	28.2 [21.1, 36.0]	17.5 [12.5, 22.7]	6.1 [4.3, 8.2]	30.4 [23.5, 37.2]	48.2 [35.0, 61.8]
Additive regression	22	27.6 [20.9, 34.8]	17.6 [12.7, 22.9]	5.9 [4.0, 8.0]	30.5 [23.8, 37.4]	46.4 [32.7, 60.0]
EVOLVEpro (ESM-2 15B)	22	28.6 [22.0, 35.5]	18.0 [13.3, 23.0]	6.0 [4.1, 8.2]	30.5 [24.2, 36.7]	49.1 [35.5, 62.7]
EvolveRank (ESM-2 650M)	22	29.3 [22.9, 35.8]	18.4 [13.7, 23.4]	6.0 [4.2, 8.0]	31.3 [24.7, 37.6]	48.2 [35.5, 61.4]
EvolveRank (ESM-2 15B)	22	**30.8 [23.8, 38.4]**	19.3 [14.3, 24.6]	**6.4 [4.5, 8.6]**	**33.1 [26.4, 39.7]**	**52.3 [39.1, 65.5]**
EvolveRank (SaProt-650M AF2)	22	29.9 [22.3, 38.0]	18.7 [13.4, 24.4]	6.1 [4.1, 8.4]	32.0 [25.1, 38.8]	47.7 [34.5, 61.4]
EvolveRank (SaProt-1.3B AF2)	22	28.0 [20.7, 35.8]	17.6 [12.3, 23.6]	5.8 [3.8, 8.2]	30.9 [23.9, 37.8]	44.1 [30.0, 58.2]
EvolveRank (ESMC-6B)	22	28.4 [21.8, 35.7]	17.5 [12.5, 22.9]	6.1 [4.1, 8.4]	30.5 [24.0, 37.1]	47.7 [34.5, 61.8]
EVOLVEpro (ESM-2 15B) + EvolveRank (ESM-2 15B)	22	30.4 [23.3, 37.8]	19.0 [14.0, 24.4]	6.2 [4.2, 8.5]	32.5 [25.9, 39.1]	48.6 [34.5, 62.3]
Kermut (ESM-2 650M) + EvolveRank (ESM-2 15B)	22	30.5 [23.0, 38.6]	**19.5 [14.3, 25.1]**	6.3 [4.3, 8.7]	32.7 [25.6, 39.7]	48.2 [35.0, 61.8]

E.3 Lattice

Table E.4 reports the full Lattice comparison, with the six protein families weighted equally and methods grouped by whether they begin from a random batch, use a zero-shot prior, or merge two rankings. Table E.4 also isolates the contribution of the ESM-2 650M zero-shot signal, comparing its use as a continuous Gaussian-process prior against its use as a one-off warm start.

Table E.4: **Lattice benchmark, averaged over protein family.** 6 protein families equally weighted (DHFR, GB1, ParD2, ParD3, TEV, TrpB; the 10 TrpB landscapes pooled). In the *ESM-2 650M prior* block, both methods use ESM-2 650M zero-shot scores, but Kermut only as its GP mean (keeping the shared random initiation), while warm-started EvolveRank replaces the initial variants with the zero-shot top-16. Ensembles merge the two rankings by minimum rank. Values are means over families with 95% bootstrap CIs (1,000 resamples); **N** = landscapes contributing. Boldface: best per column within each block.

Method (PLM)	N	Precision@16 ↑			Recall@16 ↑	FoundTop1 ↑
		top 10%	top 5%	top 1%		
Individual methods (shared random init)						
Random	15	9.7 [7.8, 12.3]	4.0 [3.0, 5.2]	1.2 [0.7, 1.6]	0.2 [0.0, 0.5]	0.2 [0.0, 0.5]
Zero-shot (ESM-2 650M)	15	77.6 [54.7, 95.3]	57.8 [37.5, 79.7]	29.5 [7.3, 54.5]	5.8 [0.0, 15.4]	11.7 [0.0, 35.0]
DPO (ESM3-1.4B)†	15	10.4 [8.3, 13.2]	9.4 [7.6, 11.9]	6.1 [4.5, 7.9]	2.8 [0.9, 4.7]	7.0 [0.3, 16.7]
ERA (ESM3-1.4B)†	15	52.0 [32.1, 72.6]	36.9 [19.1, 56.8]	12.0 [5.1, 20.2]	2.2 [1.0, 3.5]	3.3 [0.0, 10.0]
ALDE (GP-UCB)	15	87.7 [77.2, 96.7]	73.0 [61.3, 86.0]	46.8 [32.3, 61.5]	8.6 [3.8, 13.3]	10.5 [5.0, 16.7]
Kermut, without zero-shot	15	82.6 [67.2, 96.8]	77.3 [61.1, 91.9]	63.6 [45.8, 79.8]	19.8 [7.0, 32.8]	23.5 [8.3, 38.3]
EVOLVEpro (ESM-2 15B)	15	86.8 [70.3, 99.2]	82.5 [62.4, 98.5]	61.6 [44.7, 78.3]	18.5 [7.3, 29.0]	17.7 [6.7, 31.3]
EvolveRank (ESM-2 650M)	15	91.3 [84.7, 97.3]	**85.6 [75.0, 95.9]**	68.2 [53.8, 80.4]	19.8 [7.3, 32.7]	15.2 [1.8, 33.3]
EvolveRank (ESM-2 15B)	15	88.9 [80.4, 96.8]	80.1 [65.5, 93.6]	67.8 [47.0, 87.6]	27.5 [10.5, 43.8]	23.2 [4.8, 41.5]
EvolveRank (SaProt-650M AF2)	15	89.7 [78.0, 98.6]	82.7 [66.7, 96.2]	**70.2 [52.7, 87.6]**	26.4 [7.9, 45.0]	23.8 [5.5, 43.3]
EvolveRank (ESMC-6B)	15	**91.7 [84.8, 98.2]**	82.8 [69.6, 95.0]	69.2 [50.4, 86.8]	**28.4 [10.6, 45.4]**	**27.0 [5.0, 51.7]**
Individual methods with ESM-2 650M prior						
Kermut	15	93.3 [83.5, 99.9]	91.8 [80.5, 99.8]	80.9 [69.6, 90.0]	29.8 [12.6, 46.3]	39.2 [15.0, 59.2]
EvolveRank (ESM-2 650M), zero-shot top-16 init	15	**99.9 [99.7, 100.0]**	**97.9 [95.8, 100.0]**	**92.6 [85.3, 99.0]**	**36.5 [8.3, 67.7]**	**48.3 [15.0, 83.3]**
Ensembles (minimum-rank merge)						
EVOLVEpro (ESM-2 15B) + EvolveRank (ESM-2 15B)	15	91.8 [82.3, 99.8]	85.1 [71.5, 97.2]	69.3 [51.3, 87.2]	26.8 [11.9, 39.9]	27.7 [11.0, 44.3]
EvolveRank (ESM-2 650M) + EvolveRank (SaProt-650M AF2)	15	90.6 [83.2, 97.0]	84.3 [72.0, 94.9]	70.4 [54.9, 85.0]	31.2 [11.3, 51.1]	40.3 [16.7, 63.7]
EvolveRank (ESM-2 15B) + EvolveRank (SaProt-650M AF2)	15	88.9 [79.5, 96.4]	82.2 [68.3, 93.8]	68.3 [52.0, 84.1]	31.6 [12.7, 50.4]	36.3 [15.0, 59.7]
EvolveRank (ESMC-6B) + EvolveRank (SaProt-650M AF2)	15	91.2 [81.9, 98.9]	85.9 [72.3, 97.7]	69.9 [52.7, 86.2]	32.2 [13.0, 51.3]	35.3 [8.6, 62.0]
Kermut + EvolveRank (ESM-2 15B)	15	**92.9 [87.2, 97.8]**	**90.8 [83.7, 96.7]**	**81.8 [73.0, 89.2]**	**38.9 [16.0, 60.0]**	**49.3 [19.3, 77.7]**