Active Learning-Assisted Directed Evolution

Jason Yang [a,†], Ravi G. Lal [a,†], James C. Bowden [b,^], Raul Astudillo [b], Mikhail A. Hameedi [c], Sukhvinder Kaur [d], Matthew Hill [d], Yisong Yue [b,*], Frances H. Arnold [a,c,*]

[a] Division of Chemistry and Chemical Engineering, California Institute of Technology, Pasadena, California 91125, United States
[b] Division of Engineering and Applied Sciences, California Institute of Technology, Pasadena, California 91125, United States
[c] Division of Biology and Biological Engineering, California Institute of Technology, Pasadena, California 91125, United States
[d] Elegen Corp, 1300 Industrial Road #16, San Carlos, California 94070, United States

[†] These authors contributed equally.
[^] present address: Computer Science, University of California-Berkeley
*Corresponding Authors: Frances H. Arnold (frances@cheme.caltech.edu), Yisong Yue (yyue@caltech.edu)

ABSTRACT:
Directed evolution (DE) is a powerful tool to optimize protein fitness for a specific application. However, DE can be inefficient when mutations exhibit non-additive, or epistatic, behavior. Here, we present Active Learning-assisted Directed Evolution (ALDE), an iterative machine learning-assisted DE workflow that leverages uncertainty quantification to explore the search space of proteins more efficiently than current DE methods. We apply ALDE to an engineering landscape that is challenging for DE: optimization of five epistatic residues in the active site of an enzyme. In three rounds of wet-lab experimentation, we improve the yield of a desired product of a non-native cyclopropanation reaction from 12% to 93%. We also perform computational simulations on existing protein sequence-fitness datasets to support our argument that ALDE can be more effective than DE. Overall, ALDE is a practical and broadly applicable strategy to unlock improved protein engineering outcomes.

KEYWORDS: protein engineering, directed evolution, enzyme engineering, protoglobin, carbene, stereoselectivity, machine learning, Bayesian optimization, active learning, uncertainty quantification

INTRODUCTION

Protein engineering is an optimization problem, where the goal is to find the amino acid sequence that maximizes "fitness," a quantitative measurement of the efficacy or functionality for a desired application, from chemical synthesis to bioremediation and therapeutics. Protein fitness optimization can be thought of as navigating a protein fitness landscape, a mapping of amino acid sequences to fitness values, to find higher-fitness variants.[1] However, since protein sequence space is vast, as a protein of length N can take on 20^N distinct sequences and functional proteins are vanishingly rare, finding an optimal sequence is hard. Because functional proteins are surrounded by other functional proteins one mutation away,[2] protein engineers often use directed evolution (DE) to optimize protein fitness.[3,4]

In its simplest form, DE involves accumulating beneficial mutations by searching through sequences near one that exhibits some level of desired function for variants that exhibit enhanced performance on a target fitness metric (**Fig. 1a**). This approach can be thought of as greedy hill climbing optimization across the protein fitness landscape (**Fig. 1b**). DE is limited because screening for performance can only explore a small, local region of sequence space. Additionally, taking one mutational step at a time can cause the experiment to become stuck at a local optimum, especially on rugged protein fitness landscapes where mutation effects exhibit epistasis.[5] Machine learning (ML) techniques offer a pathway to circumvent these obstacles, providing strategies to more efficiently navigate these complex landscapes.[6–10]

While supervised ML has been used to propose ideal combinations of mutations–such as in ML-assisted DE (MLDE)[11,12]–these approaches are often limited to small design spaces as they do not take advantage of the fundamentally iterative manner in which protein engineering can take place in real-world applications. By contrast, active learning is an ML paradigm that gathers data

iteratively using a supervised model which is, in turn, updated as new data are acquired (**Fig. 1c**). By leveraging uncertainty quantification to choose which variants should be tested at each step, active learning has the potential to unlock improved engineering outcomes (**Fig. 1d**).[13–17] Approaches related to active learning have been used in the wet lab to optimize artificial metalloenzymes, nucleases, and other proteins.[18–20] Past work has also explored the use of Bayesian optimization (BO), a particular class of active learning algorithms, to experimentally improve the thermostability of protein chimeras[21,22] and to optimize proteins with one to several mutations.[13,23] However, few studies have explored the utility of active learning methods in comparison to DE, especially where epistatic effects are prevalent.[19,24] In addition, understanding of the practical role of uncertainty quantification in the context of deep learning[25–27] and high-dimensional[28] representations learned from protein language models[29,30] is limited.

To address the limitations of existing methods, we introduce Active Learning-Assisted Directed Evolution (ALDE), a computationally assisted workflow for protein engineering that employs batch Bayesian optimization. ALDE alternates between collecting sequence-fitness data using a wet-lab assay and training an ML model to prioritize new sequences to screen in the wet lab (**Fig. 1C**); it resembles existing wet-lab mutagenesis and screening workflows for DE and is generally applicable to any protein engineering objective. In this study, we use ALDE to find the ideal combination of five mutations in the active site of a biocatalyst based on a protoglobin from *Pyrobaculum arsenaticum* (*Par*Pgb) for performing a non-native cyclopropanation reaction with high yield and stereoselectivity. We chose this model system because the residues of interest are in close structural proximity and there is evidence of negative epistasis, which hinders DE. After performing three rounds of ALDE (exploring only ~0.01% of the design space), the optimal variant has 99% total yield and 14:1 selectivity for the desired diastereomer of the cyclopropane product.

The mutations present in the final variant are not expected from the initial screen of single mutations at these positions, demonstrating that the consideration of epistasis through ML-based modeling is important. We solidify our argument that ALDE is more effective than DE by computationally simulating ALDE on two combinatorially complete protein fitness landscapes. We also provide an extensive analysis of the effects of protein sequence encodings, models, acquisition functions, and uncertainty quantification for protein fitness optimization, to determine best practices for real-world engineering campaigns. In short, we find that frequentist uncertainty quantification works more consistently than typical Bayesian approaches, and incorporating deep learning does not always boost performance. Ultimately, we demonstrate that ALDE is a practical and effective tool for navigating protein fitness landscapes and provide experimental and computational tools (https://github.com/jsunn-y/ALDE) so that the method is easy to use and broadly applicable.

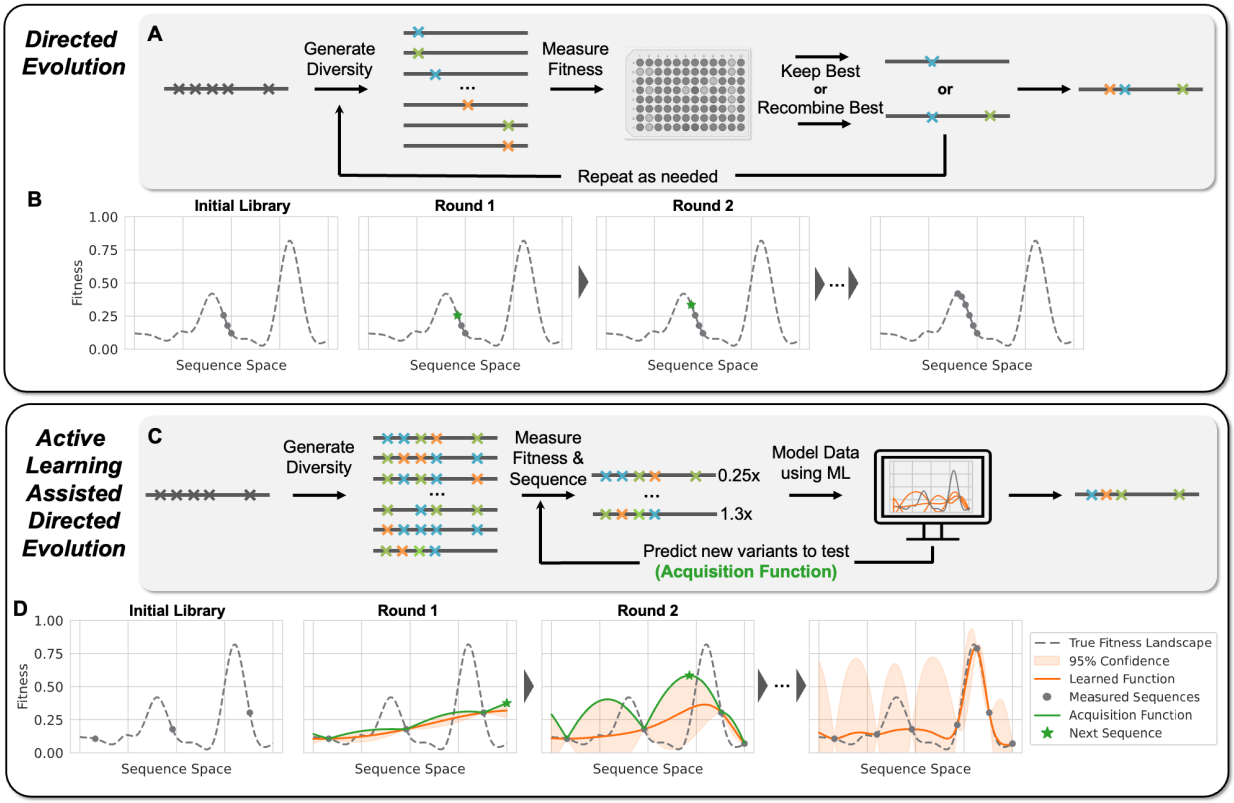


Fig. 1. Conceptual differences between DE and ALDE. (A) A common workflow for DE, where a starting protein is mutated and fitnesses of variants are measured (screened). The best variant is used as the starting point for the next round of mutation and screening, until desired fitness is achieved. **(B)** Conceptualization of DE as greedy hill climbing optimization on a hypothetical protein fitness landscape. **(C)** Workflow for ALDE. An initial training library is generated, where *k* residues are mutated simultaneously (for example *k=5*). A small subset of this library is randomly picked, after which the variants are sequenced and their fitnesses are screened. A supervised ML model with uncertainty quantification is trained to learn a mapping from sequence to fitness. An acquisition function is used to propose new variants to test, balancing exploration (high uncertainty) and exploitation (high predicted fitness). The process is repeated until desired fitness is achieved. **(D)** Conceptualization of active learning on a hypothetical protein fitness landscape. Active learning is often more effective than DE for finding optimal combinations of mutations. In these conceptualizations, a single sequence is queried in each round, but in practical settings, active learning operates in batch where multiple sequences are tested in each round.

RESULTS

Practical implementation of ALDE

Broadly, ALDE alternates between library synthesis/screening in the wet lab to collect sequence-fitness labels and computationally training an ML model to learn a mapping from sequence to fitness in order to suggest a new batch of sequences to test (**Fig. 1c**), resembling batch BO. Before beginning ALDE, a combinatorial design space on *k* residues is defined, corresponding

to 20^k possible variants. The choice of *k* will vary depending on the system, as larger values of *k* can consider a greater extent of epistatic effects (allowing for better possible outcomes) but will likely require collecting more data to find an optimal variant. First, those *k* residues are simultaneously mutated, and an initial round of sequence-fitness data is collected in the wet lab. ALDE is compatible with low-N, batch protein engineering settings where tens to hundreds of sequences are screened in each round. The collected sequence-fitness data are then used to computationally train a supervised ML model that can predict sequence from fitness. Different ways to encode protein sequence numerically and different types of models which can provide uncertainty quantification are analyzed in this study. Afterward, an acquisition function is applied to the trained model to rank all sequences in the design space, from most to least likely to have high fitness. Several acquisition functions are evaluated in this study, to balance *exploration* of new areas of protein space with *exploitation* of variants that are predicted to have high fitness (**Fig. 1d**). The computational component of ALDE can be performed using the codebase at https://github.com/jsunn-y/ALDE. For the next round of ALDE, the top N variants from the ranking are then assayed in the wet-lab to provide additional sequence-fitness data, and the cycle is repeated until fitness is sufficiently optimized.

The active site of ParPgb is a challenging design space for standard DE

To initiate wet lab studies with ALDE, we identified a target enzymatic activity on a protein design space that would be difficult to engineer with simple DE methods. Enzyme-catalyzed carbene transfer reactions have the potential to be useful in many synthetic chemistry applications, and thus we decided to focus on the cyclopropanation of 4-vinylanisole (**1a**) using ethyl diazoacetate (**EDA**) as a carbene precursor to afford the 1,2-disubstituted cyclopropanes *trans*-**2a**

and *cis*-**2a** (**Fig. 2a**). Enzyme engineering for styrenyl cyclopropanation poses a stimulating challenge for evolution toward two properties, higher yield *and* improved selectivity toward one of the diastereomers of the cyclopropane product. While this non-native chemistry has been demonstrated with cytochromes P411,[31] we decided to engineer this activity in a protoglobin. Protoglobins are archaeal hemoproteins, which are attractive engineering targets due to their high thermostability (T_{50} ~ 60°C), small size (~200 amino acids),[32] and ability to perform novel carbene and nitrene transfer chemistries.[33–36] After screening a diverse set of protoglobins, including wild-types and engineered homologs, for cyclopropanation activity (**Fig. S31** of **Supplementary Information**), we decided to proceed with *Par*Pgb W59L Y60Q (ParLQ) as a starting point (parent) for ALDE. The ParLQ variant demonstrates only moderate cyclopropanation yield (~40% yield) and stereoselectivity (3:1 preferring *trans*-**2a**) under screening conditions. Because our goal was to arrive at a variant with high yield and high selectivity for the *cis*-product, we defined the objective to be explicitly optimized as the difference between the yield of *cis*-**2a** and the yield of *trans*-**2**a.

Based on previous engineering studies using protoglobin scaffolds, we selected five active-site residues (W56, Y57, L59, Q60, and F89; WYLQF) positioned above the distal face of the heme cofactor, which display epistatic effects and are known to impact non-native activity (**Fig. 2b**).[34,35] Single-site saturation mutagenesis (SSM) was performed at these sites, and variants were screened by gas chromatography for their cyclopropanation products. None of the screened mutants demonstrated a significant, desirable shift in the value of the objective (**Fig. 2c**) or related metrics such as *cis* yield and *cis*/*trans* selectivity (**Extended Data Fig. 1**). Given these data, a protein engineer might opt to perform a simple recombination of all positive variants to exploit the typically additive character of mutations.[37] However, in our recombination studies of the single-

site mutants with the highest fold-change in *cis* yield (DAYFW), the objective (DGMDW), or the selectivity (DHMVW), respectively, we did not observe a variant which generated ***cis*-2a** with high yield and selectivity (**Fig. 2d**). Overall, these findings suggest that our design problem is quite challenging for standard DE approaches.

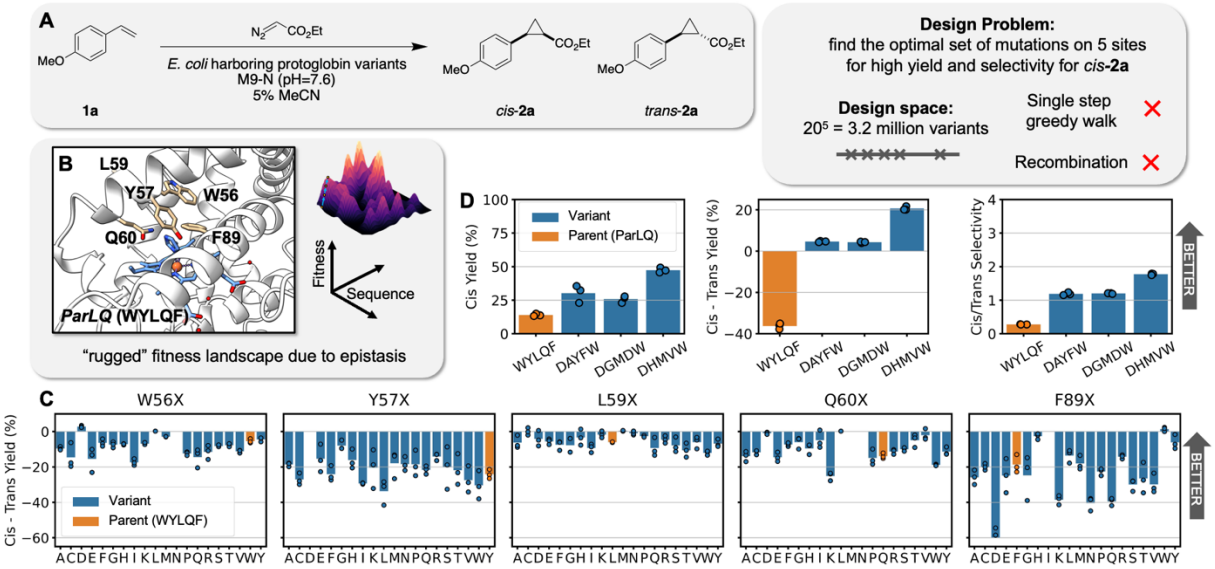


Fig. 2: A challenging, epistatic protein design space: optimization of five active site residues in *Par*Pgb. (A) Our objective was to optimize an enzyme to catalyze the formation of the *cis* product of a cyclopropanatiom reaction with high yield and high selectivity, which we quantify in a single value as *cis – trans* Yield. **(B)** The parent protein ParLQ is two mutations (W59L and V60Q) away from the wild-type ParPgb sequence. Five residues in the active site of ParLQ which were likely to exhibit epistasis were targeted: W56, Y57, L59, Q60, and F89. **(C)** The single mutations from parent at the five targeted sites do not offer significant improvements to the objective of *cis – trans* Yield. Very few single-mutation variants have the desired selectivity (positive *cis – trans* Yield), and it would not be obvious which variant to take forward in a DE campaign. Parent yields vary between runs but consistently show moderate yield and selectivity for the t*rans* product. **(D)** Various recombinations of ideal single mutations are not effective proteins for the desired objective (*cis – trans* Yield), and related metrics such as *cis* Yield and *cis/trans* Selectivity. DAYFW, DGMDW, and DHMVW are the ideal combinations of single mutations naively predicted to have the highest *cis* Yield, *cis – trans* Yield (objective), and *cis*/*trans* Selectivity, respectively. Overall, these results suggest an optimization problem that is challenging for standard DE methods.

Using ALDE to efficiently optimize ParPgb for a non-native carbene transfer reaction

With the design space confined to five residues and a well-defined objective, we began an ALDE engineering campaign. First, we synthesized an initial library of ParLQ variants which were mutated at all five positions under study (**Fig. 3a**). Mutants in this library were generated through sequential rounds of PCR-based mutagenesis methods utilizing NNK degenerate codons. We

elected to use random selection from this library because we did not know if any zero-shot predictors might enrich the starting library with useful variants.[7,12] In fact, retrospective analysis of the initial library revealed that our objective is not strongly correlated with conventional zero-shot predictors,[12,38] likely because the objective involves non-native chemistry, for which fitness is not sufficiently captured by evolutionary or stability-based metrics alone (**Extended Data Fig. 2**). Four 96-well plates of these random variants were picked and sequenced using the LevSeq long-read pooled sequencing method (**Fig. S7–10** of **Supplementary Information**),[39] yielding 216 unique variants without stop codons. Screening revealed that nearly all of the variants had higher cyclopropanation activity than free-heme background activity, likely because *Par*LQ was moderately active to begin with, and its high thermostability allows it to tolerate multiple mutations. The majority of variants displaying improved cyclopropanation yield strongly favored formation of *trans*-**2a**; however, several of the randomly selected sequences were capable of forming *cis*-**2a** in much higher yield than any previously tested ParLQ variant (**Fig. 3b**). Notably, the F89Y mutation was particularly important for inverting selectivity to favor the *cis*-**2a**, but only in the context of certain mutations at positions 56, 57, 59, and 60.

The ALDE computational package was used to train a predictive model on sequences and labels in the initial 216-member library and to suggest sequences for testing based on our acquisition function. Based on our extensive computational simulations (described in the following section), we decided to use the DNN ensemble with one-hot encoding of the five targeted residues for model training and Thompson sampling as the acquisition function. Genes encoding the top 90 amino acid sequences, optimized for expression in *E. coli*, were prepared by exact DNA synthesis for screening (Round 1, **Fig. 3b**). Details regarding DNA sequence design are described in the included supplementary materials. Subsequent activity screening showed that nearly a third of

Round 1 sequences met the objective better than the best variant in the initial, randomly selected set (**Fig. 3b**). The best variant in the Round 1 library, MKFNY (W56M Y57K L59F Q60N F89Y), demonstrated a total cyclopropanation yield of 93% and a *cis:trans* selectivity ratio of 12:1.

We then gave the newly collected data back to the ALDE computational algorithm for a second round of active learning. The top 90 predicted sequences were again synthesized and tested exactly as before (Round 2, **Fig. 3c**). Interestingly, the model *explored* the sequence space more in this round, as reflected in the expanded mutational diversity present in Round 2 and the increased variance in the activities of these sequences (both reaction yield and diastereoselectivity) (**Fig. 3d–f**). Impressively, the top-performing variant among these sequences (MPFDY) displayed a total cyclopropane yield of 99% and a 14:1 *cis:trans* selectivity ratio. None of the mutations in MPFDY obviously optimized the objective in the single-site mutagenesis studies (**Fig. 2c**); they work together, however, to deliver an optimal variant. Furthermore, after screening the reaction products of all predicted variants with chiral gas chromatography methods, we found that all of these sequences were generally capable of generating ***cis*-2a** in high enantiopurity (**Extended Data Fig. 3**).

Having concluded the ALDE-based evolutionary campaign with substrate **1a,** we sought to understand the substrate scope of the sequences explored in this project. We screened eight styrene derivatives (**1b**–**1i**) for cyclopropanation using the sequences from Round 2 of ALDE (**Extended Data Fig. 4**). The variants show different yields for each of the substrates, even though some of these substrates differed from **1a** only by a single atom. Nevertheless, for every substrate, nearly all of the Round 2 variants were higher yielding and more selective for their respective *cis*-diastereomers than the parent protein, ParLQ (**Fig. S51–66** of **Supplementary Information**).

Interestingly, the top-performing variants for each substrate differed in sequence from MPFDY, the top enzyme for **1a** cyclopropanation.

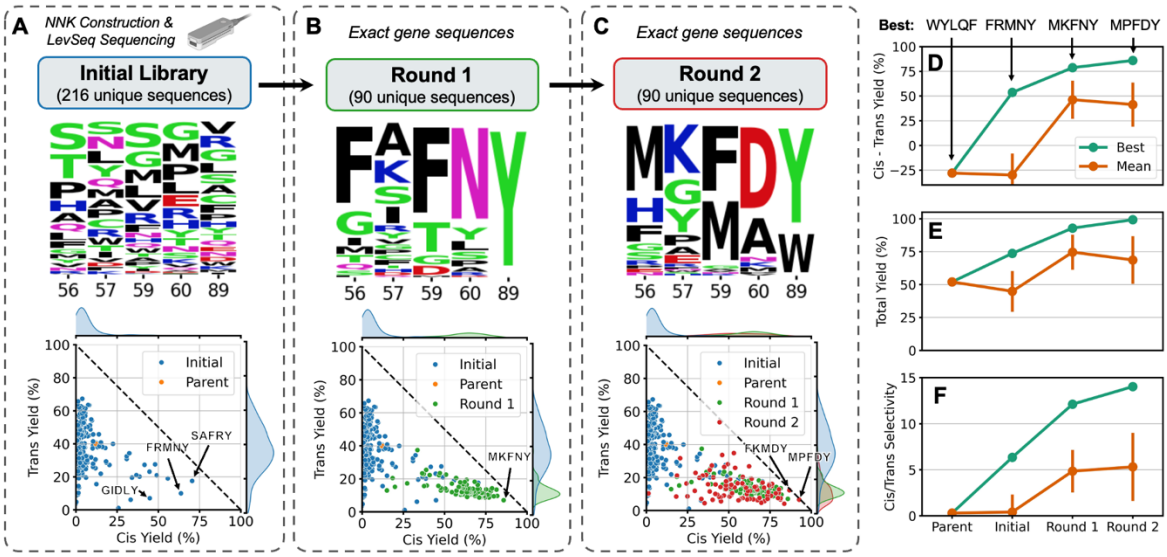


Fig. 3. ALDE optimization trajectory on the *Par*Pgb active site. The optimization campaign started with **(A)** constructing an initial library with mutations at all five sites under study using NNK degenerate codons, randomly selecting 384 for screening for product formation, and mapping to sequences using LevSeq. This was followed by two rounds of ALDE–**(B)** Round 1 and **(C)** Round 2. In Round 1 and Round2, exact genes were ordered as ENFINIA DNA produced by Elegen Corp. and screened for product formation. For each round, we present the distribution of amino acids sampled at each site and the distribution of yields for the *cis* and *trans* products, with a few of the top-performing variants labeled. Overall improvement in **(D)** *cis – trans* Yield, **(E)** Total Yield, and **(F)** *cis*/*trans* Selectivity over several rounds of ALDE for the best variant in each round and the mean across variants in each round. The best variant in each round, defined by the obejctive of *cis – trans* Yield is labeled. Error bars indicate the standard deviation across variants in the round.

Computational simulations on combinatorial protein datasets support the utility of ALDE

The design choices used for the wet-lab ALDE campaign were determined by performing computational simulations on two combinatorial landscape datasets for GB1[40] and TrpB[41]. On these landscapes, fitnesses have been measured experimentally for nearly all of the 20^4 = 160,000 variants in a library where four amino acid residues were mutated to all possible amino acids. GB1 refers to the B1 domain of protein G, an immunoglobin binding protein where fitness is measured by binding affinity-based sequence enrichment. The fitness of TrpB, the β-subunit of tryptophan synthase, was measured by coupling growth to the rate of tryptophan formation. Our baseline was DE greedy walk, where one residue was mutated to all possible amino acids, the best mutation was

fixed, and the process was repeated at each of the residues under study (**Fig. 4A**). DE simulations were performed from all active variants as starting points, using all 24 possible orders to enumerate the residues under interest.

The ALDE simulation consisted of batch BO, as shown in **Fig. 4b**. In each simulation, a random batch of 96 initial samples was selected, followed by four rounds of 96 samples each, with the surrogate model retrained and proposing new samples (via the acquisition function) in each round. This simulation setup was chosen to closely imitate a real wet-lab active learning campaign. The different parameters explored for ALDE, including encodings, models, and acquisition functions, are summarized in **Table 1**. We expanded the analysis beyond Gaussian process (GP) models, which are the typical surrogate models for BO, to deep kernel learning (DKL) models[25,27] and frequentist models based on boosting and deep neural network (DNN) ensembles. This was motivated by the rise of high dimensional encodings of protein sequences, such as those from protein language models (i.e. ESM2[29]), which have shown utility in certain property prediction tasks.[42,43] Visualizations of the acquisition functions (greedy, upper confidence bound (UCB), and Thompson sampling (TS)) on hypothetical models are given in **Fig. 4c**, with more details in **Methods**.

Table 1. Summary of encodings of protein sequences, models, and acquisition functions tested in this work.

Encoding	Dimension per Residue		Description
AAIndex	4		Continuous fixed amino acid descriptors
Georgiev[44]	19		Continuous fixed amino acid descriptors
Onehot	20		Categorical (which amino acid)
ESM2[29]	1280		Learned embedding from a protein language model (ESM2 with 650 million parameters)
Model	Bayesian?	Deep Learning?	Description
Boosting Ensemble	N	N	An ensemble of 5 boosting models
Gaussian Process (GP)	Y	N	A collection of continuous functions described by a posterior

DNN Ensemble	N	Y	An ensemble of 5 multilayer perceptrons (deep neural networks, DNNs)
Deep Kernel Learning (DKL)[25]	Y	Y	A GP on the last layer of a deep neural network

Acquisition Function	Deterministic?	Description
Greedy	Y	Acquires the maximum value of the mean from the posterior
Upper Confidence Bound (UCB)	Y	Acquires the maximum value of a certain confidence interval from the posterior (tuned by a hyperparameter)
Thompson Sampling (TS)	N	Acquires the maximum value of a random function sampled from the posterior

The performance of each simulated ALDE campaign was quantified as the maximum fitness achieved at the end of the campaign, normalized to the variant with maximum fitness in the design space (**Fig. 4d**). Full optimization trajectories at each iteration of the campaign are provided in **Extended Data Fig. 5**. We conclude that active learning can significantly outperform the average performance of DE and random sampling, and results are consistent across the two different protein datasets. ALDE also outperforms a single round of MLDE (**Extended Data Fig. 6**). Higher dimensional encodings (Onehot and ESM2) generally work better with deep learning-based models (DNN Ensemble and DKL), while non-deep learning models might learn better from low dimensional AAIndex and Georgiev parameters. The simulations further suggest that encodings from protein language models may not offer much benefit, which corroborates previous findings[12] but stands in contrast to other protein properties that can be predicted more effectively by transfer learning from protein language models.[20,42,43] ESM2 encodings cannot be used by GPs, likely because they are too high dimensional. In our acquisition functions, samples in the batch were sampled independently of each other. We also explored batch expected improvement,[45] but this ran extremely slowly without noticeable improvement in performance. Overall, the frequentist ensemble models perform the most consistently across different encodings.

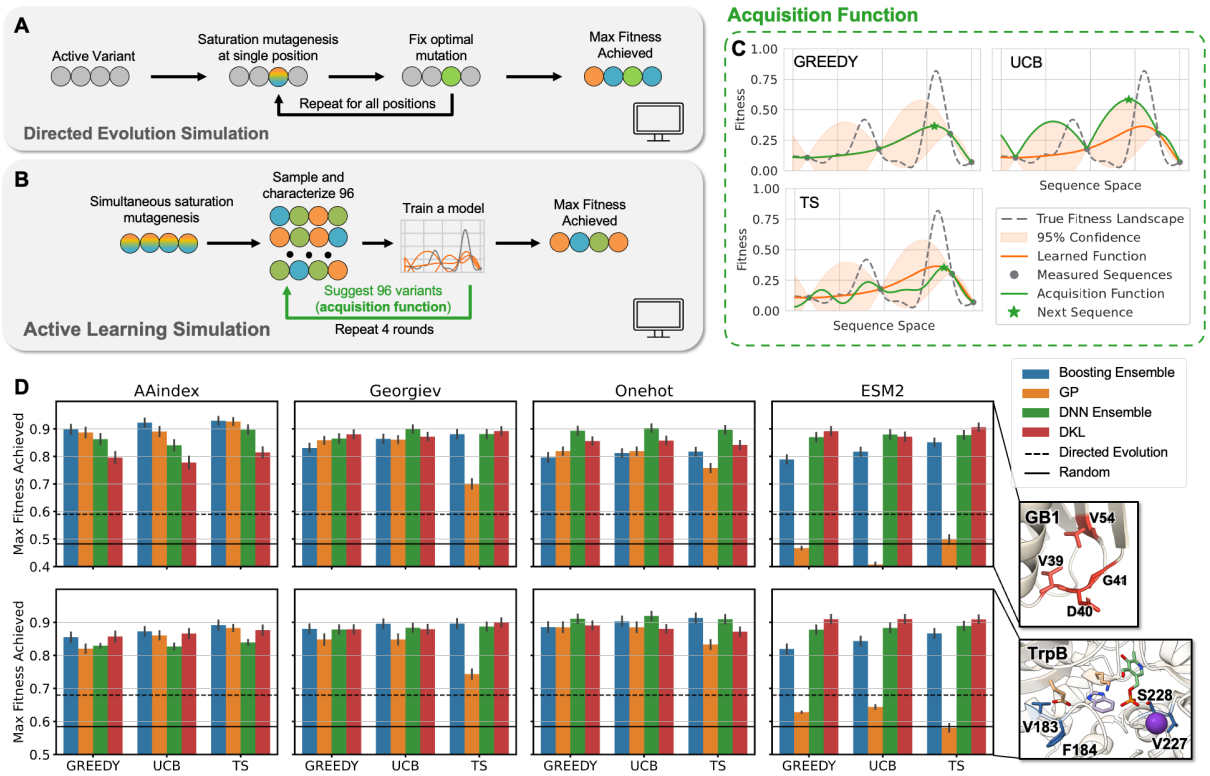


Fig. 4. Performance of simulated ALDE campaigns on two combinatorially complete protein datasets, GB1 and TrpB. (A) Each DE simulation as a greedy single-step walk on four residues, where each residue is fixed to the optimal mutation until all four residues have been iterated across. DE simulations start from every variant that has some measurable function, with all 24 possible orderings of four residues simulated. **(B)** Each ALDE simulation starts from a random sample of 96 variants on the 4-site landscape, with four rounds of learning and proposing new sequences to test, each with 96 protein variants. **(C)** Hypothetical visualization of the three acquisition functions explored in this work: greedy, upper confidence bound (UCB), and Thompson sampling (TS). **(D)** ALDE for four encodings, four models, and three acquisition functions generally outperforms the average DE simulation and random sampling on the GB1 and TrpB datasets. Performance is quantified as the normalized maximum fitness achieved by the end of the ALDE campaign. Error bars indicate standard deviation across 70 random initializations.

To better understand which models are the most advantageous, we assessed how well calibrated their uncertainties were (**Fig. 5A**). For a calibrated model, an n% confidence interval should contain n% of true labels across different values of n, which can be evaluated and visualized based on a calibration curve. Hypothetical calibrated, underconfident, and overconfident models are visualized in **Fig. 5B**, with their associated calibration curves. The calibration curves for different encodings and models are given in **Extended Data Fig. 7**. The area between a calibration curve and perfect calibration (dashed line) is defined as its miscalibration area, which should be low. Another way to measure uncertainty calibration is by measuring the Spearman correlation

between uncertainty from the model (σ) and the mean absolute error from the model (MAE), which should be high.

Overall, the Boosting and DNN Ensembles have the lowest MAEs, which suggests that they are the most accurate models (**Fig. 5A**). DNN ensembles have the lowest miscalibration areas, suggesting that they are the most calibrated and best models overall. These results are generally consistent across encodings and datasets, with a few outliers. In general, calibrated uncertainty is desirable,[46,47] and it is thought that it is important to understand how calibration shifts when extrapolating beyond the training set.[48,49] However, in this study, we find that performance in ALDE simulations (by max fitness achieved) is not necessarily correlated to how calibrated the uncertainties are for each model. For example, DKL performs the best for the ESM2 encoding, but these models have the least calibrated uncertainties and the highest MAEs. Because calibration is measured on the entire combinatorial design space, it may not directly correspond to the ability to find an optimal variant.

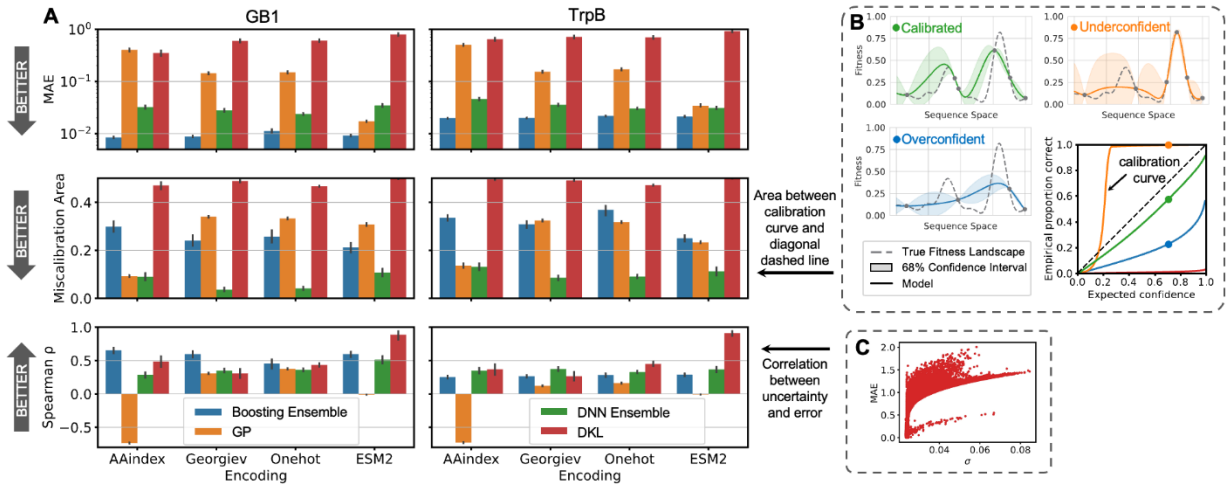


Fig. 5. Analysis of uncertainty quantification on simulated ALDE campaigns. (A) Metrics used to evaluate how well calibrated each of the four models are for four encodings. Metrics for evaluation are the mean absolute error (MAE), the miscalibration area for the calibration curve, and the Spearman correlation between uncertainty and error. All metrics are calculated based on all measured points in the combinatorial design space. All results are based on the campaigns using UCB as the acquisition function, during the final round of the campaign. Error bars indicate standard deviation across 70 random initializations. **(B)** Visualizations of three hypothetical models with underconfident, calibrated, and overconfident uncertainties, and their respective calibration curves. **(C)** Visualization of how the Spearman correlation between uncertainty and error is calculated.

DISCUSSION

Overall, ALDE is an effective method for navigating protein fitness landscapes, and it offers several advantages compared to DE. First, ALDE can unlock engineering outcomes not possible with simple DE. By considering multiple interacting positions, ALDE can search for combinations of mutations which may demonstrate desirable epistatic effects[50,51] and reduce the risk of getting trapped at a local optimum. By combining mutations, it can also optimize multiple properties simultaneously. We demonstrated the advantage of ALDE on *Par*Pgb as a particular wet-lab case study – though proof is not possible without testing every DE greedy single-step walk (which is experimentally intractable in the wet lab). Computational simulations of ALDE support this conclusion, as ALDE consistently outperforms MLDE and DE baselines. Interestingly, we found that frequentist ensembles work the best in terms of performance and uncertainty quantification,[26,52] rather than Bayesian approaches such as typical GP models used in BO. Other ways to quantify uncertainty and improve overall performance could be explored in the future.[15,26] Overall, classical notions of uncertainty quantification seem to play a more limited role than expected in these real-world applications.

In the wet-lab engineering campaign, we were pleased to find that ALDE enabled access to a broader enzyme substrate scope, whereas using DE often "locks" one into high yield for only a single substrate or closely related ones. Here, we observe an emergent advantage inherent to ALDE: since sequences that balance *exploration* and *exploitation* for a given task are proposed, they can be serendipitously proficient at related tasks.

ALDE is enabled by several recent advancements in biotechnology. For the initial library constructed using degenerate codons, high-throughput sequencing was necessary to identify the sequences of variants in each well. For this work we utilized LevSeq,[39,53] a method that leverages

real-time nanopore sequencing. Furthermore, rapid and reliable access to directly synthesized DNA (Elegen Corp.) was instrumental to the speed with which evolution was performed. The ALDE workflow was significantly enhanced with (1) the delivery of exact genes in one week, which shortened time between rounds of evolution, (2) the high fidelity of the delivered gene products meant that no sequencing was required for Rounds 1 and 2 of ALDE, and (3) no over-screening was needed because the exact sequences were arrayed individually. Overall, the time and screening cost of the wet-lab engineering campaign with ALDE was lower than for a greedy walk strategy with DE. A total of six 96-well plates were screened before arriving at a final variant: four plates of random variants, and two plates of predicted sequences within three rounds. By comparison, a greedy walk with DE would have required around five rounds of evolution with increased screening in the later rounds, which would require greater experimental resources such as chemical reagents and analysis time. We expect that exact gene synthesis will be increasingly important for powering active learning workflows in protein engineering.

In this work, we illustrated ALDE's power for simultaneously increasing the activity and selectivity of an enzyme for a non-natural reaction, but ALDE is a general workflow that can be used for a broad range of protein engineering applications. Additionally, ALDE could be integrated into robotic systems for automated and efficient protein engineering workflows, and library design could utilize tools such as DeCOIL.[54] While we only engineered on five residues in this study, ALDE should naturally extend to even larger design spaces on more residues, as long as assay-labeled data is collected on variants with mutations spread across those residues. This will require some initial domain knowledge or screening to identify sites that will be useful for increasing fitness; library design could also benefit from limiting the number of simultaneous mutations or using zero-shot scores.[12,18] Future work here may also involve generative modeling if it is not

possible to enumerate the acquisition function on the entire design space. Overall, accompanied by a user-friendly codebase, ALDE is a broadly applicable tool that can unlock more efficient and effective protein engineering.

Contributions

J.Y. : conceptualization, methodology, software, investigation, analysis, writing – original draft, writing – editing

R.G.L : conceptualization, methodology, investigation, analysis, writing – original draft, writing – editing

J.C.B : methodology, software, writing – editing

R.A. : methodology, software, writing – editing

M.A.H. : investigation, writing – editing

S.K.: resources, DNA synthesis

M.H.: resources, DNA synthesis

Y.Y. : resources, writing – editing, supervision, funding

F.H.A : resources, writing – editing, supervision, funding

Acknowledgements

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Award Number DE-SC0022218. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. This work was also supported by the NSF Division of Chemical, Bioengineering, Environmental and Transport Systems (CBET 1937902). J.Y. and R.G.L. are partially supported by National Science Foundation Graduate Research Fellowships. The authors thank Yueming Long and Emre Guersoy for help with sequencing and Shilong Gao for collecting useful initial data. The authors also thank Christopher Yeh for helpful discussions and Sabine Brinkmann-Chen for critical reading of the manuscript.

REFERENCES

1. Romero, P. A. & Arnold, F. H. Exploring protein fitness landscapes by directed evolution. *Nat Rev Mol Cell Bio* **10**, 866–876 (2009).
2. Smith, J. M. Natural selection and the concept of a protein space. *Nature* **225**, 563–564 (1970).
3. Packer, M. S. & Liu, D. R. Methods for the directed evolution of proteins. *Nat. Rev. Genet.* **16**, 379–394 (2015).
4. Wang, Y. *et al.* Directed Evolution: Methodologies and Applications. *Chem. Rev.* **121**, 12384–12444 (2021).
5. Miton, C. M., Buda, K. & Tokuriki, N. Epistasis and intramolecular networks in protein evolution. *Curr. Opin. Struct. Biol.* **69**, 160–168 (2021).
6. Yang, K. K., Wu, Z. & Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods* **16**, 687–694 (2019).
7. Yang, J., Li, F.-Z. & Arnold, F. H. Opportunities and Challenges for Machine Learning-Assisted Enzyme Engineering. *ACS Cent. Sci.* **10**, 226–241 (2024).
8. Wittmann, B. J., Johnston, K. E., Wu, Z. & Arnold, F. H. Advances in machine learning for directed evolution. *Curr. Opin. Struct. Biol.* **69**, 11–18 (2021).
9. Freschlin, C. R., Fahlberg, S. A. & Romero, P. A. Machine learning to navigate fitness landscapes for protein engineering. *Curr. Opin. Biotechnol.* **75**, 102713 (2022).
10. Aghazadeh, A. *et al.* Epistatic Net allows the sparse spectral regularization of deep neural networks for inferring fitness functions. *Nat. Commun.* **12**, 5225 (2021).
11. Wu, Z., Kan, S. B. J., Lewis, R. D., Wittmann, B. J. & Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proc. Natl. Acad. Sci.* **116**, 8852–8858 (2019).
12. Wittmann, B. J., Yue, Y. & Arnold, F. H. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Syst.* **12**, 1026-1045.e7 (2021).
13. Qiu, Y., Hu, J. & Wei, G.-W. Cluster learning-assisted directed evolution. *Nat. Comput. Sci.* **1**, 809–818 (2021).
14. Qiu, Y. & Wei, G.-W. CLADE 2.0: Evolution-Driven Cluster Learning-Assisted Directed Evolution. *J. Chem. Inf. Model.* **62**, 4629–4641 (2022).
15. Hie, B., Bryson, B. D. & Berger, B. Leveraging Uncertainty in Machine Learning Accelerates Biological Discovery and Design. *Cell Syst.* **11**, 461-477.e9 (2020).
16. Greenman, K. P., Amini, A. P. & Yang, K. K. Benchmarking Uncertainty Quantification for Protein Engineering. *bioRxiv* (2023).
17. Hie, B. L. & Yang, K. K. Adaptive machine learning for protein engineering. *Curr. Opin. Struct. Biol.* **72**, 145–152 (2022).
18. Vornholt, T. *et al.* Enhanced Sequence-Activity Mapping and Evolution of Artificial Metalloenzymes by Active Learning. *ACS Cent. Sci.* **10**, 1357–1370 (2024).
19. Thomas, N. *et al.* Engineering highly active and diverse nuclease enzymes by combining machine learning and ultra-high-throughput screening. *bioRxiv* (2024).
20. Jiang, K. *et al.* Rapid protein evolution by few-shot learning with a protein language model. *bioRxiv* (2024).
21. Romero, P. A., Krause, A. & Arnold, F. H. Navigating the protein fitness landscape with Gaussian processes. *Proc. Natl. Acad. Sci.* **110**, E193–E201 (2012).
22. Rapp, J. T., Bremer, B. J. & Romero, P. A. Self-driving laboratories to autonomously navigate the protein fitness landscape. *Nat. Chem. Eng.* **1**, 97–107 (2024).

23. Hu, R. *et al.* Protein engineering via Bayesian optimization-guided evolutionary algorithm and robotic experiments. *Brief. Bioinform.* **24**, bbac570 (2023).
24. Gantz, M. Microdroplet screening rapidly profiles a biocatalyst to enable its AI-assisted engineering. *bioRxiv* (2024).
25. Wilson, A. G., Hu, Z., Salakhutdinov, R. & Xing, E. P. Deep Kernel Learning. *arXiv* (2015).
26. Abe, T., Buchanan, E. K., Pleiss, G., Zemel, R. & Cunningham, J. P. Deep Ensembles Work, But Are They Necessary? *arXiv* (2022).
27. Bowden, J. *et al.* Bayesian Optimization with Bayesian Deep Kernel Learning. *In Preparation*.
28. Eriksson, D., Pearce, M., Gardner, J., Turner, R. D. & Poloczek, M. Scalable Global Optimization via Local Bayesian Optimization. *NeurIPS* (2019).
29. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
30. A. Elnaggar *et al.* ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7112–7127 (2022).
31. Coelho, P. S., Brustad, E. M., Kannan, A. & Arnold, F. H. Olefin Cyclopropanation via Carbene Transfer Catalyzed by Engineered Cytochrome P450 Enzymes. *Science* **339**, 307–310 (2013).
32. Pesce, A., Bolognesi, M. & Nardini, M. Protoglobin. in *Advances in Microbial Physiology* vol. 63 79–96 (Elsevier, 2013).
33. Knight, A. M. *et al.* Diverse Engineered Heme Proteins Enable Stereodivergent Cyclopropanation of Unactivated Alkenes. *ACS Cent. Sci.* **4**, 372–377 (2018).
34. Porter, N. J., Danelius, E., Gonen, T. & Arnold, F. H. Biocatalytic Carbene Transfer Using Diazirines. *J. Am. Chem. Soc.* **144**, 8892–8896 (2022).
35. Gao, S. *et al.* Enzymatic Nitrogen Incorporation Using Hydroxylamine. *J. Am. Chem. Soc.* **145**, 20196–20201 (2023).
36. Hanley, D. *et al.* Stereospecific Enzymatic Conversion of Boronic Acids to Amines. *J. Am. Chem. Soc.* (2024).
37. Park, Y., Metzger, B. P. H. & Thornton, J. W. Epistatic drift causes gradual decay of predictability in protein evolution. *Science* **376**, 823–830 (2022).
38. Hopf, T. A. *et al.* Mutation effects predicted from sequence co-variation. *Nat. Biotechnol.* **35**, 128–135 (2017).
39. Long, Y. *et al.* LevSeq: Rapid Generation of Sequence-Function Data for Directed Evolution and Machine Learning. *Under Review*.
40. Wu, N. C., Dai, L., Olson, C. A., Lloyd-Smith, J. O. & Sun, R. Adaptation in protein fitness landscapes is facilitated by indirect paths. *eLife* **5**, e16965 (2016).
41. Johnston, K. E. *et al.* A combinatorially complete epistatic fitness landscape in an enzyme active site. *Proc. Natl. Acad. Sci.* (2024).
42. Rao, R. *et al.* Evaluating Protein Transfer Learning with TAPE. *arXiv* (2019).
43. Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci.* **118**, (2021).
44. Georgiev, A. G. Interpretable Numerical Descriptors of Amino Acid Space. *J. Comput. Biol.* **16**, 703–723 (2009).
45. Letham, B., Karrer, B., Ottoni, G. & Bakshy, E. Constrained Bayesian Optimization with Noisy Experiments. *arXiv* (2018).

46. Luo, Y., Liu, Y. & Peng, J. Calibrated geometric deep learning improves kinase–drug binding predictions. *Nat. Mach. Intell.* (2023).
47. Stanton, S., Maddox, W. & Wilson, A. G. Bayesian Optimization with Conformal Prediction Sets. *PLMR* **206**, 959–986.
48. Fannjiang, C., Bates, S., Angelopoulos, A. N., Listgarten, J. & Jordan, M. I. Conformal prediction under feedback covariate shift for biomolecular design. *Proc. Natl. Acad. Sci.* **119**, e2204569119 (2022).
49. Fannjiang, C. & Listgarten, J. Is Novelty Predictable? *Cold Spring Harb. Perspect. Biol.* (2023).
50. Fröhlich, C. *et al.* Epistasis arises from shifting the rate-limiting step during enzyme evolution of a β-lactamase. *Nat. Catal.* **7**, 499–509 (2024).
51. Hollmann, F., Sanchis Martinez, J. & Reetz, M. T. Learning from Protein Engineering by Deconvolution of Multi-Mutational Variants. *Angew. Chem. Int. Ed.* e202404880 (2024).
52. Wilson, A. G. & Izmailov, P. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. *arXiv* (2022).
53. Wittmann, B. J., Johnston, K. E., Almhjell, P. J. & Arnold, F. H. evSeq: Cost-Effective Amplicon Sequencing of Every Variant in a Protein Library. *ACS Synth. Biol.* **11**, 1313–1324 (2022).
54. Yang, J. *et al.* DeCOIL: Optimization of Degenerate Codon Libraries for Machine Learning-Assisted Protein Engineering. *ACS Synth. Biol.* acssynbio.3c00301 (2023).
55. Nov, Y. When Second Best Is Good Enough: Another Probabilistic Look at Saturation Mutagenesis. *Appl. Environ. Microbiol.* **78**, 258–262 (2012).
56. Gibson, D. G. *et al.* Enzymatic assembly of DNA molecules up to several hundred kilobases. *Nat. Methods* **6**, 343–345 (2009).
57. Balandat, M. *et al.* BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. *arXiv* (2020).
58. Gardner, J. R., Pleiss, G., Bindel, D., Weinberger, K. Q. & Wilson, A. G. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. *arXiv* (2021).
59. Rasmussen, C. E. & Williams, C. K. I. *Gaussian Processes for Machine Learning*. (MIT Press, Cambridge, Mass, 2006).
60. Chen, T. & Guestrin, C. XGBoost: A Scalable Tree Boosting System. in *International Conference on Knowledge Discovery and Data Mining* 785–794 (2016).
61. Desautels, T., Krause, A. & Burdick, J. W. Parallelizing Exploration-Exploitation Tradeoffs in Gaussian Process Bandit Optimization. *JMLR* **15**, 4053–4103 (2014).
62. Rahimi, A. & Recht, B. Random Features for Large-Scale Kernel Machines. *NeurIPS* (2007).

METHODS & PROTOCOLS

Cloning of Random ParPgb Variants

Cloning for Single Site-Saturation Mutagenesis. Chemically competent *Escherichia coli* (*E. coli*) cells (T7 Express Competent *E. coli*) were purchased from New England Biolabs (NEB,

Ipswich, MA). Phusion polymerase and *Dpn*I were purchased from NEB. SSM experiments were performed using primers bearing degenerate codons (NNK) using a modified QuikChange™ protocol.[55] The PCR conditions were (final concentrations): Phusion HF Buffer 1x, 0.2 mM dNTPs each, 0.5 μM of forward primers, 0.5 μM reverse primer, and 0.02 U/μL of Phusion polymerase. The standard Phusion PCR protocol was used. Upon completion of PCRs, the remaining template was digested with *Dpn*I. Gel purification was performed with a Zymoclean Gel DNA Recovery Kit (Zymo Research Corp, Irvine, CA). The purified PCR product was then assembled using the Gibson assembly protocol.[56]

Transformation of Single Site Mutants. 96-well deep-well plates are shaken in an INFORS HT Multitron Shaker in all instances. The assembly products obtained were used to transform T7 Express Competent *E. coli* (High Efficency) cells (NEB, Ipswich, MA) following the recommended protocol. Upon heat-shock transformation, mixtures were recovered in 0.4 mL Luria-Bertani medium (LB) (Research Products Int.), after which the cells were incubated at 37 ºC with shaking at 220 rpm for 30 minutes before being plated on LB-agar plates with 100 μg/mL ampicillin (LB-amp agar plates). Single colonies from LB-agar plates were picked using sterilized toothpicks, which were used to individually inoculate 400 μL of LB containing 100 μg/mL of ampicillin (LB-amp) in 2 mL 96-well deep-well plates. The plates were incubated at 37 ºC and shaken at 220 rpm for 16-18 hours. The following morning 50 μL of preculture from each well were added to the wells of a 96-well flat-bottom tissue culture plate (ThermoFisher) preloaded with 50 μL of 50% glycerol solution. These glycerol stocks were stored at -80°C for future inoculation. Additionally, the sequences of protoglobin genes contained in every well were sequenced using the evSeq protocol.[53]

Cloning for Multisite-Saturation Mutagenesis. Mutations were simultaneously incorporated as with single SSM using the ParLQ_quadNNK primers (**Table S4**). The library transformation was recovered in 0.4 mL LB. 50 µL of transformation mixture were used to inoculate 6 mL of LB-Amp in a 15 mL plastic culture tube. This culture was allowed to shake overnight at 37°C. The following morning, this library preculture was miniprepped using a QIAprep Spin Miniprep Kit (Qiagen, Hilden, Germany). This miniprep sample was used as the new template for mutagenesis with the primers for SSM of site 89. The Gibson products for the new five-site library were transformed using the recommended protocol into T7 Express Competent *E. coli*. Upon heat-shock transformation, mixtures were recovered in 0.4 mL Luria-Bertani medium (LB) (Research Products Int.), after which the cells were incubated at 37 ºC with shaking at 220 rpm for 30 minutes before being plated on LB-agar plates with 100 µg/mL ampicillin (LB-amp agar plates). Single colonies from LB-agar plates were picked using sterilized toothpicks, which were used to individually inoculate 400 µL of LB containing 100 µg/mL of ampicillin (LB-amp) in 2 mL 96-well deep-well plates across 4 separate plates. The plates were incubated at 37 ºC and shaken at 220 rpm for 16-18 hours. The following morning 50 µL of preculture from each well were added to the wells of a 96-well flat-bottom tissue culture plate (ThermoFisher) preloaded with 50 µL of 50% glycerol solution. These glycerol stocks were stored at -80°C for future inoculation. Additionally, the sequences of protoglobin genes contained in every well were sequenced using LevSeq sequencing.[39]

Cloning of ParPgb Predicted Sequences

96-Well Plate Gibson Protocol. Exact genes encoding ParLQ mutants predicted by Active Learning-Assisted Directed Evolution (ALDE) were synthesized and delivered by Elegen Corp.

(San Carlos, CA). DNA fragments were received as dry residues in 96-well PCR plates in 2-4 μg quantities. These DNA samples were dissolved in 100 μL of double-distilled H_2O (ddH_2O), yielding concentrations between 20-40 ng/μL. 0.7 μL of these gene solutions were added to the wells of a 96-well PCR plate (Globe Scientific Inc., Mahwah, NJ). 1.0 μL of an aqueous solution containing 60 ng/μL of linearized pET-22b(+) backbone with overhangs designed for Gibson ligation with the ordered DNA sequences was added to each of the wells of this plate. Finally, to each well was added 5 μL of Gibson assembly mix. The 96-well plate was then incubated at 50°C for 60 minutes, after which the Gibson products were placed on ice. These Gibson products could then either be directly used for transformation or stored at -20°C for later use.

96-Well Plate Transformation Protocol. To each well of the previously described Gibson assembly plate was added 5 μL of T7 Express Competent *E. coli*. The cell solutions were allowed to incubate on ice for 20 minutes, after which they were heat-shocked at 42°C for 10 seconds in a water bath. The cells were then recovered with the addition of 100 μL of LB. Without outgrowth at 37°C, 10 μL of each transformation mixture was used to inoculate the wells of a 2 mL 96-well deep-well plate in which the wells had been preloaded with 400 μL LB-Amp. This plate was incubated at 37 ºC and shaken at 220 rpm for 16-18 hours. The following morning the plate was removed from shaking and allowed to sit at room temperature for 8-10 hours. After this rest phase, 1 μL from each well was used to reinoculated yet another 96-well deep-well plate preloaded with 400 μL LB-Amp. This cell passage plate was incubated at 37 ºC and shaken at 220 rpm for 16-18 hours. The following morning 50 μL of preculture from each well was added to the wells of a 96-well flat-bottom tissue culture plate (ThermoFisher) preloaded with 50 μL of 50% glycerol solution. These glycerol stocks were stored at -80°C for future inoculation.

Protocols for the Screening of ParPgb Variants

96-Well Plate Library Expression. The wells of a 2 mL 96-well deep-well plates were filled with 400 µL LB-Amp. Previously generated 96-well plate glycerol stocks were removed from -80°C storage and placed on dry ice. Multichannel pipet tips were used to scratch the frozen glycerol stock surface and used to inoculate the aforementioned deep-well plate. These pre-expression cultures were incubated at 37 ºC and shaken at 220 rpm for 16-18 hours. For expression cultures, the following morning 50 µL of these precultures were used to inoculate 900 µL of Terrific Broth (TB) (Research Products Int.) with 100 µg/mL of ampicillin (TB-amp) per well in 96-well deep-well plates. These expression cultures were initially incubated at 37 ºC and 220 rpm for 2.5 hours, at which point they were allowed to sit at room temperature for 30 minutes. Expression of proteins was induced with isopropyl-β-D-thiogalactoside (IPTG) and cellular heme production was increased with 5-aminolevulinic acid (ALA). An induction mixture containing IPTG and ALA in TB-amp (50 µL) was added to each well such that the final concentrations of IPTG and ALA were 0.5 mM and 1.0 mM, respectively. The total culture volumes were 1 mL. The plates were then incubated at 22 ºC and 220 rpm overnight.

96-Well Plate Library Reactions and Screening. Expression cultures containing *E. coli* expressing hemoproteins of interest were centrifuged at 4000 × g for 10 minutes at 4 ºC. The supernatant was discarded, and nitrogen-free M9 minimal medium (M9-N, 380 µL) was added to each well. The pellets were resuspended in this medium via shaking at room temperature for 30 minutes. The plates were then transferred into a vinyl Coy anaerobic chamber (0 – 30 ppm O_2). To each well was added 20 µL of a MeCN solution with 200 mM of the desired styrene substrate and 300 mM of ethyl diazoacetate (EDA). The final reaction volume was 400 µL, and the final concentrations of the styrene and EDA were 10 mM and 15 mM, respectively. The plates were

then sealed carefully with a foil cover and shaken at room temperature for 16 hours in the Coy chamber. Once complete, plates were worked up for processing by adding 600 μL of a 1:1 solution of ethyl acetate:cyclohexane containing 1,3,5-trimethoxybenzene as an internal standard (1.0 mM concentration). A silicone sealing mat (AWSM1003S, ArcticWhite) was used to cover the plate and the two layers were thoroughly mixed by rapid inversion of the plate. The plate was then centrifuged (5000 × g for 5 minutes at room temperature) to separate the phases. Afterwards, a 200 μL aliquot of the organic layer was transferred to a GC vial insert in a GC vial, and the samples were analyzed by GC-FID.

Machine Learning Details

The initial training data for the ParPgb campaign was obtained by merging sequencing data with screening yield data. Measured yields were averaged for sequences with the same amino acid combination and normalized to the yield of the *cis* product formation of the parent variants (WYLQF) on each plate. These normalized values were used for model training and acquiring new points, which followed the same protocol as the computational simulations on GB1 and TrpB. For the wet-lab campaign, we trained the model with onehot encodings, the DNN ensemble with 5 models and bootstrapping using 90% of the available training data for each model, and Thompson sampling as the acquisition function. These design choices correspond to the most consistent strategy based on the computational simulations. Detailed instructions on how to reproduce our results and run ALDE for other engineering campaigns are provided at https://github.com/jsunn-y/ALDE.

Most Bayesian optimization algorithms consist of two main components: (1) a probabilistic surrogate model of the objective function and (2) an acquisition function. The surrogate model

predicts the objective function values at unobserved inputs, while the acquisition function quantifies the potential benefit of evaluating any given batch of inputs based on these predictions. In each iteration of the Bayesian optimization loop, a new batch of inputs is selected by maximizing the acquisition function. After evaluating the objective function at these new inputs, the surrogate model is updated, and the process repeats. Below, we describe in detail the probabilistic models and acquisition functions explored in this work, which were implemented using BoTorch[57] and GPyTorch.[58]

Probabilistic Models for Bayesian Optimization

Let $\boldsymbol{X}$ denote the input space (i.e., the space of feasible protein sequences) and let $f: \boldsymbol{X} \to \boldsymbol{R}$ denote the objective function (i.e., the metric we wish to optimize). In this work, we explore four classes of probabilistic surrogate models of the objective function: regular Gaussian processes (GP), deep kernel Gaussian processes (DKL), deep ensembles (DNN ensemble), and boosting ensembles.

Gaussian Processes. A Gaussian process model is defined in terms of a prior mean function $\mu_0: \boldsymbol{X} \to \boldsymbol{R}$ and a prior covariance function $K_0: \boldsymbol{X} \times \boldsymbol{X} \to \boldsymbol{R}$ and it encodes a Bayesian prior distribution over f. Given a dataset of n evaluations of the objective function, denoted as $\mathcal{D}_n = \{(x_i, y_i)\}_{i=1}^n$, one can derive the posterior distribution of f given $\mathcal{D}_n$. If these evaluations are corrupted by i.i.d. additive Gaussian noise, i.e., $y_i = f(x_i) + \epsilon_i$, where $\epsilon_1, \dots, \epsilon_n$ are i.i.d. Gaussian with mean zero and variance σ^2, the posterior is again a Gaussian process characterized by a posterior mean function $\mu_n: \boldsymbol{X} \to \boldsymbol{R}$ and a posterior covariance function $K_n: \boldsymbol{X} \times \boldsymbol{X} \to \boldsymbol{R}$. These functions can be computed in closed form in terms of the prior mean and covariance functions as well as the data using the classical Gaussian process regression formulas. [59] The noise

variance σ^2 and other hyperparameters of the model (such as the lengthscale parameters) can be estimated by maximizing the log marginal likelihood.

Deep Kernel Learning. Gaussian process models with classical covariance functions, such as the Matern or squared exponential covariance functions, are known to perform poorly in high-dimensional input spaces.[28] To address this limitation, Wilson et al. (2015) proposed *deep kernel learning*.[25] Succinctly, this approach uses a covariance function of the form $\mathrm{K}(x, x') = k\big(\phi_w(x), \phi_w(x')\big)$, where k is a regular covariance function (e.g., squared exponential) and ϕ_w is a deep neural network with weights w. These weights are treated like hyperparameters of the model, which can also be estimated by maximizing the log marginal likelihood.

Boosting Ensembles. Boosting models leverage a sequential training strategy where each new model is trained to correct the errors of the previously combined models.[60] The final prediction is often a weighted sum of the predictions made by earlier models, where the weights reflect each model's accuracy. Unlike methods such as bagging, which train models independently and in parallel, boosting specifically designs each new model to address the weaknesses of the existing ensemble, thereby creating a strong predictive model from a sequence of weaker ones. While boosting does not inherently offer a probabilistic interpretation like Bayesian methods, it is highly effective for reducing bias and variance in predictive modeling tasks. Here, we train the boosting ensembles with bootstrapping; each ensemble consists of 5 models where 90% of the total training data is randomly seen during training.

Deep Ensembles. Deep neural network (DNN) ensemble models are constructed by training identical deep neural network architectures multiple times, each with different random initializations of the weight parameters. Here, we train the deep ensembles with bootstrapping; each ensemble consists of 5 models where 90% of the total training data is randomly seen during

training. These independently trained networks are then collectively used as if they were samples from a Bayesian posterior distribution over the objective function f. Unlike Gaussian processes, deep ensembles lack a proper Bayesian interpretation. However, Izmailov and Wilson argue it is possible to see these models as a form of approximate Bayesian inference.[52] We adopt this view in our work.

Acquisition Functions for Bayesian Optimization

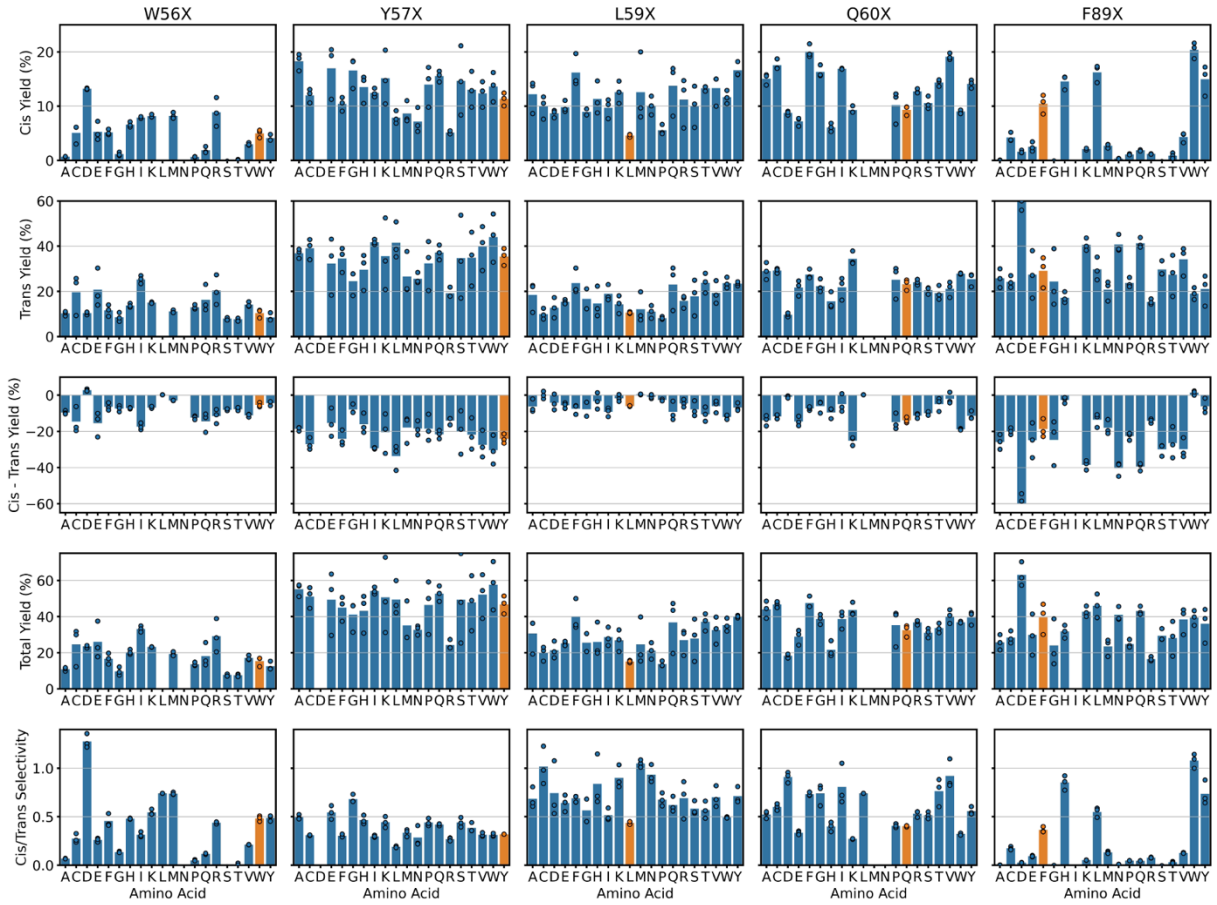
Expected Improvement. The expected improvement (EI) acquisition function is given by $\alpha_n(x) = E_n[\{f(x) - f_n^*\}^+]$, where $f_n^* = \max_{i=1,\dots,n} f(x_i)$ and the expectation is computed with respect to the posterior distribution given $\mathcal{D}_n$.[45] For Gaussian posterior distributions and noise-free observations (where f_n^* is a constant rather than a random variable), the EI can be expressed in a closed form using the posterior mean and variance. In scenarios where these conditions do not hold, computing the EI often requires approximate calculation, typically through Monte Carlo sampling techniques. When extending the EI to the batch setting, the acquisition function becomes $\alpha_n(X) = E_n\left[\{\max_{x \in X} f(x) - f_n^*\}^+\right]$, where $X = (x_1, \dots, x_q) \in \boldsymbol{X}^q$ is a batch of q inputs (qEI). Maximizing the batched EI poses significant computational challenges due to the requirement to optimize over $\boldsymbol{X}^q$. However, by exploiting the submodularity of the acquisition function, an efficient approximation can be achieved through a greedy optimization strategy, selecting each input in the batch sequentially. In this study, we tested qEI, but it ran slowly without noticeable improvement, so it was not included in the final results.

Upper Confidence Bound. The upper confidence bound (UCB) acquisition function is defined by $\alpha_n(x) = \mu_n(x) + \beta_n^{1/2}\sigma_n(x)$, where $\mu_n(x)$ and $\sigma_n(x)$ are the posterior mean and standard deviation, respectively, and β_n is a parameter that controls the exploration-exploitation

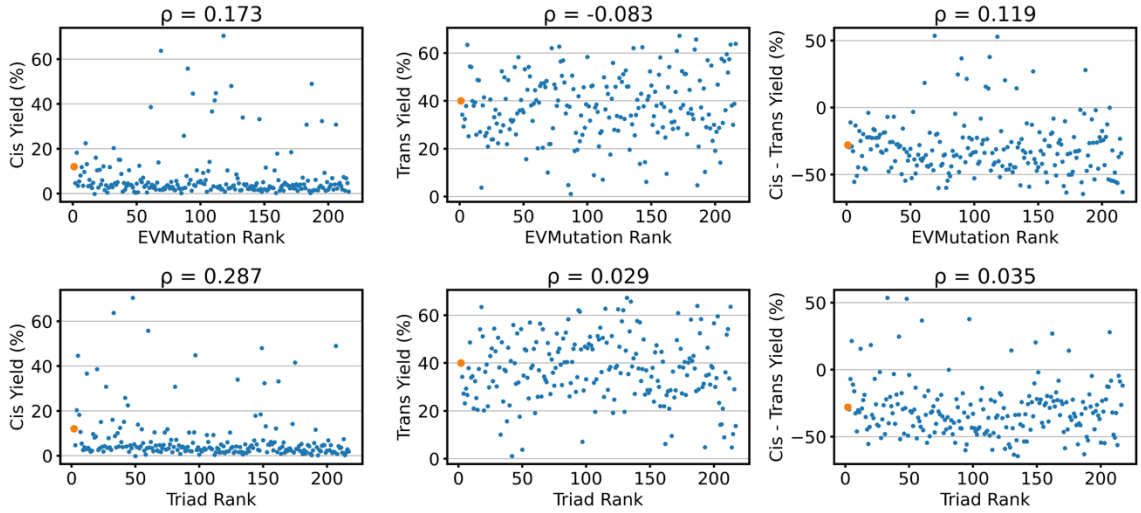
trade-off. In our experiments, we set $\beta_n = 4$. While there are sophisticated batch extensions of the UCB acquisition function available in the literature,[61] our approach utilizes a straightforward heuristic. Specifically, we form batches by selecting the q inputs that yield the highest values of $\alpha_n(x)$, evaluated across all discrete x in the design space. The Greedy acquisition function can be thought of as a specific case of UCB with $\beta_n = 0$ so the acquisition function becomes $\alpha_n(x) = \mu_n(x)$. For the frequentist ensemble models, we evaluate $\mu_n(x)$ and $\sigma_n(x)$ as the mean and standard deviation of all models in the ensemble, respectively.

Thompson Sampling. Thompson Sampling (TS) is a randomized selection strategy where the next input to evaluate is obtained by drawing a sample (function) from the posterior distribution of f and selecting the point that maximizes this sample. For the GP and DKL models, we approximate samples from the posterior using 1000 random Fourier features.[62] For the frequentist ensemble models, the random function sample is drawn as one of the models in the ensemble. In the batch setting, each input in the batch is obtained as an independent sample. Unlike the EI and UCB, TS is inherently stochastic as opposed to deterministic.

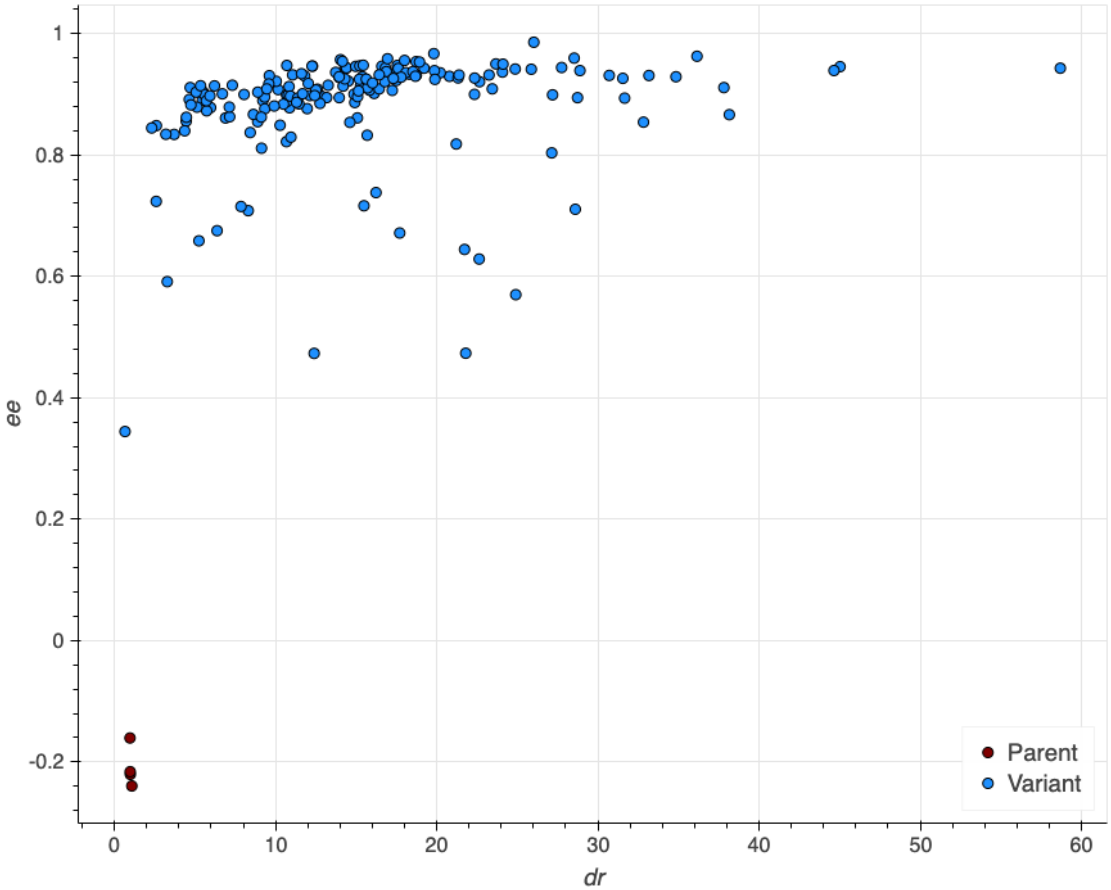
EXTENDED DATA



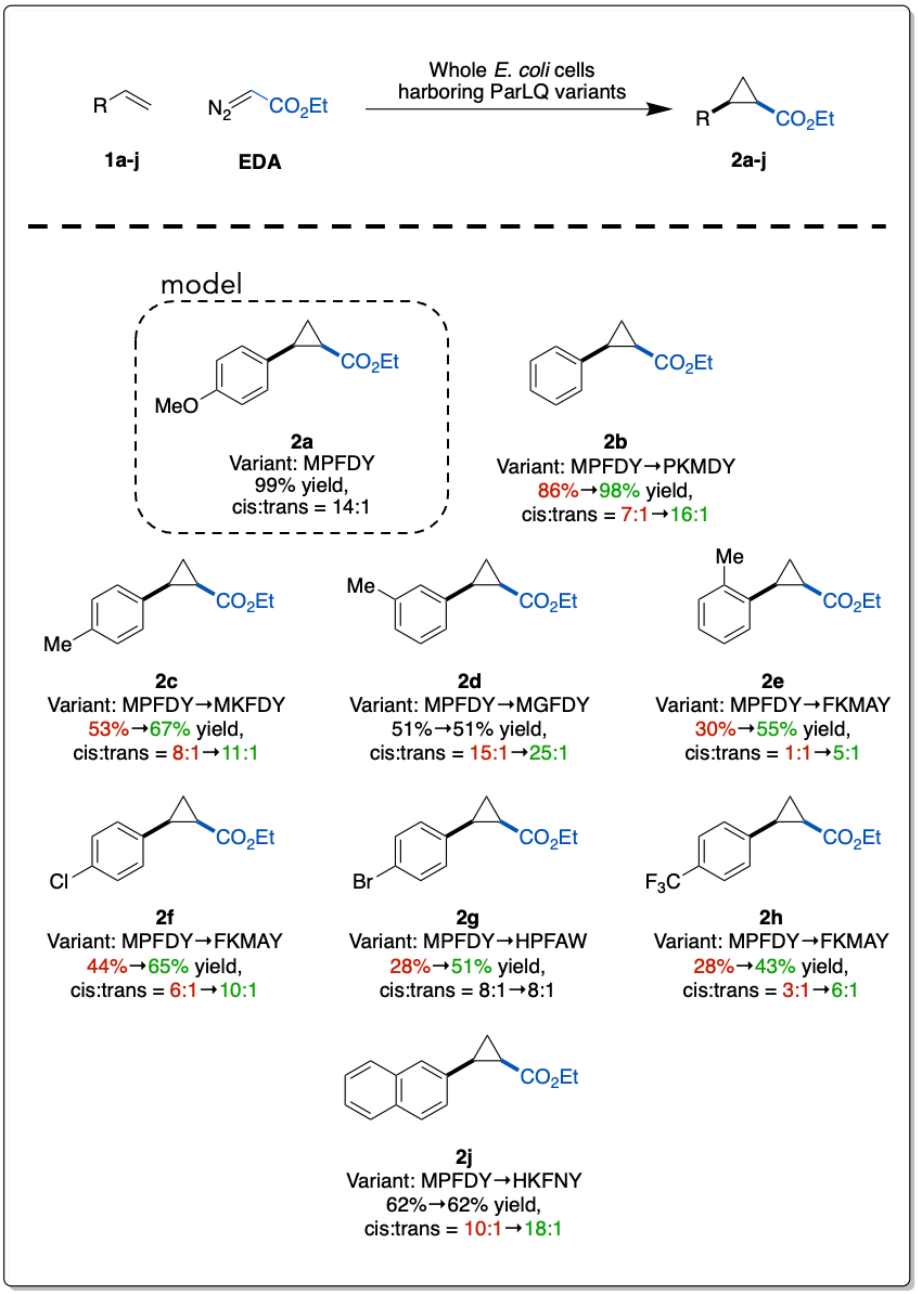
Extended Data Fig. 1. Fitnesses of variants from SSM at the five positions under study, measured by various objectives.



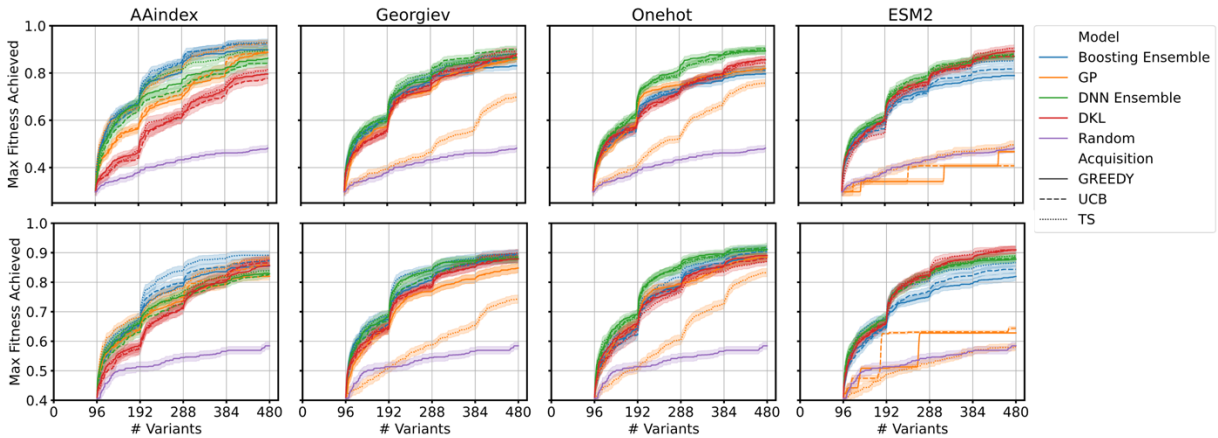
Extended Data Fig. 2. Correlation between zero-shot predictors (EVMutation Rank and Triad Rank) and different fitness metrics (Cis, Trans, and Cis - Trans Yield) for the initial random library of variants used in the *ParPgb* wet-lab campaign. EVMutation rank refers to the evolutionary likelihood of a variant (1 is the most likely), and Triad rank refers to the computationally predicted stability of a variant as a ΔΔG value (1 is the most stable). Orange dot refers to the parent sequence, WYLQF. Each title shows the spearman correlation between the zero-shot predictor and the fitness metric. Cis yield is weakly correlated to the zero-shot predictors, but the overall objective is not.



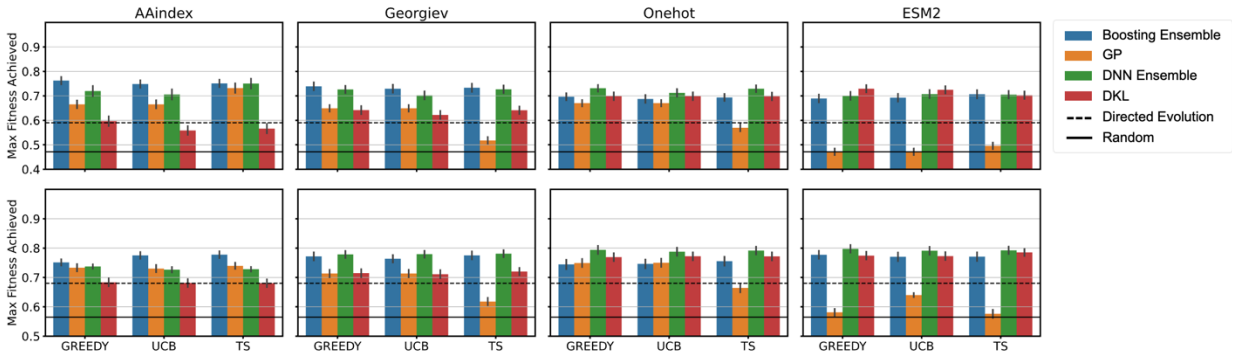
Extended Data Fig 3. Enantioselectivity data for all ALDE predicted variants in Round 1 and Round 2 for the production of **2a**. Enantiomeric excess (ee) values are plotted against diastereomeric ratio (dr) values for each variant.



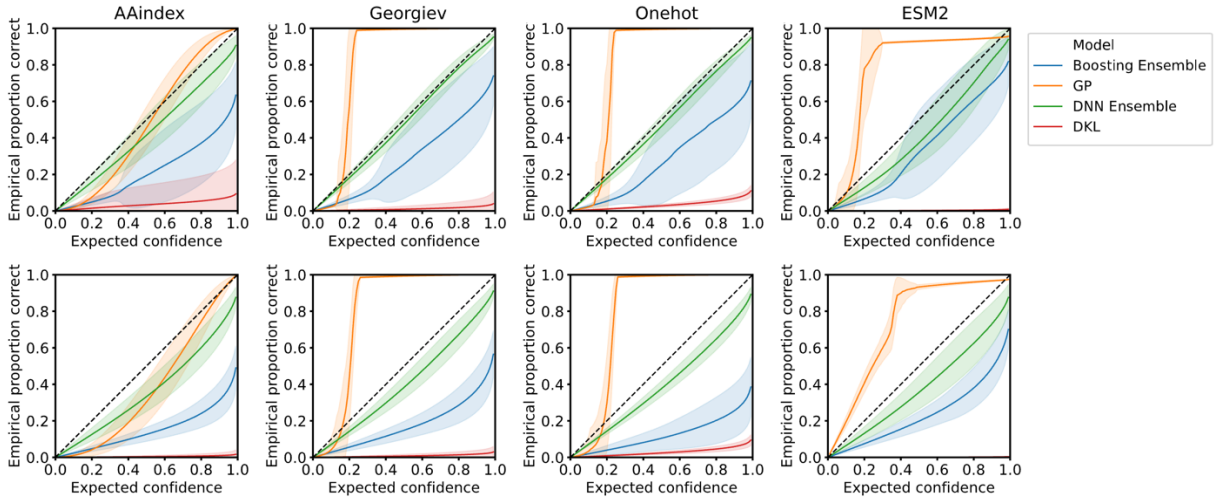
Extended Data Fig. 4. Substrate scope for *Par*Pgb variants explored during Round 2 of the wet-lab ALDE campaign. All substrates were tested on each of the variants. Total yields and selectivities are shown for the top-performing variant for each substrate as well as MPFDY. Improvements in yield and selectivity from MPFDY are indicated with a shift from red to green.



Extended Data Fig. 5. Optimization trajectories for ALDE campaigns for 4 encodings, 4 models, and 3 acquisition functions. Simulation involved batch BO with an initial batch of 96 samples, followed by 4 batches of 96 samples each. Top row is GB1 and bottom row is TrpB. Error bars indicate standard deviation across 70 random initializations.



Extended Data Fig. 6. Performance of MLDE baseline for 4 encodings, 4 models, and 3 acquisition functions, compared to the average DE simulation and to random sampling. Performance is quantified as the normalized maximum fitness achieved by MLDE, where the training set is 384 random samples and the test set in 96 samples proposed by the model using a greedy acquisition function. Top row is GB1 and bottom row is TrpB. Error bars indicate standard deviation across 70 random initializations.



Extended Data Fig. 7. Calibration curves for 4 encodings and 4 models. The x axis is the expected confidence from the posterior, given a certain confidence interval and the y value is the actual proportion of true labels that fall within the confidence interval. Calibration curve is evaluated across all sequences in the design space with labels, on models trained on the final batch (384 train samples) from ALDE campaigns using UCB. Top row is GB1 and bottom row is TrpB.