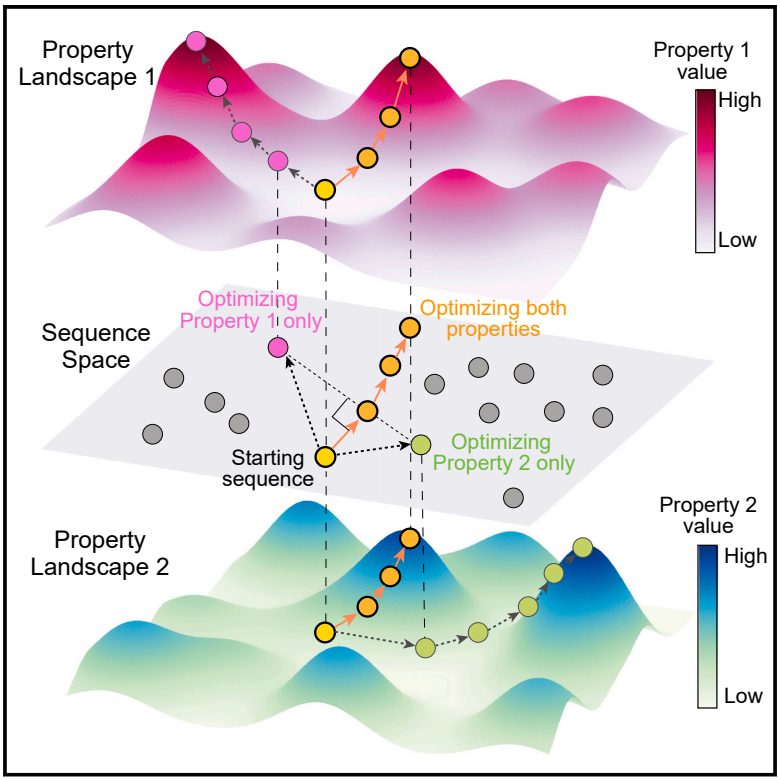
iScience

Article

Pareto-optimal sampling for multi-objective protein sequence design

Graphical abstract



Highlights

- A generative AI model for multi-objective protein sequence design
- Property-guided protein sequence design with Pareto optimization
- Construction of benchmark datasets for multi-property protein sequence design
- Designed sequences optimally trade off multiple properties while maintaining diversity

Authors

Jiaqi Luo, Kerr Ding, Yunan Luo

Correspondence

yunan@gatech.edu

In brief

Artificial intelligence; Biochemistry; Biocomputational method; Structural biology

Luo et al., 2025, iScience *28*, 112119
March 21, 2025
https://doi.org/10.1016/j.isci.2025.112119

CellPress

iScience

CellPress
OPEN ACCESS

Article

Pareto-optimal sampling for multi-objective protein sequence design

Jiaqi Luo,[1] Kerr Ding,[1] and Yunan Luo[1,2,*]
[1]School of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30308, USA
[2]Lead contact
*Correspondence: yunan@gatech.edu
https://doi.org/10.1016/j.isci.2025.112119

SUMMARY

Supervised machine learning (ML) has significantly advanced sequence-based protein property prediction. However, its inverse application, designing protein sequences with desired properties, remains under-explored. The challenges in sequence design stem from the vast search space and the rugged protein fitness landscape. In this work, we present MosPro, an efficient ML algorithm for property-guided protein sequence design. We frame sequence design as a discrete sampling problem. Utilizing a pre-trained differentiable ML model that predicts properties of sequences, MosPro shapes a distribution that assigns high probability mass to regions for high-property sequences. To generate designs, MosPro efficiently samples sequences from this constructed distribution. We further develop a Pareto optimization algorithm to propose sequences that are simultaneously optimized for multiple properties. Evaluations on experimental fitness landscapes demonstrated that MosPro generates sequences that optimally trade off multiple desiderata. Our results suggested an unparalleled potential of generative ML for efficient and controllable design for functional proteins.

INTRODUCTION

Machine learning (ML) has significantly advanced the prediction of protein properties from sequences, driven by the sophisticated predictive capabilities of modern ML models and the abundant sequence-property data generated by high-throughput experiments. In this context, "property" refers broadly to any quantitative protein functionalities, such as binding affinity, stability, or catalytic activity. The core computational problem here is to train an ML model f_θ, parameterized by θ, to predict a protein's property score $y \in \mathbb{R}$ from its sequence $x \in \Sigma^L$, with Σ denoting the set of amino acids and L the sequence length.

While ML methods have been widely used for predicting protein properties from sequences, the inverse problem – designing protein sequences that manifest desired properties – is under-explored compared to its forward counterpart, despite its significance in biomedical applications. This inverse design poses a challenging optimization problem: $x^* \leftarrow \text{argmax}_{x \in \Sigma^L} f^*(x)$, where $f^*(\cdot)$ is an unknown function mapping a protein's sequence to its true property. In the laboratory, this problem is akin to protein engineering that seeks to design or discover proteins with useful properties. This goal is extremely challenging due to the vast sequence space (20^L possible sequences for a length-L protein), rendering exhaustive search impractical, especially considering the majority of this space comprises non-functional proteins. To overcome these obstacles, experimental techniques such as directed evolution perform iterative rounds of random mutagenesis and recombination to identify sequences with enhanced properties. Although effective and recognized by the Nobel Prize,[1] the efficiency of directed evolution is still a bottleneck due to the significant experimental burden.

To accelerate lab-based protein engineering, ML methods have been employed to more efficiently explore the sequence space, proposing sequences likely to exhibit improved functions. Generative ML models, such as variational autoencoders,[2] generative adversarial networks,[3] or protein language models,[4] are typically trained on sequences of naturally occurring proteins to learn to assign every possible sequence with a probability, reflecting their "evolutionary plausibility." The learned probability distribution is then used to sample sequences (e.g., by MCMC sampling[5,6]), which are subsequently assessed by an ML property predictor $f_\theta(x)$ that serves as an effective surrogate for experimental property measurement $f^*(x)$. A limitation of these methods, however, is that they only assess the properties of designed sequences *post hoc*, rather than explicitly optimizing them, resulting in a relatively low success rate in identifying proteins with enhanced properties.

Recent efforts have integrated property predictors with generative ML models to target the generation of sequences with enhanced properties, shifting from random exploration to focused improvement within the vast sequence space. The key idea behind these approaches is to learn a distribution that assigns a high likelihood to sequences with both high evolutionary plausibility and enhanced property. Common strategies include: i) implicitly tweaking the generative model, such as through latent space optimization[7] or sequence density reweighting,[8] to increase the likelihood of producing sequences with desirable properties; or ii) explicitly

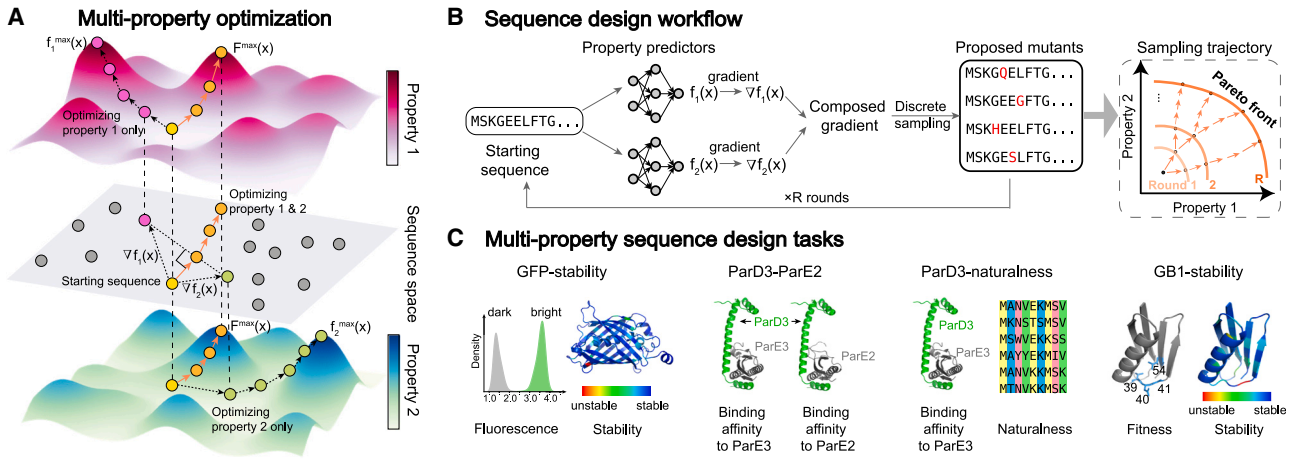


Figure 1. Schematic overview of MosPro, a deep learning-guided algorithm for multi-objective protein sequence design

(A) Schematic of the multi-objective optimization of two properties in sequence design. The central plane is the sequence space with each coordinate representing a protein sequence. The top surface depicts the sequence-property landscape for Property 1, with a peak at $f_1^{\max}(x)$ indicating the maximum level of the property. The bottom surface outlines the landscape of another property with a peak at $f_2^{\max}(x)$. MosPro identifies the optimal direction of sequence search by combining the gradient information on both landscapes, leading to the optimal sequence $F^{\max}(x)$ that enhances both properties.

(B) The sequence design workflow of MosPro. MosPro composes the gradients of two deep learning-based property predictors and utilizes a gradient-guided discrete sampling algorithm to propose mutants. The proposed mutants are then used as the starting sequences in the next round. This gradient-guided sampling process is repeated for multiple rounds, with the final designed sequences reaching the Pareto front that improves both properties without compromising each.

(C) The multi-property sequence design tasks. Four multi-property sequence design tasks are constructed to evaluate MosPro, including GFP-stability, ParD3-ParE2, ParD3-naturalness, and GB1-stability, each with two properties to optimize.

introducing and accumulating beneficial mutations to the wild-type sequence in a hill-climbing manner on the sequence-property landscape.[9–11]

Despite these advancements, most existing property-guided sequence design methods focus only on single-property optimization. However, simultaneously optimizing multiple properties is crucial in many biomedical applications. For instance, in antibody design, alongside the primary functionality (e.g., viral binding affinity), other properties, such as thermostability, solubility, or expression, are also critical for creating functional and developable designs. A key challenge in multi-property optimization is the potential conflict between objectives; for example, solely enhancing the specificity or solubility of antibodies might compromise their affinity.[12–14] Although recent studies have started exploring multi-property sequence design,[15–17] the development of ML algorithms that effectively balance multiple protein properties remains largely under-explored.

In this work, we present MosPro (multi-objective sequence design for proteins), a deep learning-guided framework for multi-objective protein sequence design. MosPro designs property-enhanced sequences by leveraging the gradient information from a differentiable ML property predictor to inform a discrete sampling algorithm for searching and accumulating beneficial mutations from a starting sequence for property enhancement. Unlike random or localized search methods used in experimental directed evolution, MosPro's ML-guided approach leads to a more effective and broader exploration of the sequence-property landscape. To simultaneously optimize multiple, potentially conflicting properties, MosPro navigates the parameter spaces of the property-specific predictors and identifies a parameter update direction that maximizes improvement across all properties. This process results in Pareto-optimal designs that represent the best trade-off across multiple objectives, where no single objective could be further improved without sacrificing others. To evaluate MosPro, we created a benchmark of four multi-objective sequence design tasks, covering various properties such as binding affinity, fluorescence, and stability. MosPro demonstrated an outstanding ability to discover sequences that balance multiple objectives more effectively than existing single or multi-objective optimization methods. These results suggested an innovative paradigm in multi-objective protein sequence design, with significant implications for protein engineering.

RESULTS

MosPro: Multi-property sequence design with Pareto-optimal sampling

MosPro is a deep learning-guided algorithm for multi-objective protein sequence design (Figure 1). MosPro first trains multiple property-specific supervised ML models to predict a protein's properties of interest from its sequence. To design proteins with multiple desired properties, MosPro starts from a set of initial sequences and iteratively optimizes the sequences so that the multiple properties are jointly optimized. In each iteration, MosPro employs a discrete sampling algorithm,[18] guided by the trained property predictors, to propose beneficial mutations that are most likely to enhance the properties (STAR Methods).

MosPro simultaneously optimizes multiple properties by finding the Pareto-optimal solutions in the sequence space (Figure 1A). Pareto optimality is a common concept in multi-objective

optimization and, in our context, refers to sequences for which no single property can be further optimized without compromising other properties. Unlike previous studies that commonly reduce the multi-property protein sequence design to single-objective optimization by linearly combining the objectives with fixed weights,[11] MosPro design sequences toward the multi-property Pareto optimality with adaptive weights guided by the property predictors. In each round of the sequence design, MosPro applies a Pareto optimal gradient composition scheme, inspired by previous studies,[15,18,19] to identify a sequence update direction along which all properties can be mostly improved (STAR Methods). Repeating this sampling process accumulates beneficial mutations to the starting sequences, akin to a hill-climbing process on the sequence-property landscape, ultimately generating sequences for which multiple properties are optimized (Figure 1B).

Benchmark to evaluate multi-property sequence design

Most deep mutational scanning studies primarily focus on a single property of interest in each experiment,[20] and datasets with multi-property-labeled protein sequences are rarely available. To evaluate MosPro's capability in sequence design, we constructed a benchmark dataset for multi-property sequence design by leveraging experimentally characterized protein fitness landscapes.[21–23] In this dataset, we paired each variant sequence's experimentally assayed property (e.g., binding affinity or fluorescence) with another property, such as stability or naturalness, computed using a biophysical model or protein language model (STAR Methods). Protein mutant stability is quantified by the change in Gibbs free energy relative to the wild-type protein ($\Delta\Delta G$) using FoldX,[24] where a lower $\Delta\Delta G$ indicates a more stable mutant. The naturalness of a protein sequence measures its resemblance to naturally evolved proteins, which is crucial since natural-like proteins are more likely to satisfy evolutionary or structural constraints and avoid undesired functional consequences. For example, it was found that the designed antibodies with high naturalness scores are less likely to trigger unwanted immune responses.[25]

In total, we constructed four multi-property sequence design tasks within this benchmark dataset, where each protein variant sequence is labeled with two properties (Figure 1C). In the evaluation, sequence design methods are tasked with designing sequences that optimize both properties. The first design task in our benchmark is to design green fluorescent protein (GFP) variants with improved fluorescence and stability. We obtained the fluorescence brightness values of $51,715$ GFP mutants from a previous study[21] and additionally labeled each sequence with changes in structural stability due to mutations ($\Delta\Delta G$) as quantified by FoldX, resulting in the task denoted as "GFP-stability."

The second and third tasks focus on the ParD3 protein, a bacterial antitoxin part of the ParD-ParE family of toxin-antitoxin systems, which are often found on bacterial chromosomes. In this system, the toxin can inhibit cell growth or viability unless antagonized by a cognate antitoxin. A prior study[23] experimentally assayed the binding affinities to both ParE3 and ParE2 of all $20^3 = 8,000$ amino acid sequence variants at three sites in ParD3, forming a design task that requires the optimization of both binding affinities, denoted as "ParD3-ParE2." We additionally labeled each ParD3 variant sequence with a naturalness score, resulting in another design task denoted as "ParD3-naturalness."

The fourth task involves GB1, a Streptococcal protein that binds immunoglobulin. The binding affinity of GB1 to immunoglobulin was experimentally assayed for 149,361 variants at four mutation sites in a previous study,[22] in which the "fitness" was defined based on GB1 variants' binding affinity and foldability. This fitness is used as the first property in this task. We additionally computed the $\Delta\Delta G$ stability scores using FoldX as the second property, forming the "GB1-stability" design task.

Our benchmark dataset represents a useful resource for this work and future studies to evaluate how well MosPro and other methods optimize multiple properties in protein sequence design, as we demonstrate in the following sections.

MosPro offers Pareto-optimal multi-property protein sequence design

Using our benchmark dataset, we compared MosPro to several strong baseline methods for property-guided protein sequence design, including GGS,[10] the state-of-the-art discrete sequence design method for single-property optimization; linear scalarization,[26,27] which reduces the two property objectives f_1 and f_2 into a single objective $f = \lambda f_1 + (1 - \lambda) f_2$, with $\lambda \in [0, 1]$ an interpolation hyperparameter; and NSGA-2,[17,28] a genetic algorithm for multi-objective optimization based on non-dominated sorting. The details and settings of these baseline methods can be found in STAR Methods. The designed sequences were evaluated using a set of evaluation oracles depending on the respective property, including ground-truth experimental data (for ParD3 bindings and GB1 fitness), independent computational models (for stability), and ML predictors (for GFP fluorescence). Note that for GFP, while the ML-based evaluation oracle was trained on all available property data, the training data of MosPro's property predictor was restricted to variants with the lowest 40% property values. This setup assesses MosPro ability to evolve low-property variants to high-property ones, simulating the scenario in real-world protein engineering scenarios. The constrained training data also prevents MosPro from memorizing the sequence of high-performing variants and avoids potential circularity issues or data leakage.

In all four design tasks, MosPro offered significant property enhancements (Figure 2). Specifically, MosPro optimally balanced the two objectives and outperformed baseline methods in both or at least one property objective. The effectiveness of MosPro's multi-objective optimization was clearly demonstrated when compared to the linear scalarization method, which interpolates both properties into a single scalar. Ideally, by varying the interpolation weight λ from 0 to 1, linear scalarization should traverse every tradeoff between the two property objectives, shifting from one extreme that emphasizes one property to the other extreme that promotes the second property. However, in our evaluation, linear scalarization failed to exhibit the expected tradeoff trajectory between the two objectives, especially in the ParD3-naturalness and GB1-stability tasks (Figures 2C and 2D). This result highlights the intrinsic challenge of multi-objective optimization and shows that simply combining the two properties in the objective space is not sufficiently informative for guiding multi-property sequence design.

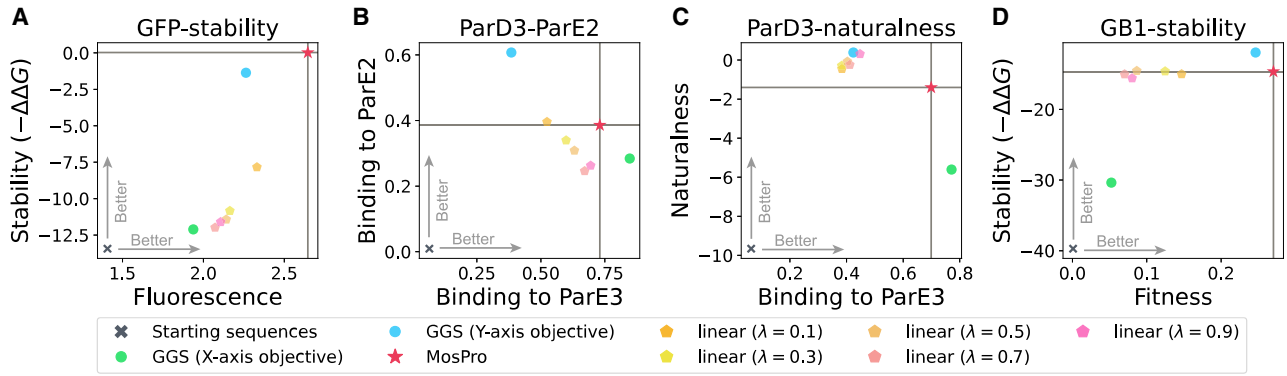


Figure 2. Evaluation of sequences designed by MosPro and baselines

MosPro was compared to baseline methods for multi-property protein sequence design across four design tasks (A–D). Each dot represents the property score averaged over 500 sequences designed by MosPro, GGS, and the linear scalarization method with various interpolation weights. As a reference, a dot corresponding to the starting sequences is also shown to indicate the initial property levels before the optimization in each task. Dashed lines anchored at the point of MosPro serve as a reference for better comparisons with baseline methods. For visualization consistency, the stability metric shown in (A) and (D) was defined as $-\Delta\Delta G$, with a higher value indicating more stable structures. Thus, the desired direction for all properties across all tasks is maximization. Data are represented as mean property values on 500 sequences designed by MosPro.

In contrast, MosPro operates in the sequence space to identify the optimal direction of sequence update such that both properties could be jointly enhanced.

Another advantage of MosPro over linear scalarization is that it does not require a manually specified weight to balance the properties. In linear scalarization, the optimal weight λ is often unknown in practice for an unseen task and sensitive to various aspects of the properties, such as their scale or rate of change with respect to sequence mutation, making a manually selected value potentially suboptimal. MosPro avoids reliance on manually specified hyperparameters by solving a min-max optimization problem (STAR Methods), allowing both properties to be enhanced to the fullest extent. The evaluation results across the four tasks suggested that MosPro's strategy is more effective for multi-property sequence design than the simple linear scalarization approach (Figure 2).

We also compared MosPro to GGS,[10] the state-of-the-art single-property sequence design method based on a discrete sampling algorithm.[18] Since GGS focuses on single-objective optimization, in each design task we trained separate GGS models for the two properties. Although GGS effectively optimized the target property, this often came at the expense of a significantly lower level of the other property (Figure 2). In contrast, MosPro simultaneously improved both properties without compromising either, underscoring the necessity of joint optimization to identify Pareto-optimal solutions for multi-property sequence design. In some tasks, such as GFP-stability, MosPro even outperformed GGS in both objectives, despite GGS's designs being specifically optimized for the respective individual property (Figure 2A).

Additionally, we compared MosPro to NSGA-2,[28] a sequence design method that explicitly optimizes multiple properties using a genetic algorithm. MosPro provided better-improved designs for most cases across various properties and proteins (Figure S2). These performance improvements stem from MosPro's capability to broadly explore the sequence space and identify mutations that enhance multiple properties. In contrast, NSGA-2 relies on random mutations or recombination of previously observed mutations, which may limit its ability to identify property-enhancing mutations. Overall, these results demonstrate that MosPro can simultaneously optimize multiple properties in protein sequence design and achieve Pareto-optimal tradeoffs, outperforming existing sequence design methods that optimize multiple properties jointly or individual properties separately.

MosPro effectively enhances multiple properties

Having established MosPro's superior ability to strike the optimal better balance between two properties than existing methods, we further investigated its effectiveness in evolving protein sequences to enhance their properties. We visualized the property distributions of the MosPro's randomly sampled starting sequences and its final designed sequences in each task (Figures 3A–3D). These starting sequences were intentionally chosen from the low-score region (lowest 40% percentile) to assess MosPro's ability to evolve them into high-score regions (STAR Methods). The resulting distributions illustrate that MosPro effectively evolved the starting sequences (Figure 3, blue dots) into sequences with clearly improved properties (Figure 3, red dots). By overlaying MosPro's designs with the experimentally assayed property landscapes (Figure 3, gray dots), we found that many MosPro designs were enriched in high-property regions of the landscape or even close to the Pareto front corresponding to best-performing variants for both properties observed in the experimental landscapes. We note that the property predictor used by MosPro to guide sequence design was trained only on low-property variants, indicating that high-property variants designed by MosPro are not a result of memorization training data but are attributed to its strong generalizability in exploring unseen regions of the landscapes.

By connecting each starting sequence with its corresponding designed sequences, we observed clear improvement trajectories for both properties (Figures 3E–3H). This improvement was particularly pronounced in the GFP-stability task (Figures 3A and 3E).

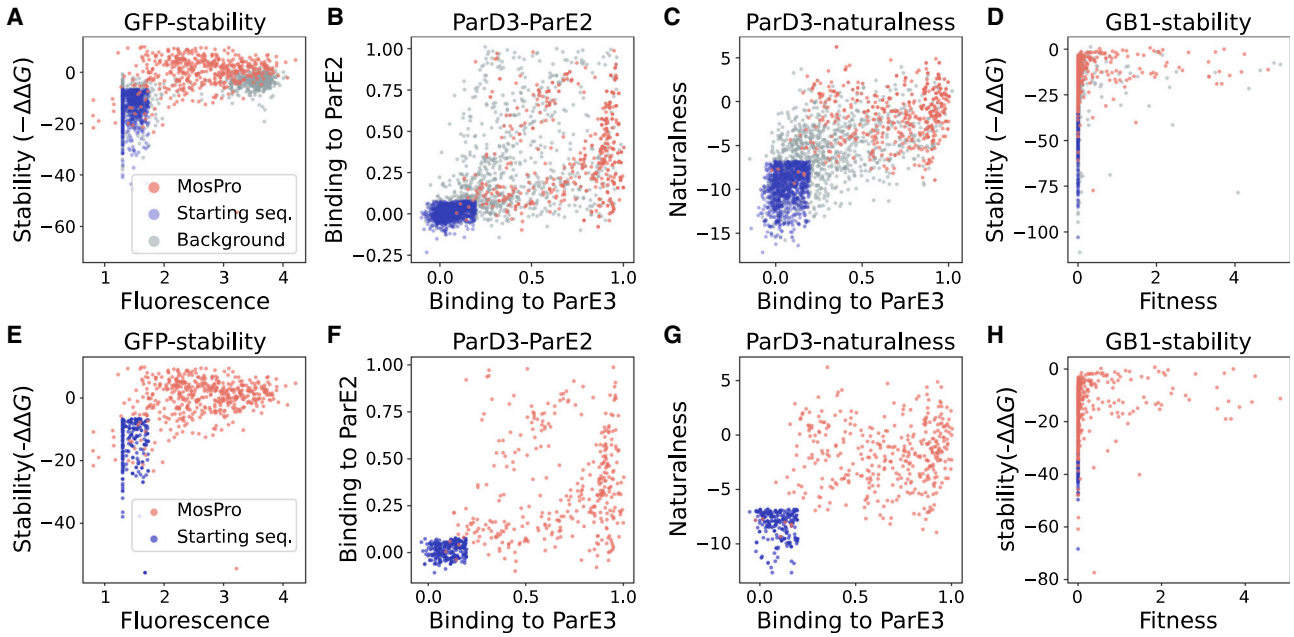


Figure 3. Property distributions of sequences designed by MosPro

(A–D) Property distribution of 500 MosPro-designed sequences, 1,000 starting sequences, and 1,000 background sequences. The background sequences are randomly sampled from the variants experimentally assayed in the empirical landscapes that constitute our benchmark dataset. Each dot indicates the property values of a sequence.

(E–H) Property improvement trajectories between starting sequences and corresponding MosPro designs. The dots in (E–H) are the same as in (A–D), except that starting sequences rejected by MosPro as non-evolvable sequences (STAR Methods) were dropped. For visualization consistency, the stability metric shown in (A) and (E) was defined as $-\Delta\Delta G$, with a higher value indicating more stable structures.

The fluorescence landscape of GFP variants exhibits a bimodal distribution,[21] with MosPro's starting sequences residing in the low-fluorescence mode (Figure 3A). However, MosPro's designed variants generalized across the boundary of modes, enriching in the high-fluorescence region. Further investigation of property levels at each iteration during the optimization process revealed that MosPro efficiently evolved the starting sequences to exhibit nearly 2-fold enhancement in fluorescent brightness (Figure 4A). In addition, MosPro gradually stabilized the variant structure during the evolving process, refining randomly sampled starting sequences with poor structural stability into variants that maintained the structure stability comparable to the wild type, as quantified by the FoldX $\Delta\Delta G$ score (Figure 4A). These results suggest MosPro's strong ability to design GFP variants with enhanced fluorescence and stability.

In the two antitoxin ParD3 tasks, MosPro's designed sequences consistently approached or reached the Pareto front with a high probability (Figures 3B, 3C, and 3F–3H). MosPro's sequence optimization started from the randomly sampled variants located in the low-score region for both properties (Figures 3B and 3C, lower left) and evolved these variants toward the high-score region, with a vast majority rapidly reaching extreme levels of both properties (Figures 3B and 3C, upper right) within three or four iterations of optimization (Figure 4B).

For the GB1-stability task, it should be acknowledged that the specific distribution characteristics of the GB1 fitness property present a significant challenge for optimization. Among the 160,000 sequences in the GB1 dataset, about 95% exhibit extremely low fitness (0.35) (Figure 3D, gray dots). This biased distribution makes fitness optimization challenging due to the extreme scarcity of high-fitness sequences. Despite this, MosPro effectively enhanced both fitness and stability for GB1 (Figures 3D and 3H). The stability of the designed sequences showed significant improvements compared to the starting sequences. Although many designed sequences remain in the low-fitness region, a considerable number successfully reached the sparsely populated high-fitness region and approached the Pareto front (Figure 3D).

MosPro designs variant sequences through an iterative mutating procedure (STAR Methods). In each iteration, MosPro accumulates beneficial mutations for both properties and gradually evolves the variants to reach the Pareto front (Figure 1B). We observed that MosPro achieved steady improvements in property values over iterations (Figure 4) and delivered variants that were up to 15 mutations away from the starting sequences, with enhanced properties. Together, these results suggest that MosPro can effectively identify mutation trajectories in the sequence space to jointly enhance multiple properties of the proteins.

MosPro designs structurally viable sequences

To further validate the quality of the sequences designed by MosPro, we selected the Pareto-optimal sequences it generated and employed AlphaFold2[29] to predict and evaluate their 3D

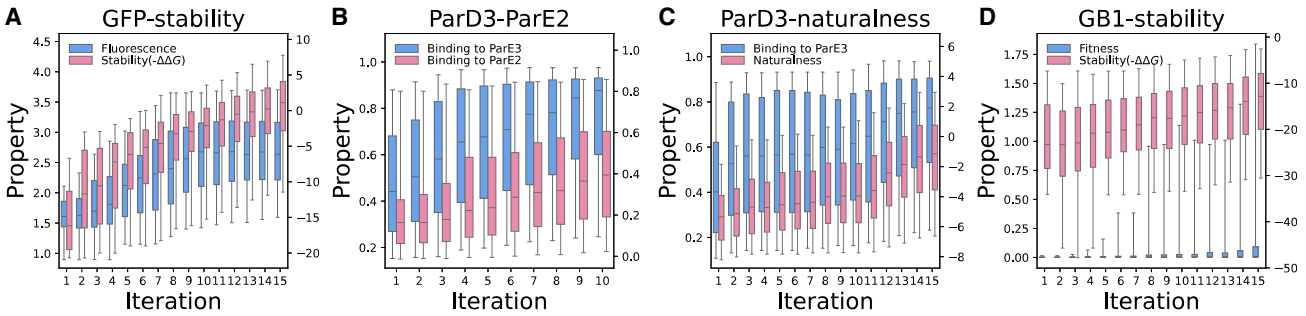


Figure 4. Cumulative property improvements over iterative design

MosPro design sequences by iteratively introducing beneficial mutations into the starting sequence. The box plots show the property improvement over iterations in the four design tasks, including GFP-stability (A), ParD3-ParE2 (B), ParD3-naturalness (C), and GB1-stability (D). The bar at the *i*-th iteration represents the three quartile property values of the sequences designed by MosPro up to the *i*-th iteration. The whiskers extend to property values ranging from the 5th to 95th percentile. The two sets of bars with different colors correspond to the two different properties in each design task. In the ParD3-ParE2 task, only 10 optimization iterations were performed, as the ParD3 protein has only three mutation sites and the optimized property values converged around 10 iterations. For visualization consistency, the stability metric shown in (A) and (D) was defined as $-\Delta\Delta G$, with a higher value indicating more stable structures.

structures. Specifically, we evaluated the sequences designed by MosPro on the GFP-stability task as a case study. AlphaFold2 produces per-residue scores called predicted Local Distance Difference Test (pLDDT), ranging from 0 to 100, to quantify its local confidence, with higher scores indicating higher confidence and usually more accurate predictions.[29] We averaged the per-residue pLDDT scores over the mutation sites as the proxy metric of the foldability of the designed proteins, following previous studies.[30,31]

Figures S3A–S3C show the AlphaFold2 structures of two GFP variants randomly sampled from MosPro's Pareto optimal designs and the wild-type GFP. The two MosPro variants exhibited high pLDDT scores across most residues, with a mean pLDDT above 95. For reference, pLDDT scores higher than 90 are considered the highest accuracy category,[29] suggesting an excellent structural validity for our designs. When compared to the wild-type structures, the two designed variants display comparable pLDDT scores across the entire sequence while achieving improved fitness (Figures S3A–S3C).

As a global evaluation, we compared the pLDDT scores of the mutation sites in MosPro-designed Pareto-optimal sequences against a set of reference sequences, which were randomly sampled from the experimentally assayed variants,[21] serving as a positive control that represents folded sequences. We found that the distributions of pLDDT scores at the mutation sites of MosPro's designed variants closely resembled those of the reference set of folded sequences (Figure S3D), demonstrating MosPro's effectiveness in generating structurally foldable protein variants.

DISCUSSION

Designing sequences with enhanced properties is a fundamental challenge in protein engineering, primarily due to the vast search space and the complex relationship between protein sequences and their properties. ML-based protein sequence design has gained significant attention because it can efficiently explore the sequence space and find the most promising sequence candidates for experimental validation. In this approach, surrogate models that predict the sequence-property relationship are commonly used to guide the design of property-enhanced sequences. For example, techniques such as genetic algorithms and reinforcement learning[5,9,32] have been employed to introduce mutations and used the surrogate models to screen desired mutants, evolving sequences toward higher properties. Another line of work has focused on controllable sequence design for specific properties by training generative ML models using high-property sequences[4,33,34] or by optimizing within latent spaces.[7,35] We compared the key differences of MosPro with previous methods in Table S4. While these methods were successful, they primarily focused on optimizing a single property. However, real-world protein design tasks often require the simultaneous optimization of multiple properties.

In this work, we presented MosPro, a Pareto-optimal sampling framework for multi-property protein sequence design. MosPro trains a separate sequence-to-property predictor for each property. Unlike the commonly used linear scalarization approach, which employs a fixed set of weights to combine multiple properties into a single objective,[11] MosPro operates directly in the protein sequence space, identifying the optimal sequence mutations that can jointly enhance multiple properties. By iteratively accumulating beneficial mutations, MosPro ultimately designs sequences that simultaneously enhance multiple properties. Our evaluation of MosPro on several multi-property protein sequence design tasks demonstrated its capability for multi-property sequence design. Moreover, MosPro is a versatile and practical framework for multi-objective protein sequence design. By accommodating any two or more target properties of interest, MosPro can train the corresponding sequence-property predictors using available data of functionally characterized variants and subsequently perform Pareto-optimal sampling to generate protein sequences that optimally balance these objectives. We anticipate that MosPro will serve as a useful tool to accelerate multi-objective protein design in wet labs when target properties exhibit strong trade-offs and when optimizing one property may be detrimental to others. MosPro holds significant potential for many drug design and protein engineering applications, such as optimizing both specificity and binding affinity in antibody design

or enhancing structural stability without compromising enzymatic performance in enzyme engineering.

Limitations of the study

Despite MosPro's effectiveness in designing high-performance protein sequences, some limitations remain. As with prior studies,[8–10,35] MosPro's performance is influenced by the accuracy of the sequence-to-property predictor that guides the sequence design process. Although in the tasks explored in this work, our property predictors demonstrated high prediction accuracy (> 0.9 Pearson correlation on the validation set), accurately modeling protein sequence-property relationships remains an open challenge, particularly when the property landscape is highly complex or when the experimentally measured data is insufficient for effective ML training. Addressing these challenges may require generating and curating larger, high-quality datasets of functionally characterized protein variants[20] or developing data-efficient ML models tailored for low-sample settings.[5,36,37] Furthermore, the discrete sampling algorithm employed in MosPro, Gibbs With Gradient,[18] performs better on smooth landscapes, where sequences with smaller edit distances exhibit smaller differences in properties. However, protein property landscapes are often highly rugged. MosPro currently trains predictors directly on these rugged landscapes, which can pose challenges to prediction accuracy. To address this, future improvements could involve constructing smoother approximations of these landscapes[10] or applying smoothing techniques to predictor parameters,[38] enabling the sampling algorithm to better navigate these complex landscapes.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Yunan Luo (yunan@gatech.edu).

Materials availability

This study did not generate new unique reagents.

Data and code availability

- The multi-property protein sequence data has been deposited at Zenodo and is publicly available as of the date of publication. The DOI is listed in the key resources table.
- All original code has been deposited at GitHub and is publicly available as of the date of publication. The link is listed in the key resources table.
- Any additional information required to reanalyze the data reported in this article is available from the lead contact upon request.

ACKNOWLEDGMENTS

This work is supported in part by the National Institute of General Medical Sciences of the National Institutes of Health under award R35GM150890, a Seed Grant Program from the NSF AI Institute: Molecule Maker Lab Institute (grant no. 2019897) at the University of Illinois Urbana-Champaign, and a Seed Grant Program from GaTech IDEaS. This work used the Delta GPU Supercomputer at NCSA of UIUC through allocation CIS230097 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by NSF grants #2138259, #2138286, #2138307, #2137603, and #2138296. The authors acknowledge the computational resources provided by Microsoft Azure through the Cloud Hub program at GaTech IDEaS and the Microsoft Accelerate Foundation Models Research (AFMR) program.

AUTHOR CONTRIBUTIONS

Conceptualization: Y.L.; methodology: J.L. and Y.L.; software: J.L.; validation: J.L.; formal analysis: J.L.; investigation: J.L.; resources: Y.L.; data curation: J.L.; writing - original draft: J.L. and Y.L.; writing - review and editing: J.L., K.D., and Y.L.; visualization: J.L. and K.D.; supervision: Y.L.; project administration: Y.L.; funding acquisition: Y.L.

DECLARATION OF INTERESTS

The authors declare no competing interests.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES

During the preparation of this work, the authors used ChatGPT in order to improve the readability and language of the article. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- METHOD DETAILS
  - Problem formulation
  - Definition 1(Pareto Optimality)
  - Overview of MosPro
  - Sequence-property prediction models
  - Property-guided protein sequence design
  - Pareto optimization of multiple properties
  - Multi-property benchmarking datasets
  - *In silico* evaluation settings
  - Baseline methods
- QUANTIFICATION AND STATISTICAL ANALYSIS

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2025.112119.

Received: October 28, 2024
Revised: January 3, 2025
Accepted: February 24, 2025
Published: February 27, 2025

REFERENCES

1. Arnold, F.H. (2019). Innovation by Evolution: Bringing New Chemistry to Life (Nobel Lecture). Ange. Chem. Int. Ed. *58*, 14420–14426.
2. Sevgen, E., Moller, J., Lange, A., Parker, J., Quigley, S., Mayer, J., Srivastava, P., Gayatri, S., Hosfield, D., Korshunova, M., et al. (2023). Prot-vae: Protein transformer variational autoencoder for functional protein design. Preprint at bioRxiv. https://doi.org/10.1101/2023.01.23.525232.
3. Repecka, D., Jauniskis, V., Karpus, L., Rembeza, E., Rokaitis, I., Zrimec, J., Poviloniene, S., Laurynenas, A., Viknander, S., Abuajwa, W., et al. (2021). Expanding functional protein sequence spaces using generative adversarial networks. Nat. Mach. Intell. *3*, 324–333.
4. Madani, A., Krause, B., Greene, E.R., Subramanian, S., Mohr, B.P., Holton, J.M., Olmos, J.L., Jr., Xiong, C., Sun, Z.Z., Socher, R., et al. (2023). Large language models generate functional protein sequences across diverse families. Nat. Biotechnol. *41*, 1099–1106.

5. Biswas, S., Khimulya, G., Alley, E.C., Esvelt, K.M., and Church, G.M. (2021). Low-n protein engineering with data-efficient deep learning. Nat. Methods *18*, 389–396.

6. Hie, B., Candido, S., Lin, Z., Kabeli, O., Rao, R., Smetanin, N., Sercu, T., and Rives, A. (2022). A high-level programming language for generative protein design. Preprint at bioRxiv. https://doi.org/10.1101/2022.12.21.521526.

7. Gómez-Bombarelli, R., Wei, J.N., Duvenaud, D., Hernández-Lobato, J.M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T.D., Adams, R.P., and Aspuru-Guzik, A. (2018). Automatic chemical design using a data-driven continuous representation of molecules. ACS Cent. Sci. *4*, 268–276.

8. Brookes, D.H., and Listgarten, J. (2018). Design by adaptive sampling. Preprint at arXiv. https://doi.org/10.48550/arXiv.1810.03714.

9. Ren, Z., Li, J., Ding, F., Zhou, Y., Ma, J., and Peng, J. (2022). Proximal exploration for model-guided protein sequence design. In International Conference on Machine Learning, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, eds. (PMLR), pp. 18520–18536.

10. Kirjner, A., Yim, J., Samusevich, R., Jaakkola, T., Barzilay, R., and Fiete, I. (2023). Optimizing protein fitness using gibbs sampling with graph-based smoothing. Preprint at arXiv. https://doi.org/10.48550/arXiv.2307.00494.

11. Emami, P., Perreault, A., Law, J., Biagioni, D., and St John, P. (2023). Plug & play directed evolution of proteins with gradient-based discrete mcmc. Mach. Learn. Sci. Technol. *4*, 025014.

12. Julian, M.C., Li, L., Garde, S., Wilen, R., and Tessier, P.M. (2017). Efficient affinity maturation of antibody variable domains requires co-selection of compensatory mutations to maintain thermodynamic stability. Sci. Rep. *7*, 45259.

13. Shehata, L., Maurer, D.P., Wec, A.Z., Lilov, A., Champney, E., Sun, T., Archambault, K., Burnina, I., Lynaugh, H., Zhi, X., et al. (2019). Affinity maturation enhances antibody specificity but compromises conformational stability. Cell Rep. *28*, 3300–3308.e4.

14. Rabia, L.A., Desai, A.A., Jhajj, H.S., and Tessier, P.M. (2018). Understanding and overcoming trade-offs between antibody affinity, specificity, stability and solubility. Biochem. Eng. J. *137*, 365–374.

15. Tagasovska, N., Frey, N.C., Loukas, A., Hötzel, I., Lafrance-Vanasse, J., Kelly, R.L., Wu, Y., Rajpal, A., Bonneau, R., Cho, K., et al. (2022). A pareto-optimal compositional energy-based model for sampling and optimization of protein sequences. Preprint at arXiv. https://doi.org/10.48550/arXiv.2210.10838.

16. Makowski, E.K., Kinnunen, P.C., Huang, J., Wu, L., Smith, M.D., Wang, T., Desai, A.A., Streu, C.N., Zhang, Y., Zupancic, J.M., et al. (2022). Co-optimization of therapeutic antibody affinity and specificity using machine learning models that generalize to novel mutational space. Nat. Commun. *13*, 3788.

17. Hong, L., and Kortemme, T. (2024). An Integrative Approach to Protein Sequence Design through Multiobjective Optimization. PLoS Comput Biol.

18. Grathwohl, W., Swersky, K., Hashemi, M., Duvenaud, D., and Maddison, C. (2021). Oops i took a gradient: Scalable sampling for discrete distributions. In ICML, M. Meila and T. Zhang, eds. (PMLR), pp. 3831–3841.

19. Sener, O., and Koltun, V. (2018). Multi-task learning as multi-objective optimization. Adv. Neural Inf. Process. Syst. *31*, 525–536.

20. Notin, P., Kollasch, A., Ritter, D., Van Niekerk, L., Paul, S., Spinner, H., Rollins, N., Shaw, A., Orenbuch, R., Weitzman, R., et al. (2024). Proteingym: Large-scale benchmarks for protein fitness prediction and design. Adv. Neural Inf. Process. Syst. *36*, 64331–64379.

21. Sarkisyan, K.S., Bolotin, D.A., Meer, M.V., Usmanova, D.R., Mishin, A.S., Sharonov, G.V., Ivankov, D.N., Bozhanova, N.G., Baranov, M.S., Soylemez, O., et al. (2016). Local fitness landscape of the green fluorescent protein. Nature *533*, 397–401.

22. Wu, N.C., Dai, L., Olson, C.A., Lloyd-Smith, J.O., and Sun, R. (2016). Adaptation in protein fitness landscapes is facilitated by indirect paths. Elife *5*, e16965.

23. Aakre, C.D., Herrou, J., Phung, T.N., Perchuk, B.S., Crosson, S., and Laub, M.T. (2015). Evolving new protein-protein interaction specificity through promiscuous intermediates. Cell *163*, 594–606.

24. Schymkowitz, J., Borg, J., Stricher, F., Nys, R., Rousseau, F., and Serrano, L. (2005). The foldx web server: an online force field. Nucleic Acids Res. *33*, W382–W388.

25. Shanehsazzadeh, A., Bachas, S., McPartlon, M., Kasun, G., Sutton, J.M., Steiger, A.K., Shuai, R., Kohnert, C., Rakocevic, G., Gutierrez, J.M., et al. (2023). Unlocking *de novo* antibody design with generative artificial intelligence. Preprint at bioRxiv. https://doi.org/10.1101/2023.01.08.523187.

26. Miettinen, K. (1999). Nonlinear Multiobjective Optimization, *12* (Springer Science & Business Media).

27. Ghane-Kanafi, A., and Khorram, E. (2015). A new scalarization method for finding the efficient frontier in non-convex multi-objective problems. Appl. Math. Model. *39*, 7483–7498.

28. Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Trans. Evol. Comput. *6*, 182–197.

29. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. (2021). Highly accurate protein structure prediction with alphafold. Nature *596*, 583–589.

30. Alamdari, S., Thakkar, N., van den Berg, R., Lu, A.X., Fusi, N., Amini, A.P., and Yang, K.K. (2023). Protein generation with evolutionary diffusion: sequence is all you need. Preprint at bioRxiv. https://doi.org/10.1101/2023.09.11.556673.

31. Ruffolo, J.A., Nayfach, S., Gallagher, J., Bhatnagar, A., Beazer, J., Hussain, R., Russ, J., Yip, J., Hill, E., Pacesa, M., et al. (2024). Design of highly functional genome editors by modeling the universe of crispr-cas sequences. Preprint at bioRxiv. https://doi.org/10.1101/2024.04.22.590591.

32. Angermueller, C., Dohan, D., Belanger, D., Deshpande, R., Murphy, K., and Colwell, L. (2019). Model-based reinforcement learning for biological sequence design. In International conference on learning representations, A. Rush, S. Mohamed, D. Song, K. Cho, and M. White, eds.

33. Gupta, A., and Zou, J. (2018). Feedback gan (fbgan) for dna: A novel feedback-loop architecture for optimizing protein functions. Preprint at arXiv. https://doi.org/10.48550/arXiv.1804.01694.

34. Fannjiang, C., and Listgarten, J. (2020). Autofocused oracles for model-based design. Adv. Neural Inf. Processing Syst. *33*, 12945–12956.

35. Gruver, N., Stanton, S., Frey, N.C., Rudner, T.G., Hotzel, I., Lafrance-Vanasse, J., Rajpal, A., Cho, K., and Wilson, A.G. (2023). Protein design with guided discrete diffusion. Preprint at arXiv. https://doi.org/10.48550/arXiv.2305.20009.

36. Hsu, C., Nisonoff, H., Fannjiang, C., and Listgarten, J. (2022). Learning protein fitness models from evolutionary and assay-labeled data. Nat. Biotechnol. *40*, 1114–1122.

37. Zhao, J., Zhang, C., and Luo, Y. (2024). Contrastive fitness learning: Reprogramming protein language models for low-n learning of protein fitness landscape. In International Conference on Research in Computational Molecular Biology, J. Ma, ed. (Springer), pp. 470–474.

38. Yang, Y., Zha, K., Chen, Y.C., Wang, H., and Katabi, D. (2021). Delving into deep imbalanced regression. In International Conference on Machine Learning (ICML), M. Meila and T. Zhang, eds. (PMLR).

39. Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., et al. (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. Science *379*, 1123–1130.

40. Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms. Preprint at arXiv. https://doi.org/10.48550/arXiv.1109.2378.

41. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nat. Methods *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2.

42. Liu, X., Tong, X., and Liu, Q. (2021). Profiling pareto front with multi-objective stein variational gradient descent. NeurIPS *34*, 14721–14733.

43. Boyd, S.P., and Vandenberghe, L. (2004). Convex Optimization (Cambridge university press).

44. Désidéri, J.A. (2012). Multiple-gradient descent algorithm (mgda) for multiobjective optimization. C. R. Math. *350*, 313–318.

45. Jaggi, M. (2013). Revisiting frank-wolfe: Projection-free sparse convex optimization. In International conference on machine learning, S. Dasgupta and D. McAllester, eds. (PMLR), pp. 427–435.

46. Ding, K., Chin, M., Zhao, Y., Huang, W., Mai, B.K., Wang, H., Liu, P., Yang, Y., and Luo, Y. (2024). Machine learning-guided co-optimization of fitness and diversity facilitates combinatorial library design in enzyme engineering. Nat. Commun. *15*, 6392.

47. Fromer, J.C., Graff, D.E., and Coley, C.W. (2024). Pareto optimization to accelerate multi-objective virtual screening. Digital Discov. *3*, 467–481.

48. Bachas, S., Rakocevic, G., Spencer, D., Sastry, A.V., Haile, R., Sutton, J.M., Kasun, G., Stachyra, A., Gutierrez, J.M., Yassine, E., et al. (2022). Antibody optimization enabled by artificial intelligence predictions of binding affinity and naturalness. Preprint at bioRxiv. https://doi.org/10.1101/2022.08.16.504181.

49. Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., et al. (2022). Language models of protein sequences at the scale of evolution enable accurate structure prediction. Preprint at bioRxiv. https://doi.org/10.1101/2022.07.20.500902.

50. Tsien, R.Y. (1998). The green fluorescent protein. Annu. Rev. Biochem. *67*, 509–544.

51. Prendergast, F.G., and Mann, K.G. (1978). Chemical and physical properties of aequorin and the green fluorescent protein isolated from aequorea forskalea. Biochemistry *17*, 3448–3453.

52. Sjöbring, U., Björck, L., and Kastern, W. (1991). Streptococcal protein g. gene structure and protein binding properties. J. Biol. Chem. *266*, 399–405.

53. Sauer-Eriksson, A.E., Kleywegt, G.J., Uhlén, M., and Jones, T.A. (1995). Crystal structure of the c2 fragment of streptococcal protein g in complex with the fc domain of human igg. Structure *3*, 265–278.

54. Castro, E., Godavarthi, A., Rubinfien, J., Givechian, K., Bhaskar, D., and Krishnaswamy, S. (2022). Transformer-based protein generation with regularized latent space optimization. Nat. Mach. Intell. *4*, 840–851.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Multi-property sequence design		
benchmark dataset	This study	Zenodo: https://doi.org/10.5281/zenodo.14884203
Software and algorithms		
FoldX	FoldX Consortium	https://foldxsuite.crg.eu/
Other		
Original code	This study	https://github.com/luo-group/MosPro

METHOD DETAILS

Problem formulation

We formulate our multi-property protein sequence design problem as a multi-objective optimization problem. Denote the vast search space of protein sequences as $\mathcal{X} = \Sigma^L$, where Σ is the set of all canonical amino acids and L is the protein length. Given the amino acid sequence of a protein, our goal is to optimize this sequence by making substitution mutations such that multiple properties of interest of this protein are simultaneously optimized. Mathematically, we aim to find $x \in \mathcal{X}$ that maximizes each element in the vector-based property function: $F(x) = (f_1(x), f_2(x), \ldots, f_m(x))$, where $f_i(x) : \mathcal{X} \to \mathbb{R}$ for $i \in [m]$ is an unknown sequence-property map for the i-th property. Generally, it is not likely to simultaneously optimize all m properties to their maximum value since there usually exist conflicts among the properties. As an alternative, we seek to generate a *Pareto set* of protein sequences where no property can be further optimized without compromising others, which leads to the notion of *Pareto Optimality*:

Definition 1(Pareto Optimality)

For a set of protein sequences $\{x_1, x_2, \ldots, x_N\}$ and a vector-based property function of m properties $F(x) = (f_1(x), f_2(x), \ldots, f_m(x))$, we say that x is *Pareto dominated* by x' if and only if $f_i(x) \leq f_i(x')$, $\forall i \in [m]$ and $f_j(x) < f_j(x')$ for some $j \in [m]$. A sequence x is called *Pareto optimal* if and only if it is not Pareto dominated by any other sequences in the set. The collection of all Pareto optimal sequences $\{x_1^*, x_2^*, \ldots x_k^*\}$ is called the *Pareto set*, and their property values $F(x_1^*), F(x_2^*), \ldots, F(x_k^*)$ are called the *Pareto front*.

Overview of MosPro

An overview of MosPro is shown in Figure 1. MosPro first trains multiple property-specific supervised ML models on available sequence-property data. Next, to achieve property-guided sequence design starting from the initial sequence (e.g., the wild-type sequence), MosPro iteratively applied a discrete sampling algorithm[18] to sample mutations likely to improve the properties. The mutation sampling is guided by the trained property ML models, whose gradient vector suggests the mutation direction predicted to improve a particular property. Lastly, to simultaneously optimize multiple properties, MosPro applies a Pareto optimization scheme to identify an optimal gradient direction along which all properties could be improved most. Following this gradient direction, MosPro samples a substitution mutation and performs a gradient ascent-like procedure to update the current sequence. In analogy to a hill-climbing process on the sequence-property landscape, MosPro iteratively applies this sampling process to accumulate beneficial mutations, rejects intermediate sequences showing no promise of improved properties, and eventually generates sequences for which multiple properties are optimized. Below we describe the details of these steps.

Sequence-property prediction models

Using the experimentally assayed sequence-to-property data[21–23] as training data, we first train a set of property-specific ML prediction models that will later be used as oracles to guide the design of property-improved sequences. Each ML model consists of a 1D-convolution layer and two fully connected layers, which receive the one-hot-encoded protein sequence as input and predict a scalar that represents the activity level of a particular property from a protein's sequence. We used the Mean Square Error (MSE) as the loss function and set learning rate = 0.001, weight decay = 10^{-5}, batch size = 128, and used Adam optimizer with early stop patience of 20. The details of the hyperparameters are listed in Table S3. We randomly split 10% of the data as the validation set. All the predictors can achieve ~0.9 or higher in terms of Pearson and Spearman correlations on the validation set. While we found that the 1D CNN models achieved accurate prediction on our datasets and were computationally efficient

in our experiments, more capable neural network architectures, such as protein language models,[39] can be explored in future work.

Since we are particularly interested in assessing the ability of our sequence design method to improve protein properties from low-score region to high-score region, we constrained the training set of our ML predictors to include only low-property sequences (with a property score lower than the 40th percentile in the entire dataset). To address the data scarcity challenge in existing sequence-property data, we employed a negative data augmentation strategy following a previous study,[10] motivated by the fact that most sequences in the sequence space are non-functional. Specifically, we doubled the training set by randomly sampling sequences from the entire sequence space (Σ^L) and assigning each of them the lowest property score observed in the original training set.

Property-guided protein sequence design

We now introduce the property-guided sequence design method of MosPro, which is based on a recently developed discrete sampling algorithm called Gibbs With Gradient (GWG).[18] While recent studies have explored the application of this sampling algorithm for protein sequence design,[10,11] we extend this technique to the multi-property sequence design problem and additionally introduce a gradient normalization approach to facilitate the sampling process.

For simplicity, we first consider the sequence design task for optimizing a single property and assume that we have trained a property predictor $f_\theta : \Sigma^L \to \mathbb{R}$. Without loss of generality, we assume that a higher property score corresponds to a more desired functionality. The goal of the discrete sampling algorithm is to construct a sequence proposal distribution $p(x)$ such that sequences with higher predicted property score will be sampled with higher likelihood, i.e., $p(x) = \frac{1}{Z} \exp(f_\theta(x))$ with Z as the normalization constant. One common approach to achieve such sampling in discrete space is using Gibbs sampling. At each step, Gibbs sampling proposes a sequence x' conditional on the current sequence x, following a proposal distribution $q(x'|x)$, where x' is constructed by introducing mutations into x.

In our problem, we leverage the property predictor f_θ to predict which mutations are more likely to lead to property improvement. Grathwohl et al.[18] proposed a method called Gibbs With Gradient (GWG) that follows the gradient direction of $f_\theta(x)$ with respect to x to sample beneficial mutations, as this is predicted to be the steepest direction for property enhancement. Specifically, given the current sequence x, GWG defined the proposal distribution for the updated sequence x' as

$$q(x'|x) \propto \exp(\tilde{d}(x)/2) \cdot 1(x' \in H(x)), \quad \text{(Equation 1)}$$

where $H(x)$ is the 1-Hamming ball (the set of sequences differing from x by one amino acid) and $\tilde{d}(x) \triangleq f_\theta(x') - f_\theta(x)$, which can be approximated using Taylor-series expansion:

$$\tilde{d}(x)_{ij} = \nabla_x f_\theta(x)_{ij} - x_i^T \nabla_x f_\theta(x)_i. \quad \text{(Equation 2)}$$

The distribution $q(x'|x)$ can be positionwisely factorized as $q(x'|x) = \prod_{i=1}^{L} q_i(x'_{(i)}|x)$, where $x'_{(i)}$ is the i-th amino acid of sequence x'. Since we are considering the 1-Hamming ball, the proposed sequence x' can be constructed by sampling a different amino acid from the alphabet Σ for the i-th position:

$$x'_{(i)} \sim q_i(\cdot|x) = \text{Categorical}\left(\text{Softmax}\left(\frac{\tilde{d}(x)}{\tau}\right)\right), \quad \text{(Equation 3)}$$

where τ is the temperature parameter and set to 0.01 throughout our experiments, following the default choice of the previous work.[10] The proposed sequence x' is then accepted using Metropolis-Hasting with probability

$$\alpha = \min\left(\exp(f_\theta(x') - f_\theta(x)) \frac{q(x|x')}{q(x'|x)}, 1\right). \quad \text{(Equation 4)}$$

Iterative sampling

Inspired by the iterative process in direct evolution, MosPro designs property-enhanced proteins by repeating the above sampling process for T rounds to accumulate beneficial mutations to the starting sequences which only have a low level of properties. This iterative sampling allows MosPro to escape from the low-property region and explore broadly in the sequence space to identify high-property sequences. In each round $r \in [T]$, we sampled m mutants for each starting sequence, forming a set X_r. To prevent the designed sequence set X_r from being redundant or exponentially large, we followed the strategy in previous work[10] to perform hierarchical clustering[40] on X_r. Specifically, we applied the hierarchical clustering on X_r with 500 cluster centers based on the hamming distance between sequences. We used the hierarchical clustering functions 'linkage' and 'fcluster' in the SciPy package[41] to perform the clustering. After clustering, we only kept the sequences with top-1 predicted property scores in each cluster as the starting sequences for the next round.

Pareto optimization of multiple properties

When there are multiple properties of interest, our goal is to sample sequences based on their property-specific predictors $f_{\theta_1}, \ldots, f_{\theta_m}$ such that their properties are Pareto optimal. A straightforward solution is linear scalarization[26] that combines multiple objectives with convex combination $f_\lambda(x) = \sum_{i=1}^{m} \lambda_i f_{\theta_i}(x)$ such that $\sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0$. Despite being simple, the optimization results heavily depend on the choices of combination weights. It is also found that while this approach can obtain Pareto optimal solutions that lie on the convex part of the Pareto front, it is unable to handle the concave part of Pareto fronts.[42,43]

We thus employ the multiple gradient descent (MGD)[44] to address this challenge. MGD was originally developed for multi-objective optimization, in which the goal is to minimize multiple criteria in continuous space. Here, we applied it to sample protein sequences, in which the optimizations of multiple properties can be optimally balanced. Specifically, we use a gradient descent-like update: $x' \leftarrow x - \eta g^*(x)$, where η is a small step size and $g^*(x)$ a vector field describing the update direction, which is chosen to maximize the smallest improvement rate among all properties:

$$g^*(x) \propto \arg\max_{g \in \mathbb{R}^L} \left\{ \min_{i \in [m]} \langle g, g_i(x) \rangle, \text{s.t. } \|g\| \leq 1 \right\}, \quad \text{(Equation 5)}$$

where $g_i(x) \triangleq \nabla_x f_{\theta_i}(x)$. Here, $g^*(x)$ is encouraged to have positive inner products with all $g_i(x)$, i.e., all predicted properties will be improved by going to sequence x'. When there are conflicting gradients, $g^*(x)$ will be set to 0. Using Lagrangian duality, Desideri[44] showed that the optimal solution is $g^*(x) = \sum_{i=1}^{m} \lambda_i \nabla f_{\theta_i}(x)$, where the weight vector $\{\lambda_i\}_{i=1}^{m}$ is the solution of the optimization problem:

$$\min_{\lambda_i} \| \sum_{i=1}^{m} \lambda_i \nabla f_{\theta_i}(x) \|_2^2 \quad \text{s.t.} \sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, \forall i \in [m], \quad \text{(Equation 6)}$$

This optimization has a closed-form solution for $m = 2$:

$$\lambda_1 = \text{clamp}_{0,1}\left(\frac{(\nabla f_{\theta_1}(x) - \nabla f_{\theta_2}(x))^T \nabla f_{\theta_2}(x)}{\| \nabla f_{\theta_1}(x) - \nabla f_{\theta_2}(x) \|_2^2} \right), \quad \text{(Equation 7)}$$

where $\text{clamp}_{0,1}(x) = \max(\min(x, 1), 0)$, and $\lambda_2 = 1 - \lambda_1$. For $m > 2$, a fast algorithm known as the Frank-Wolfe algorithm[45] can leverage the solutions from Equation 7 to efficiently find the solutions of Equation 6.[19]

After obtaining the gradient direction $g^*(x)$ that optimally maximizes all properties, we use it to replace the gradient term $\nabla f_\theta(x)$ for the single property in Equation 2, pushing the designed sequences toward the Pareto front for both the properties.

Several recent methods have been proposed for protein sequence design in multi-objective scenarios. For instance, Emami et al.[11] introduced a gradient-guided Markov Chain Monte Carlo (MCMC) sampling framework that used fixed weights to linearly combine individual gradients (linear scalarization), but this approach is computationally expensive. Pareto Optimality[43] is also widely used in multi-objective optimization to achieve optimal trade-offs. MODIFY,[46] for example, identifies the Pareto front to co-optimize protein fitness and diversity for enzyme library design, though it relies entirely on unsupervised property predictors to score variants, unlike MosPro which uses differentiable, supervised predictors to guide sequence design. Similarly, MolPAL[47] employs Pareto optimization in active learning for multi-objective virtual screening of small molecules, focusing on data acquisition rather than explicit molecule design to deliver enhanced properties. Another line of work[15] employs a Pareto-optimal gradient composition scheme similar to ours but operates in the continuous space of latent sequence representations using an energy-based model. Unlike this latent space-based approach, which requires decoding entire sequences from latent vector representations, MosPro directly proposes specific edit positions and mutant types on the starting sequence for iterative sequence mutations. This explicit mutation framework not only reduces the computational complexity of designing a full sequence from scratch but also provides more precise control over the engineering budget during the design process.

Gradient normalization

In our experiments, we observed that the significant difference between the L_2 norms of $\nabla_x f_{\theta_1}(x)$ and $\nabla_x f_{\theta_2}(x)$ will lead to a biased update vector $g^*(x)$. For instance, if $\| \nabla_x f_{\theta_1}(x) \|_2 \gg \| \nabla_x f_{\theta_2}(x) \|_2$, the vector $g^*(x)$ will be very close to $\nabla_x f_{\theta_2}(x)$, making the sampling process biased toward optimizing for property 2 (Figure S1). Therefore, we propose to normalize the gradient of the fitness predictor using the Frobenius norm before performing the optimization in Equation 5, i.e.,

$$\nabla_x f_\theta(x)_{\text{norm}} = \nabla_x f_\theta(x) / \| \nabla_x f_\theta(x) \|_F, \quad \text{(Equation 8)}$$

where $\| \nabla_x f_\theta(x) \|_F = (\text{trace}(\nabla_x f_\theta(x)^T \nabla_x f_\theta(x)))^{\frac{1}{2}}$. We then replaced the individual gradients $g_i(x) = \nabla_x f_{\theta_i}(x)$ in Equation 5 by their normalized versions $\nabla_x f_{\theta_i}(x)_{\text{norm}}$. In our results, we found that this gradient normalization led to better sampling results (Table S1).

Multi-property benchmarking datasets

Most protein sequence libraries validated in wet labs focused on a single property. To test the optimization ability of MosPro, we incorporated two computational-based properties: stability and naturalness. Protein mutant stability is quantified by the change in Gibbs free energy relative to the wild-type protein ($\Delta\Delta G$). This metric is computed by averaging five runs of FoldX,[24] where a lower $\Delta\Delta G$ indicates a more stable mutant. The naturalness of a protein sequence measures its resemblance to naturally evolved proteins.

Natural-like proteins are more likely to satisfy the evolutionary or structural constraints and avoid undesired functional consequences. Following a previous study,[48] we define the naturalness of a mutant protein x, in relation to the wild-type x^{WT}, as follows:

$$\text{naturalness}(x) = \sum_{t \in M} \left[\log P(x_t|x_{-t}) - \log P(x_t^{WT}|x_{-t}^{WT})\right], \quad \text{(Equation 9)}$$

where M represents the index set of mutation sites in x compared to x^{WT}, x_t is the t-th amino acid of x, and x_{-t} denotes the sequence x with the t-th amino acid masked. We obtain the conditional probability $P(x_t|x_{-t})$ from a pre-trained protein language model, specifically employing the 33-layer ESM-2 model[49] to calculate the naturalness scores.

Combining the above stability and naturalness metrics with published data of experimentally characterized protein properties, we constructed four benchmark tasks of multi-property protein sequence design, in which each sequence is labeled with two properties. The size of each task can be found in Table S2. These benchmark tasks are also part of the contributions of this work, providing a useful resource for evaluating future methods on multi-objective protein sequence design. Below, we provide an overview of the four multi-property sequence design tasks.

Green fluorescent proteins (GFP)

The green fluorescent protein (GFP), known for its bright green fluorescence under blue to ultraviolet light, was originally isolated from the jellyfish *Aequorea victoria* (avGFP) and has since become indispensable in biological and biochemical research.[50,51] Its applications range from serving as a fluorescent marker in genetic engineering to facilitating various cell biology studies.[21] The wild-type avGFP consists of 238 amino acids, creating a vast search space of 20^{238} possible sequences. The study by Sarkisyan et al.[21] characterized the local fitness landscape of avGFP by measuring the fluorescence of 51,715 variants in the sequence space, with an average of 3.7 mutations per sequence. We used the experimentally assayed fluorescence as the primary property in our design task and additionally applied FoldX to compute the stability scores for all these sequences, resulting in the design task *GFP-stability* where the goal is to optimize fluorescence and stability.

Protein GB1

GB1, or the Epistatic region of protein G domain B1, is an immunoglobulin-binding protein found in Streptococcal bacteria.[52,53] Wu et al.[22] experimentally measured the fitness of 149,361 variants at four sites (V39, D40, G41, and V54) in an epistatic region of GB1. Here, the GB1 fitness is defined as a combination of its folding stability, represented as the fraction of folded protein, and its binding affinity to IgG-Fc, compared to the wild-type. This specific landscape presents a search space of $20^4 = 160,000$ potential variants. Similar to the approach taken with GFP, we computed the stability scores for all sequences within this dataset. This led to a design task for GB1, denoted as *GB1-stability*.

Antitoxin ParD3

ParD3 is an antitoxin produced by *Mesorhizobium opportunistum*, known for its ability to bind and neutralize its associated toxin, ParE3, which otherwise slows down cell growth.[23] In *Escherichia coli*, co-expression and binding of ParD3 and ParE3 inhibit the toxic effects of ParE3, facilitating normal cell growth.[23] However, mutations in ParE3 can disrupt its binding with ParD3, releasing the toxin and impairing growth.[23] For instance, the non-cognate toxin ParE2 exhibits only about 16% of the binding affinity with wild-type ParD3 as compared to ParE3. Following the criteria set by the study,[23] we define the property of ParD3 in terms of its binding affinity with both ParE3 and ParE2 toxins. From the study,[23] we sourced a deep mutational scanning library of all $20^3 = 8,000$ ParD3 mutants at three mutation sites (D60, K63, E79), and their respective binding affinities with ParE3 and ParE2. This compilation constitutes a multi-property design task denoted as *ParD3-ParE2* with the objective of optimizing the binding affinities to both ParE3 and ParE2. Additionally, we computed the naturalness scores for the ParD3 variants, leading to another multi-property task, *ParD3-naturalness*, where the objective is to optimize the binding affinity to ParE3 and the naturalness of ParD3 variants.

In silico evaluation settings

To evaluate the sequence design capabilities of MosPro, we followed the *in silico* evaluation paradigm commonly adopted by prior studies,[9,10,54] as described below for each task.

GFP-stability: We selected the sequences from the GFP dataset[21] whose fluorescence and stability properties are both in the lowest 40% values as the training data for the sequence-property prediction models and the starting sequence set. This training data split simulated the real-world scenario where only low-property sequences are available and the goal is to design sequences of higher properties. To evaluate the property values of the designed sequences, we used another sequence-to-fluorescence predictor developed by Kirjner et al.[10] for the fluorescence property, and FoldX[24] for the stability property. Note that the sequence-to-fluorescence predictor[10] used for evaluation was trained on the entire GFP dataset[21] so it can accurately assess the sequence designs in high-property regimes. Using two fluorescence predictors trained on different coverage of the GFP dataset[21] also ensures that there is no circularity issue or data leakage between training and evaluation and that the evaluation results will not be over-estimated due to the design algorithm's overfit on the training oracle.

ParD3-ParE2

The sequences from the ParD3 dataset[23] with both binding affinities to ParE3 and ParE2 at the lowest 40% values are split as the training set for the two sequence-property prediction models as well as the starting set for MosPro and the baselines. Since ParD3 dataset is a complete combinatorial library for all ParD3 mutants at three mutation sites, the evaluation of the designed sequences is a query of the ground-truth value in the ParD3 dataset.

ParD3-naturalness

We selected from the ParD3 dataset[23] the sequences with both the lowest 40% binding affinity to ParE3 and the lowest 40% naturalness scores as the training data for the sequence-property predictors. The training data is also used as the starting sequences for MosPro and baseline methods. Similar to ParD3-ParE2 task, the evaluation of the designed sequences is conducted by directly retrieving the experimentally labeled binding affinity values from the ParD3 dataset. The naturalness of the designed sequences is evaluated using the same pretrained protein language model ESM-2.[49]

GB1-stability

We also selected the sequences from GB1 dataset[22] with both fitness and stability properties at the lowest 40% values as the training set and starting sequence set. Since the experimental GB1 landscape assayed the fitness of all four-site mutants, the fitness evaluation of designed GB1 sequences is performed by retrieving the experimentally labeled fitness from the GB1 dataset. For the stability of designed sequences, we still use FoldX[24] for evaluation.

Baseline methods

GGS[10]

This baseline method used a discrete sample algorithm similar to the one used in MosPro but only focused on a single objective. We did not apply the graph-smoothing trick of this method due to its limited generalizability across different protein landscapes.

Gibbs sampling with linear scalarization

This baseline is a multi-objective optimization method based on the linear combination of objectives ('linear scalarization'), in which a weight $\lambda \in [0, 1]$ was used to combine the two properties f_1, f_2 to a single objective $f = \lambda f_1 + (1 - \lambda) f_2$, with $\lambda = 0.1, 0.3, 0.5, 0.7, 0.9$.

NSGA-2[28]

NSGA-2 is a multi-objective genetic algorithm. In each iteration, NSGA-2 creates mutants by introducing mutations into the currently available mutants or recombining the existing mutations. NSGA-2 then sorts all mutants by their predicted property values and prioritizes mutants that are Pareto optimal. We used the same sequence-to-property prediction models as in MosPro to generate the predicted property values for NSGA-2.

QUANTIFICATION AND STATISTICAL ANALYSIS

We used the Numpy package to compute the mean property values for Figure 2. The number of evaluated sequences, and the range of the bars and whiskers in Figures 4 and S2 can be found in the figure legends.