# Steering Generative Models with Experimental Data for Protein Fitness Optimization

**Jason Yang**[†]
Chemistry & Chemical Engineering
California Institute of Technology

**Wenda Chu**[†]
Computing & Mathematical Sciences
California Institute of Technology

**Daniel Khalil**
Computing & Mathematical Sciences
California Institute of Technology

**Raul Astudillo**
Computing & Mathematical Sciences
California Institute of Technology

**Bruce J. Wittmann**
Office of the Chief Scientific Officer
Microsoft Corporation

**Frances H. Arnold**
Chemistry & Chemical Engineering
Biology & Biological Engineering
California Institute of Technology

**Yisong Yue**[*]
Computing & Mathematical Sciences
California Institute of Technology

## Abstract

Protein fitness optimization involves finding a protein sequence that maximizes desired quantitative properties in a combinatorially large design space of possible sequences. Recent advances in steering protein generative models (e.g., diffusion models and language models) with labeled data offer a promising approach. However, most previous studies have optimized surrogate rewards and/or utilized large amounts of labeled data for steering, making it unclear how well existing methods perform and compare to each other in real-world optimization campaigns where fitness is measured through low-throughput wet-lab assays. In this study, we explore fitness optimization using small amounts (hundreds) of labeled sequence-fitness pairs and comprehensively evaluate strategies such as classifier guidance and posterior sampling for guiding generation from different discrete diffusion models of protein sequences. We also demonstrate how guidance can be integrated into adaptive sequence selection akin to Thompson sampling in Bayesian optimization, showing that plug-and-play guidance strategies offer advantages over alternatives such as reinforcement learning with protein language models. Overall, we provide practical insights into how to effectively steer modern generative models for next-generation protein fitness optimization.

## 1 Introduction

Proteins, sequences of amino acids, can be optimized for useful properties such as binding affinity, catalytic activity, or stability, numerically quantified as "fitness." However, protein optimization is

[*]yyue@caltech.edu
[†]These authors contributed equally to this work.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

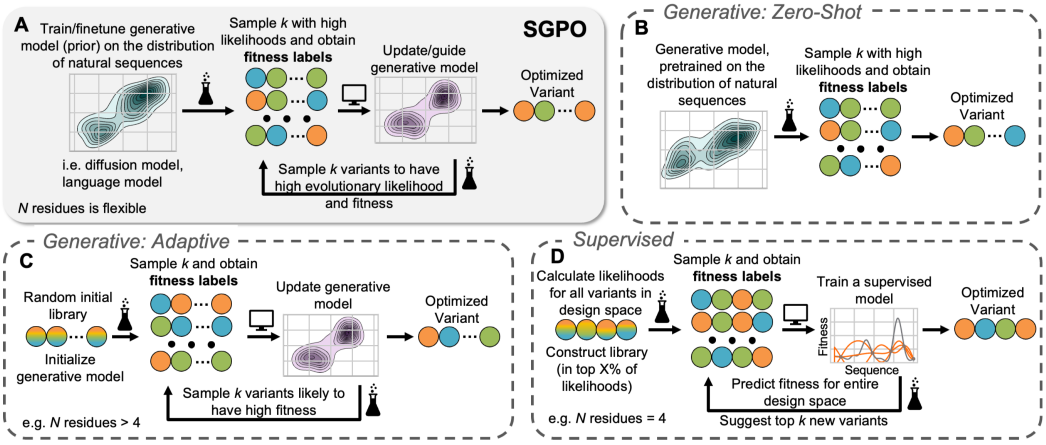


Figure 1: **Comparison of steered generation for protein optimization (SGPO) to other ML-assisted workflows for protein engineering.** **(A)** SGPO involves initializing a generative prior model to sample sequences with high natural likelihoods and steering that model with assay-labeled fitness data. Optimization is difficult because the design space is vast, and the throughput of wet-lab fitness assays (Erlenmeyer flask icon) is low, so adaptive learning across multiple iterations is beneficial. Previous methods have utilized generative models such as **(B)** fully zero-shot methods that sample highly natural sequences but do not utilize labeled fitness data or **(C)** those that only utilize labeled fitness. **(D)** Alternatively, supervised approaches involve enumerating to calculate fitness predictions for all variants in a design space, limiting them to optimizing few residues (i.e., $N < 9$).

challenging: the design space of proteins is enormous, as a protein of length $M$ can be constructed in $20^M$ different ways, of which only a negligible fraction are functional (Romero & Arnold, 2009). Moreover, most wet-lab assays only provide $10^2 - 10^3$ fitness labels. Consequently, researchers often rely on directed evolution, an iterative process aiming to incrementally improve protein fitness (Packer & Liu, 2015) through multiple rounds of mutation and experimental screening. In each round, a protein is mutated, the variants' fitnesses are measured, and the most beneficial variant is selected for the next iteration. However, this approach can be slow, often accumulating only one mutation per round, and inefficient, as it performs a local search limited to closely related protein sequences.

In recent years, there has been a growing interest in developing machine learning (ML)-assisted methods to optimize protein fitness more efficiently (Yang et al., 2019; Wittmann et al., 2021a; Hie & Yang, 2022; Yang et al., 2024, 2025c). Many recent studies have focused on generative approaches combining unlabeled and labeled data for protein design. Broadly, these methods achieve *conditional generation* by steering generative priors of natural protein sequences (Freschlin et al., 2022) using fitness data, thereby enabling incorporation of the steered models into adaptive optimization cycles (Hie & Yang, 2022). We refer to this class of methods as **S**teered **G**eneration for **P**rotein **O**ptimization (SGPO). These methods address the individual limitations of previous approaches (Fig. 1A, Table 1). First, SGPO leverages labeled data, which is essential for fitness goals that deviate from natural functions (e.g., engineering enzymes for non-native activities (Arnold, 2018; Yang et al., 2025b)), unlike zero-shot methods relying solely on generative priors of natural sequences (*Generative: Zero-Shot*, Fig. 1B, Hie et al. 2023; Sumida et al. 2024; Fei et al. 2025; Seki et al. 2025; Lambert et al. 2025). Second, generative priors (Wu et al., 2021; Hsu et al., 2024) sample sequences with high evolutionary likelihoods and potentially higher fitness, giving these methods a significant advantage over approaches relying exclusively on labeled data (Song & Li, 2023; Stanton et al., 2022; Gupta & Zou, 2019; Brookes & Listgarten, 2020; Jain et al., 2022; Kim et al., 2025; Angermueller et al., 2020; Hie & Yang, 2022) (*Generative: Adaptive*, Fig. 1C). Finally, SGPO scales to larger design spaces, unlike most supervised ML-assisted directed evolution (MLDE) approaches, which require enumerating and scoring all variants in the design space (*Supervised*, Fig. 1D) (Wu et al., 2019; Wittmann et al., 2021b; Yang et al., 2025b; Li et al., 2025a; Vornholt et al., 2024; Jiang et al., 2024; Hsu et al., 2022; Ding et al., 2024; Hawkins-Hooker et al., 2024; Zhao et al., 2024a; Thomas et al., 2025; Sun et al., 2025).

Despite these advantages, SGPO methods still face practical limitations in real-world fitness optimization, particularly across two major classes of approaches: guiding discrete diffusion models (Nisonoff et al., 2025; Stark et al., 2024; Klarner et al., 2024; Gruver et al., 2023; Lisanza et al.,

Table 1: **SGPO is a general approach for protein fitness optimization that does not face the individual limitations of other strategies.** Namely, SGPO utilizes zero-shot knowledge from the natural distribution of proteins, can be guided by assay-labeled fitness data, and can optimize many residues ($N$) simultaneously. Beyond those listed here, there are many other studies that combine different elements of these approaches.

| Approach | Prior Information Used? | Assay Fitness Used? | Scales to large $N$? | Protein Examples (non-exhaustive) |
|---|---|---|---|---|
| **SGPO** | ✓ | ✓ | ✓ | Lisanza et al. (2025); Widatalla et al. (2024); Stocco et al. (2024); Nisonoff et al. (2025); Brookes et al. (2019); Blalock et al. (2025); Goel et al. (2025); Huang et al. (2025) |
| Generative: Zero-Shot | ✓ | × | ✓ | Hie et al. (2023); Sumida et al. (2024); Fei et al. (2025); Seki et al. (2025); Lambert et al. (2025) |
| Generative: Adaptive | × | ✓ | ✓ | Song & Li (2023); Jain et al. (2022); Angermueller et al. (2020); Stanton et al. (2022); Brookes & Listgarten (2020) |
| Supervised | ✓ | ✓ | × | Wittmann et al. (2021b); Ding et al. (2024); Hawkins-Hooker et al. (2024); Zhao et al. (2024a); Sun et al. (2025) |

2025; Goel et al., 2025) and finetuning models such as protein language models (PLMs) through reinforcement learning (RL) (Ruffolo & Madani, 2024; Widatalla et al., 2024; Stocco et al., 2024; Blalock et al., 2025; Wang et al., 2025c). The limitations of prior work are summarized as follows: (1) Few previous studies have explored steering with few ($10^2 - 10^3$) labeled sequences (Lisanza et al., 2025; Stocco et al., 2024) for protein optimization based on real fitness data, e.g. activity or fluorescence, rather than computational surrogates (Lisanza et al., 2025; Blalock et al., 2025). (2) Most studies only evaluate one type of generative prior and steering strategy, so it is unclear how different combinations perform in practice. (3) There is room to incorporate principles from adaptive optimization, such as uncertainty-aware exploration (e.g., Bayesian optimization), which have shown clear benefits in protein engineering (Vornholt et al., 2024; Yang et al., 2025b).

In this study, we aim to understand the best practices for integrating SGPO into real-world engineering workflows. We focus here on modern generative models (i.e., discrete diffusion, language models) but acknowledge that other related methods are relevant, such as those based on variational autoencoders (Brookes et al., 2019; Torres et al., 2024) and other adaptive search strategies (Kirjner et al., 2024; Sinai et al., 2020; Ren et al., 2022). We explore the following questions: Which steering strategies perform best, and with which types of models? How can we utilize uncertainty to better explore the design space when performing guidance? **Overall, we make the following key contributions:**

1. We motivate SGPO as a useful, general framework and contextualize existing methods for protein optimization under this umbrella.
2. We comprehensively evaluate design decisions for SGPO, including different generative models for sequences and steering strategies (Fig. 2 & 3, Section 2), offering best practices for protein optimization with few fitness labels.
3. We introduce ideas from adaptive optimization into SGPO by proposing a method that ensembles multiple plug-and-play fitness predictors and leverages their predictive uncertainty to enable more efficient exploration.
4. We are the first to adapt *decoupled annealing posterior sampling* (Zhang et al., 2025) for SGPO, and this type of plug-and-play guidance has the strongest performance overall.

On the TrpB, CreiLOV, and GB1 protein fitness datasets, we find that SGPO methods can consistently identify high-fitness protein variants. In particular, our results highlight the advantages of plug-and-play guidance with diffusion models over finetuned language models—offering greater steerability and lower computational cost. To support future research and real-world adoption, our extensive, user-friendly code is available at `https://github.com/jsunn-y/SGPO`.

## 2 Related work

**Generative models for discrete sequences.** The most widely adopted generative models for natural protein sequences are PLMs, such as autoregressive transformers (Nijkamp et al., 2023) and masked language models (Rives et al., 2021). Increasingly, various diffusion model (Ho et al., 2020) architectures have shown efficacy for modeling discrete data ($\mathbf{x}$) (Li et al., 2025b), such as protein sequences (Alamdari et al., 2024; Wang et al., 2024), leveraging many similar learning techniques such as masking or autoregressive decoding (Sahoo et al., 2024; Lou et al., 2024; Nie et al., 2025; Shi et al., 2024) (Fig. 2). These generative prior models $p(\mathbf{x})$ can be categorized broadly into two types: those that perform diffusion in a continuous latent space (Li et al., 2022; Chen et al., 2023b; Dieleman et al., 2022) and those that diffuse directly over discrete space (Fig. 2). In the protein domain, it has also been shown that latent diffusion over embeddings from PLMs can be more effective (Meshchaninov et al., 2025; Chen et al., 2023a; Torres et al., 2025). Alternatively, models performing diffusion in discrete space use a transition matrix to update all discrete states in each timestep (D3PM) (Austin et al., 2021), which has later been formulated as continuous-time Markov chains (Lou et al., 2024; Campbell et al., 2022, 2024; Schiff et al., 2024). Two common ways to add noise to discrete sequences are to use uniform noise matrices or absorbing state (masking) matrices (Fig. 2). These have been followed by simplified frameworks showing some of the highest performance for modeling natural language, such as masked diffusion language models (MDLMs) (Sahoo et al., 2024; Hoogeboom et al., 2022; Shi et al., 2024) and a variation that uses uniform noise called uniform diffusion language models (UDLMs) (Schiff et al., 2024). We elaborate more on these methods in Section A.3.

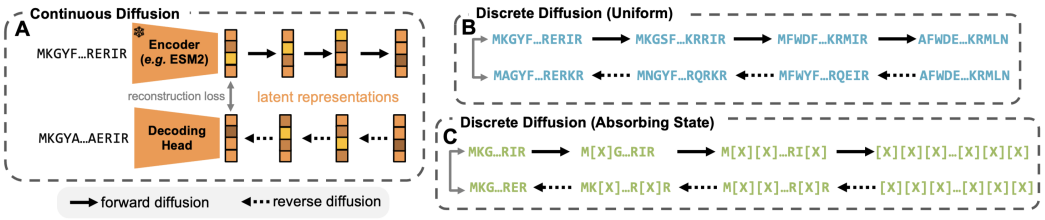


Figure 2: **Overview of different approaches to train diffusion models over discrete state spaces.** During inference, a noised latent representation or sequence is decoded into a reasonable sequence (bottom track for each method). [X] refers to a masked token.

**Plug-and-play guidance strategies.** An advantage of diffusion models is the ability to perform plug-and-play guidance based on fitness labels ($\mathbf{y}$) without finetuning the generative prior model weights, resulting in reduced training costs and potentially strong signal despite having few ($\sim 10^2$) labels. Guiding a continuous diffusion model often involves skewing the learned score function using gradients from a supervised value function that can predict labels $\mathbf{y}$ from data $\mathbf{x}$ (Chung et al., 2023; Zheng et al., 2025; Soares et al., 2025). These methods are often referred to as posterior sampling, as they aim to sample from the posterior distribution, $p(\mathbf{x}|\mathbf{y})$. Recent works extend this idea to guiding discrete diffusion models. Classifier guidance (CG) (Nisonoff et al., 2025) skews the rate matrix of the reverse time Markov chain of discrete diffusion models using a time-dependent value function, $p(\mathbf{y}|\mathbf{x}_t, t)$; variable splitting methods (DAPS) (Zhang et al., 2025; Chu et al., 2025) use discrete diffusion models as denoisers and only require a value function of clean data, $p(\mathbf{y}|\mathbf{x}_0)$; diffusion optimized sampling (NOS) (Gruver et al., 2023) trains a value function on continuous embeddings of discrete tokens and optimizes the embedding for higher fitness; sequential Monte Carlo methods (SMC) (Li et al., 2024a; Uehara et al., 2025; Wu et al., 2024; Lee et al., 2025a; Singhal et al., 2025) evolve multiple particles from a series of distributions to approximate the posterior distribution in limit. We explain these methods in more detail in Section A.4, along with other variations on the guidance process. In this study, we focus on CG, DAPS, and NOS as guidance techniques (Fig. 3). Future work could also consider guidance techniques for autoregressive language models, such as future discriminators for generation (FUDGE) (Yang & Klein, 2021), plug and play language models (PPLM) (Dathathri et al., 2020), and twisted SMC (Zhao et al., 2024b; Amin et al., 2025b). Additionally, Xiong et al. (2025) demonstrate how guidance generalizes to masked language models and order-agnostic autoregressive models.

**Reinforcement learning via model finetuning.** We consider RL broadly here as techniques that achieve conditional generation by finetuning generative models with labeled data, thus pushing

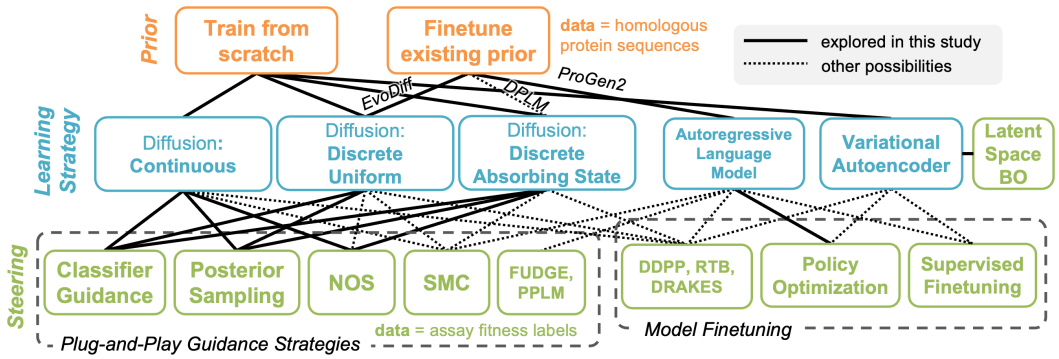


Figure 3: **Methods design space for SGPO: a non-exhaustive landscape of generative models for protein *sequences* and methods to steer them with labeled data.** Three major types of diffusion models for sequences include those that perform diffusion over continuous space and those that perform diffusion over discrete space with a uniform or absorbing state (masking) noising process. Various types of guidance strategies are compatible with certain models, in green (NOS: diffusion optimization sampling, SMC: sequential monte carlo, FUDGE: future discriminators for generation, PPLM: plug and play language models, DDPP: discrete denoising posterior prediction, RTB: relative trajectory balance, DPLM: diffusion protein language model, BO: Bayesian optimization). Differently, language models and variational autoencoders can be aligned with labeled data via reinforcement learning such as policy optimization or supervised finetuning.

those models to produce more favorable generations. There are emerging RL techniques applied to discrete diffusion models, including discrete denoising posterior prediction (DDPP) (Rector-Brooks et al., 2025), relative trajectory balance (RTB) (Venkatraman et al., 2024; Bartoldson et al., 2025; Venkatraman et al., 2025), and direct reward backpropagation with gumbel softmax trick (DRAKES) (Wang et al., 2025a). While the above strategies are specific to discrete diffusion models, supervised fine-tuning (SFT) and policy optimization are two important techniques used in RL that can be broadly applied to generative models such as language models (Fig. 3). Policy optimization has generally shown better performance than SFT (Stocco et al., 2024; Blalock et al., 2025); in particular, direct preference optimization (DPO) is often used for its algorithmic simplicity and ease of training (Rafailov et al., 2023) (details in Section A.4). RL has demonstrated utility for aligning generative models of proteins (language models, inverse folding models, variational autoencoders) with properties like stability (Widatalla et al., 2024; Blalock et al., 2025; Stocco et al., 2024; Lim et al., 2025), but these methods can have high computational costs of finetuning and may require large amounts of labels ($> 10^3$) to effectively steer generations. We include DPO with an autoregressive PLM (finetuned ProGen2 (Nijkamp et al., 2023)) as a baseline.

**Adaptive optimization.** Protein engineering is commonly conducted through adaptive workflows such as directed evolution Packer & Liu (2015) or ML-based approaches such as Bayesian optimization (Frazier, 2018; Stanton et al., 2022). These methods follow an iterative loop: labeled data is collected via expensive wet-lab assays, a surrogate model $p(\mathbf{y}|\mathbf{x})$ is trained or updated, an acquisition function implied by the surrogate is used to propose new sequences to evaluate, and the cycle repeats (Hie & Yang, 2022; Vornholt et al., 2024; Yang et al., 2025b). The surrogate model, often a Gaussian process or a deep ensemble, provides uncertainty estimates, which are used by an acquisition function (e.g., expected improvement, Thompson sampling) to balance exploration and exploitation of the design space. In this study, we adapt these ideas to guide diffusion models for protein sequence generation, as described in Section 4.3. A closely related line of work is latent space Bayesian optimization (Maus et al., 2022; Stanton et al., 2022; Gómez-Bombarelli et al., 2018; Castro et al., 2022; Torres et al., 2024; Lee et al., 2025b), which searches for optimal sequences within a latent space—typically learned by an autoencoder, which can implicitly capture a prior on natural protein sequences. In this work, we compare against APEXGo (Torres et al., 2024), a method that performs trust-region Bayesian optimization in the latent space of a variational autoencoder trained over protein sequences. There are also related methods that involve conditional sampling from a prior (Brookes et al., 2019). However, we note that SGPO offers greater flexibility by avoiding reliance on an explicit latent space, which enables the use of modern, more powerful generative models such as diffusion models and protein language models that are not easily accommodated by traditional latent Bayesian optimization pipelines.

## 3 Problem setup

We focus on evaluating methods that fall under SGPO, where the primary downstream task entails starting from a known sequence with some level of fitness for a target objective (i.e. activity, stability, fluorescence, binding, etc.) and identifying a modified sequence with maximized fitness, where real-world fitness can only be measured for $10^2$ to $10^3$ sequences. Our goal is to sample sequences with maximum fitness $\mathbf{y}$ from the *generative prior* $p(\mathbf{x})$, which is trained on the multiple sequence alignment (MSA) of homologous protein sequences that are evolutionarily related to a known protein with some level of desired fitness (details in Section A.3). This model can be thought of as capturing the distribution of sequences with high likelihood from a given protein family.

During inference, sequences can be sampled *unconditionally* from $p(\mathbf{x})$, or sampling can be *guided* using a supervised model of the form $p(\mathbf{y}|\mathbf{x}) \propto \exp(f(\mathbf{x})/\beta)$, where $f(\cdot)$ is a learned fitness predictor—also referred to as the *classifier* or *value function*. This predictor is trained on a small number of labeled sequence-fitness pairs (typically in the hundreds) to reflect practical data limitations. The goal of guided sampling is to generate protein sequences from the posterior distribution, $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{x}) \exp(f(\mathbf{x})/\beta)$. We use a computational *oracle* to acquire and evaluate fitness labels $\mathbf{y}$, to simulate how fitness would be measured in a real-world campaign. Details on training and guidance with the value function are provided in Section A.4 and Table A2. As an alternative steering method to guidance, we finetune the generative prior with labeled data using an autoregressive language model (ARLM) and DPO, which serves as a baseline. We further compare to a baseline of latent space Bayesian optimization. The strength of steering is tuned by method-specific hyperparameters.

## 4 Results

Table 2: **Summary of datasets used in this work.** Train and test fitness refer to the number of fitness labels used for training and testing the oracle. We focus on TrpB and CreiLOV, with some of the GB1 results moved to the Appendix. While the TrpB dataset has a lot more training labels, it may be more difficult to learn due to relatively high amounts of epistatic effects between residues (non-additivity of mutation effects).

| Dataset | Length | Targeted Residues | Design Space | MSA Size | Train Fitness | Test Fitness | Reference |
|---|---|---|---|---|---|---|---|
| **TrpB** Enzyme Activity | 389 | 117, 118, 119, 162, 166, 182, 183, 184, 185, 186, 227, 228, 230, 231, 301 | *N*=15 | $5.7e4$ | 75,618 | 23,313 | Johnston et al. (2024) |
| **CreiLOV** Fluorescence | 119 | All | *N*=119 | $3.7e5$ | 6,842 | 2,401 | Chen et al. (2023c) |
| **GB1** Binding | 56 | All | *N*=56 | 126 | 3.9e6 | 9.6e4 | Olson et al. (2014) |

We study three proteins, the TrpB enzyme (Johnston et al., 2024), the CreiLOV fluorescent protein (Chen et al., 2023c), and the GB1 binding protein (Olson et al., 2014) due to the availability of fitness data across many residues (Table 2). We focus protein fitness optimization to a design space of 15 residues in TrpB (only these positions are allowed to vary) and all 119 and 56 residues in CreiLOV and GB1, respectively. For each protein's variants, we evaluate fitness by approximating it via a supervised oracle trained on a large amount of real data (Section A.2).

### 4.1 Model pretraining captures the distribution of evolutionarily related protein sequences and enables sampling sequences with high fitness

Based on the methods explained in Section A.1 and A.3, we trained generative priors on natural sequences from the MSA, focusing on continuous diffusion models (*Continuous)*, discrete diffusion models with uniform (*D3PM, UDLM*) and absorbing state noising processes (*MDLM*), and autoregressive language models (*ARLM*) (Table 3). Overall, the trained models capture the natural distribution of protein sequences, with the D3PM models seeming to match the distribution the most

Table 3: **Summary of generative priors evaluated in this work.** Each generative prior was trained on an MSA of homologous natural sequences. All denoising processes were modeled using a transformer architecture (Section A.3). ***Italicized*** models were further explored in downstream guidance experiments.

| Model | Type | Noise | # Params | Notes |
|---|---|---|---|---|
| ***Continuous*** | Continuous Diffusion | Gaussian | 27.9 M | |
| Continuous-ESM | | | 25.5 M | diffusion over ESM embeddings |
| D3PM-Baseline | Discrete Diffusion | Uniform | 37.9 M | |
| ***D3PM*** | | | 37.9 M | finetuned from EvoDiff 38M-Uniform (Alamdari et al., 2024) |
| UDLM | | | 28.6 M | uniform diffusion langauge model |
| ***MDLM*** | Discrete Diffusion | Absorbing | 28.6 M | masked diffusion language model |
| ***ARLM*** | Language Model | n/a | 151 M | autoregressive language model fine-tuned from ProGen2-small (Nijkamp et al., 2023) |

closely while also generating sequences with high diversity (Fig. 4, Fig. A3). The two different diffusion models over continuous space show comparatively lower performance, and diffusing over the latent space of ESM embeddings does not boost performance on this task. The UDLM model has low performance due to mode collapse (Fig. A3, Fig. A5). Future work could finetune the pretrained diffusion protein language model (DPLM) as an MDLM (Wang et al., 2024).

Overall, we found that pretrained priors sample protein variants that have higher mean fitness, which corroborates previous studies finding that sequences with higher evolutionary likelihood are also likely to have higher fitness (Li et al., 2025a; Hie et al., 2023). Based on these results, we proceeded to perform remaining experiments with one model from each category of model type, namely the *Continuous*, *D3PM*, *MDLM*, and *ARLM* models.

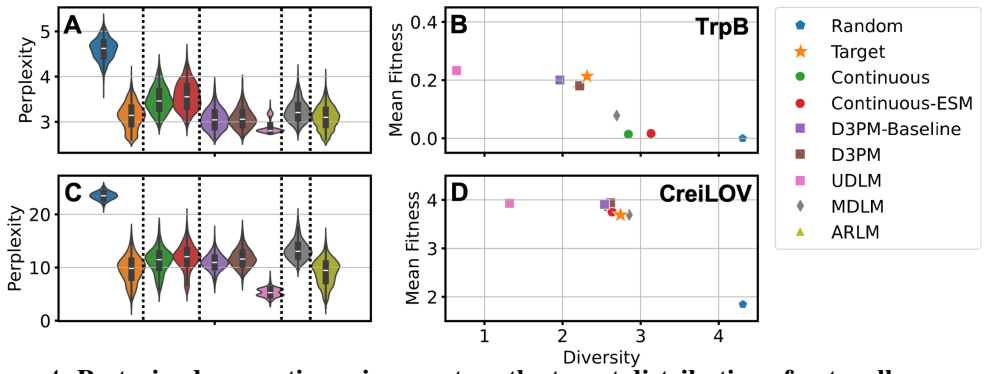


Figure 4: **Pretrained generative priors capture the target distribution of naturally occurring sequences** that are homologous to TrpB (**A-B**) and CreiLOV (**C-D**), respectively. Lower perplexity corresponds to higher likelihood in the model. The diversity of sequences was computed as the average Shannon entropy of mutated positions with mean fitness corresponding to the oracle predictions. While the various models largely achieve comparable performance, the D3PM models capture the target distribution with the highest fidelity, whereas the UDLM model is prone to mode collapse. For each model, 1000 sequences were sampled and repeats were allowed to approximate the distribution. To approximate the target distribution, 1000 sequences were sampled from the MSA used for pretraining. Perplexity was calculated by passing generated sequences through the 764 M parameter ProGen2-base model. More details on model training can be found in Table 3 and Section A.3, and GB1 results are provided in Fig. A4.

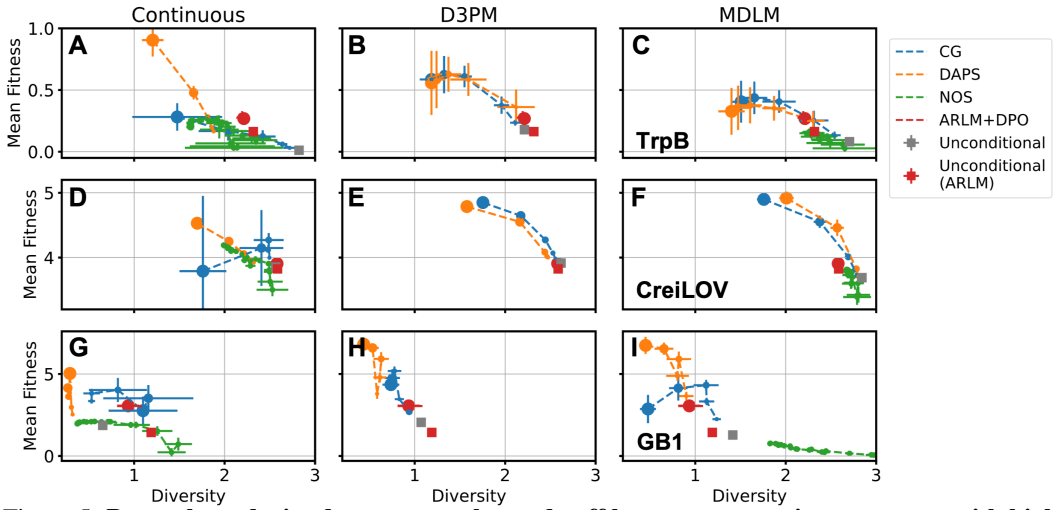


Figure 5: **Pareto boundaries demonstrate the trade-off between generating sequences with high fitness and high diversity** for TrpB (**A-C**), CreiLOV (**D-F**), and GB1 (**G-I**). Sequences sampled from the generative models (Continuous, D3PM, and MDLM), after guidance with labeled fitness data, are enriched in high-fitness protein variants, and most methods show higher performance than the ARLM+DPO baseline. Larger circle indicates a stronger guidance strength hyperparameter (excluding NOS), specified in Table A3. Each experiment was repeated using 10 different standardized sets of 200 unique sequences used for steering, each drawn from the D3PM prior, and error bars show standard deviation. Mean fitness and diversity were calculated based on 200 generated samples, with diversity calculated as the average Shannon entropy of amino acids at mutated positions. Unconditional refers to sequences sampled from the prior with no guidance.

### 4.2 Evaluating SGPO design choices

Impressively, steering with modest amounts of labeled data (200 sequence-fitness pairs) enables most models and methods to generate sequences with even higher fitness, while sacrificing some generation diversity (Fig. 5). In this low data regime, guidance with diffusion models outperforms DPO with language models; the latter does not enable as much steerability. CG and DAPS enable the strongest steerability overall, but DAPS outperforms CG for the continuous models (Fig. 5A, D). In general, guidance seems to work similarly for uniform diffusion (D3PM) and to absorbing state diffusion (MDLM). Overall, the continuous diffusion models do not perform as well as other models, as the prior does not capture the distribution of natural sequences with high fitness as well (Fig. 5A). NOS does not seem to allow for as much steerability, despite an extensive hyperparameter scan (Table A3). Finally, we conducted a closer analysis of the number of unique sequences generated by the steered models and confirmed that most models produce entirely novel sequences, suggesting that they are not over-steering (Fig. A6).

### 4.3 "Thompson sampling" using an ensemble of classifiers is effective for adaptive optimization

Next, we performed adaptive optimization experiments, which mimic real-world protein engineering scenarios and follow a setup similar to batch Bayesian optimization: in each round, a batch of sequences is sampled, evaluated for fitness, and used to retrain a supervised value function that guides sampling from the pretrained prior. We focused on the MDLM models with the CG and DAPS guidance strategies, as these combinations achieved the best performance in our earlier set of experiments (Fig. 5). Based on findings from these previous experiments, we selected the ideal guidance strength hyperparameter to balance fitness and diversity—ensuring high predicted fitness without significantly compromising sequence diversity (Table A3). For both guidance strategies, we employed an algorithm akin to Thompson sampling (Kandasamy et al., 2018; Russo et al., 2018), drawing a different value function from a frequentist ensemble of neural network regressors to guide the generation of each new sample (Yang et al., 2025b). Pseudocode for our adaptive optimization algorithm is provided in Section A.5.

Plug-and-play guidance strategies outperform baselines such as DPO with an ARLM, sampling just from the unconditional generative prior, and latent space Bayesian optimization with APEXGo (Fig. 6): Sampled sequences achieve higher values of mean and maximum fitness. Furthermore, campaigns using an ensemble of value functions and "Thompson sampling" achieve higher maximum fitness than those using only a single value function for guidance (Table A4), which may be because these models enable more exploration of sequence space (Fig. A7). However, it is difficult to ascertain wither CG or DAPS works better as a guidance strategy, as the performance is highly dependent on the guidance strength hyperparameter, and the optimal hyperparameter will not typically be known in a real-world campaign. Because the oracle may not capture the true nature of the protein fitness landscape, we also suggest making relative comparisons here rather than absolute comparisons between model performance.

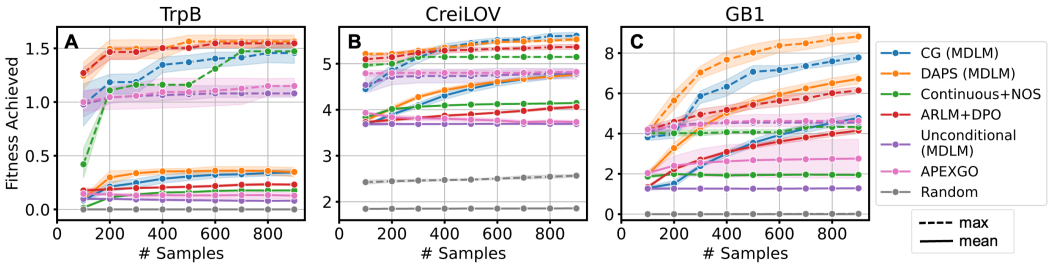


Figure 6: **Maximum/mean fitness achieved improves over multiple iterations of steering in an adaptive setting similar to batch Bayesian optimization** for TrpB (**A**), CreiLOV (**B**), and GB1 (**C**). 100 sequences were sampled in each round. Within each round, an ensemble of 10 value functions (classifiers) was trained on fitness data from all previously queried samples, and each new sample was generated by the MDLM model guided with a value function sampled from the ensemble (akin to Thompson sampling). Only unique, novel samples were acquired. Guidance strength parameter is provided in Table A3. Error bars show standard deviation between 5 different random initializations.

## 5 Discussion

In this work, we conduct a comprehensive study of SGPO methods and demonstrate that it is an effective approach for protein fitness optimization, by capturing the distribution of natural protein sequences with a generative prior and then steering the generations with labeled data. We find that DAPS with discrete diffusion models has the highest performance overall, and plug-and-play guidance-based strategies are generally more effective than finetuning language models; the latter can be difficult when only few fitness labels are available. SGPO approaches also outperform latent space Bayesian optimization (namely APEXGo), which we attribute to the difficulty in calibrating the trust region in very low-data regimes with limited rounds of optimization and the fact that latent space Bayesian optimization relies heavily on the structure of the latent space learned during generation model training, which can limit extrapolation to high-fitness but unnatural variants.

Using plug-and-play guidance approaches has other advantages. First, only one hyperparameter (guidance strength) needs to be tuned. In real-world engineering scenarios, even in the absence of ground truth fitness labels, one practical approach to selecting the guidance strength is to scan over values and choose the highest setting for which $n$ generated sequences remain unique and novel relative to previously measured sequences, where $n$ corresponds to the screening throughput available for the next round. By contrast, for DPO, various hyperparameters need to be tuned, and the training process has to be monitored closely. Even for NOS, different parameters such as the step size, the number of steps, and the stability coefficient must be tuned together. A further advantage of guidance is the low computational cost required, as the prior model weights are not updated during guidance. Pretraining/finetuning to obtain each initial prior was achieved on a single H100 GPU in less than one hour while each individual guidance experiment took minutes; pretraining language models took several hours on a single GPU.

Still, there are certain limitations of our work. We focused on proteins with fitness as mostly native function, but it would be interesting to test SGPO on other protein fitness optimization tasks where the pretrained prior may not provide as much utility. We also focused on protein optimization where

only $\approx 10^2$ fitness labels were available; different methods, such as RL, may perform better for applications where larger amounts of fitness data are available (Hie & Yang, 2022; Blalock et al., 2025). We focused on guidance strategies and did not test DPO or model finetuning-based methods with discrete diffusion models, but future work could adapt these methods for discrete diffusion (Borso et al., 2025). Furthermore, for TrpB and for language models, we manually mapped sequences back into the design space after generation (Section A.2), but explicitly building this into sampling techniques, such as inpainting in masked models (Blalock et al., 2025; Goel et al., 2025) may lead to improved performance. We did not consider insertions or deletions, but variable-length sequence generation could be considered in the future. Finally, we did not directly compare to existing approaches for protein engineering such as directed evolution for reasons explained in Section A.2.

There are several promising directions for future work to improve and extend SGPO methods. For instance, we experimented with guiding generation using value functions sampled from a Gaussian process posterior, enabling principled Thompson sampling from a fully Bayesian perspective. However, the Gaussian process struggled to model high-dimensional protein representations, leading to poor performance. This limitation could potentially be addressed with better kernel choices (Wilson et al., 2016; Michael et al., 2024; Yang et al., 2025b). Recent work has also begun to incorporate multi-objective optimization (Annadani et al., 2025; Tang et al., 2025a; Li et al., 2024b; Chen et al., 2025) and uncertainty quantification (Wu et al., 2025) when guiding diffusion models. Simultaneously, alternatives to acquisition-function-based approaches are being developed to enable Bayesian optimization in large design spaces where enumeration is infeasible (Bal et al., 2025). Other emerging approaches—closer in spirit to flow matching—are being proposed for discrete data and may offer new opportunities for exploration (Davis et al., 2024; Stark et al., 2024; Tang et al., 2025b). Finally, for masked diffusion models, strategies such as remasking or scheduling could be explored to improve inference, particularly to enhance model amenability to guidance (Wang et al., 2025b; Peng et al., 2025; Liu et al., 2025; Amin et al., 2025a). It will also be interesting to further explore guidance in other discrete domains such as natural language and small molecules (Schiff et al., 2024).

In summary, guiding generative models with labeled data offers a powerful, flexible, and principled framework for protein fitness optimization, as it effectively leverages both the evolutionary information encoded in natural protein sequences and task-specific fitness objectives. At the same time, we recognize the potential dual-use risks: such methods could, in principle, be misused to design harmful proteins, underscoring the importance of appropriate safeguards (Baker & Church, 2024; Wittmann et al., 2024). In short, our work has examined multiple effective SGPO strategies and offered insights on best-practices for real-world protein fitness optimization, laying the groundwork for further exploration and wet-lab validation.

**Acknowledgments**

This work was supported by a U.S. Army Research Office cooperative agreement (W911NF-19-2-0026 to F.H.A.) and an Amgen Chem-Bio Engineering award. J.Y. is also supported by the NSF Graduate Research Fellowship Program and the Google PhD Fellowship. We would like to thank Hunter Nisonoff, Jacob Gershon, Lucas Arnoldt, and Chenghao Liu for helpful discussions and Francesca-Zhoufan Li for help with the TrpB dataset. We would also like to thank Nate Gruver for help with the NOS implementation, Filippo Stocco for help with the DPO implementation, and Nathaniel Blalock for guidance on how to use the CreiLOV dataset.

## References

Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Neil Tenenholtz, Bob Strome, Alan M. Moses, Alex Xijie Lu, Nicolò Fusi, Ava Pardis Amini, and Kevin K. Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, November 2024. doi: 10.1101/2023.09.11.556673. URL `https://www.biorxiv.org/content/10.1101/2023.09.11.556673v2`. Preprint; version 2 posted Nov 4, 2024.

Alan N. Amin, Nate Gruver, and Andrew Gordon Wilson. Why masking diffusion works: Condition on the jump schedule for improved discrete diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a. doi: 10.48550/arXiv.2506.08316. URL `https://neurips.cc/virtual/2025/poster/115376`. NeurIPS 2025 (poster), to appear.

Alan Nawzad Amin, Nate Gruver, Yilun Kuang, Yucen Lily Li, Hunter Elliott, Calvin McCarter, Aniruddh Raghu, Peyton Greenside, and Andrew Gordon Wilson. Bayesian optimization of antibodies informed by a generative model of evolving sequences. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, Singapore, April 2025b. doi: 10.48550/arXiv.2412.07763. URL https://openreview.net/forum?id=E48QvQppIN. Spotlight.

Christof Angermueller, David Dohan, David Belanger, Ramya Deshpande, Kevin Murphy, and Lucy Colwell. Model-based reinforcement learning for biological sequence design. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=HklxbgBKvr. ICLR 2020.

Yashas Annadani, Syrine Belakaria, Stefano Ermon, Stefan Bauer, and Barbara E. Engelhardt. Preference-guided diffusion for multi-objective offline optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. doi: 10.48550/arXiv.2503.17299. URL https://neurips.cc/virtual/2025/poster/116185. Poster; to appear.

Frances H. Arnold. Directed evolution: Bringing new chemistry to life. *Angewandte Chemie International Edition*, 57(16):4143–4148, April 2018. doi: 10.1002/anie.201708408. URL https://onlinelibrary.wiley.com/doi/10.1002/anie.201708408.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/958c530554f78bcd8e97125b70e6973d-Abstract.html. NeurIPS 2021.

David Baker and George Church. Protein design meets biosecurity. *Science*, 383(6681):349, January 2024. doi: 10.1126/science.ado1671. URL https://www.science.org/doi/10.1126/science.ado1671.

Melis Ilayda Bal, Pier Giuseppe Sessa, Mojmir Mutny, and Andreas Krause. Optimistic games for combinatorial bayesian optimization with application to protein design. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, Singapore, April 2025. doi: 10.48550/arXiv.2409.18582. URL https://openreview.net/forum?id=xiyzCfXTS6. Poster.

Brian R. Bartoldson, Siddarth Venkatraman, James Diffenderfer, Moksh Jain, Tal Ben-Nun, Seanie Lee, Minsu Kim, Johan Obando-Ceron, Yoshua Bengio, and Bhavya Kailkhura. Trajectory balance with asynchrony: Decoupling exploration and learning for fast, scalable llm post-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. doi: 10.48550/arXiv.2503.18929. URL https://neurips.cc/virtual/2025/poster/117641. Poster; to appear.

Nathaniel Blalock, Srinath Seshadri, Agrim Babbar, Sarah A. Fahlberg, Ameya Kulkarni, and Philip A. Romero. Functional alignment of protein language models via reinforcement learning. *bioRxiv*, May 2025. doi: 10.1101/2025.05.02.651993. URL https://www.biorxiv.org/content/10.1101/2025.05.02.651993v1. Preprint.

Umberto Borso, Davide Paglieri, Jude Wells, and Tim Rocktäschel. Preference-based alignment of discrete diffusion models. In *ICLR 2025 Workshop on Bidirectional Human–AI Alignment (Bi-Align)*, Singapore, April 2025. doi: 10.48550/arXiv.2503.08295. URL https://openreview.net/pdf?id=qs9CTsC32h. Workshop paper.

David H. Brookes and Jennifer Listgarten. Design by adaptive sampling, February 2020. URL http://arxiv.org/abs/1810.03714. arXiv:1810.03714 [cs, q-bio, stat].

David H Brookes, Hahnbeom Park, and Jennifer Listgarten. Conditioning by adaptive sampling for robust design. In *International Conference on Machine Learning*, 2019. URL https://arxiv.org/abs/1901.10060.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pp. 28266–28279, New Orleans, LA, USA, December 2022. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b5b528767aa35f5b1a60fe0aaeca0563-Abstract-Conference.html. NeurIPS 2022.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5453–5512. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/campbell24a.html.

Egbert Castro, Abhinav Godavarthi, Julian Rubinfien, Kevin B. Givechian, Dhananjay Bhaskar, and Smita Krishnaswamy. Transformer-based protein generation with regularized latent space optimization. *Nature Machine Intelligence*, 4(10):840–851, October 2022. doi: 10.1038/s42256-022-00532-1. URL https://www.nature.com/articles/s42256-022-00532-1.

Tianlai Chen, Pranay Vure, Rishab Pulugurta, and Pranam Chatterjee. Amp-diffusion: Integrating latent diffusion with protein language models for antimicrobial peptide generation. In *NeurIPS 2023 Workshop on Generative AI and Biology (GenBio)*, New Orleans, LA, USA, December 2023a. URL https://openreview.net/forum?id=145TM9VQhx. Workshop poster; non-archival.

Ting Chen, Ruixiang Zhang, and Geoffrey E. Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. In *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, May 2023b. URL https://openreview.net/forum?id=3itjR9QxFw. ICLR 2023.

Tong Chen, Yinuo Zhang, and Pranam Chatterjee. Areuredi: Annealed rectified updates for refining discrete flows with multi-objective guidance, 2025. URL https://arxiv.org/abs/2510.00352.

Yongcan Chen, Ruyun Hu, Keyi Li, Yating Zhang, Lihao Fu, Jianzhi Zhang, and Tong Si. Deep mutational scanning of an oxygen-independent fluorescent protein creilov for comprehensive profiling of mutational and epistatic effects. *ACS Synthetic Biology*, 12(5):1461–1473, May 2023c. doi: 10.1021/acssynbio.2c00662. URL https://pubs.acs.org/doi/10.1021/acssynbio.2c00662.

Wenda Chu, Zihui Wu, Yifan Chen, Yang Song, and Yisong Yue. Split gibbs discrete diffusion posterior sampling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. doi: 10.48550/arXiv.2503.01161. URL https://neurips.cc/virtual/2025/poster/117795. Poster; to appear.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=OnD9zGAGT0k.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *Proceedings of the 8th International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=H1edEyBKDS. ICLR 2020.

Oscar Davis, Samuel Kessler, Mircea Petrache, Ismail Ilkan Ceylan, Michael Bronstein, and Avishek Joey Bose. Fisher flow matching for generative modeling over discrete data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/fadec8f2e65f181d777507d1df69b92f-Abstract-Conference.html.

Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin, Pierre H. Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, Curtis Hawthorne, Rémi Leblond, Will Grathwohl, and Jonas Adler. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, November 2022. doi: 10.48550/arXiv.2211.15089. URL https://arxiv.org/abs/2211.15089. Preprint.

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

Hongyuan Fei, Yunjia Li, Yijing Liu, Jingjing Wei, Aojie Chen, and Caixia Gao. Advancing protein evolution with inverse folding models integrating structural and evolutionary constraints. *Cell*, 188(17):4674–4692.e19, August 2025. doi: 10.1016/j.cell.2025.06.014. URL https://www.cell.com/cell/abstract/S0092-8674(25)00680-4.

Peter I. Frazier. Bayesian optimization. In *Recent Advances in Optimization and Modeling of Contemporary Problems*, INFORMS TutORials in Operations Research, pp. 255–278. INFORMS, Catonsville, MD, 2018. doi: 10.1287/educ.2018.0188. URL https://pubsonline.informs.org/doi/10.1287/educ.2018.0188.

Chase R Freschlin, Sarah A Fahlberg, and Philip A Romero. Machine learning to navigate fitness landscapes for protein engineering. *Current Opinion in Biotechnology*, 75:102713, June 2022. ISSN 0958-1669. doi: 10.1016/j.copbio.2022.102713. URL https://www.sciencedirect.com/science/article/pii/S0958166922000465.

Shrey Goel, Vishrut Thoutam, Edgar Mariano Marroquin, Aaron Gokaslan, Arash Firouzbakht, Sophia Vincoff, Volodymyr Kuleshov, Huong T. Kratochvil, and Pranam Chatterjee. Memdlm: De novo membrane protein design with property-guided discrete diffusion. In *ICLR 2025 Workshop on Learning Meaningful Representations of Life (LMRL)*, Singapore, April 2025. URL https://iclr.cc/virtual/2025/35945. Workshop poster; non-archival.

Rafael Gómez-Bombarelli, Jennifer N. Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, February 2018. doi: 10.1021/acscentsci.7b00572. URL https://pubs.acs.org/doi/10.1021/acscentsci.7b00572.

Nate Gruver, Samuel Stanton, Nathan C. Frey, Tim G. J. Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew Gordon Wilson. Protein design with guided discrete diffusion. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/29591f355702c3f4436991335784b503-Abstract-Conference.html. NeurIPS 2023.

Anvita Gupta and James Zou. Feedback GAN for DNA optimizes protein functions. *Nature Machine Intelligence*, 1(2):105–111, February 2019. ISSN 2522-5839. doi: 10.1038/s42256-019-0017-4. URL https://www.nature.com/articles/s42256-019-0017-4. Number: 2 Publisher: Nature Publishing Group.

Alex Hawkins-Hooker, Jakub Kmec, Oliver Bent, and Paul Duckworth. Likelihood-based fine-tuning of protein language models for few-shot fitness prediction and design. In *ICML 2024 Workshop on Machine Learning for Life and Material Science: From Theory to Industry Applications (ML4LMS)*, Vienna, Austria, July 2024. doi: 10.1101/2024.05.28.596156. URL https://openreview.net/forum?id=MkYhOEUJyi. Workshop poster; non-archival.

Brian L. Hie and Kevin K. Yang. Adaptive machine learning for protein engineering. *Current Opinion in Structural Biology*, 72:145–152, February 2022. ISSN 0959440X. doi: 10.1016/j.sbi.2021.11.002. URL https://linkinghub.elsevier.com/retrieve/pii/S0959440X21001457.

Brian L. Hie, Varun R. Shanker, Duo Xu, Theodora U. J. Bruun, Payton A. Weidenbacher, Shaogeng Tang, Wesley Wu, John E. Pak, and Peter S. Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, April 2023. ISSN 1546-1696. doi: 10.1038/s41587-023-01763-2. URL https://doi.org/10.1038/s41587-023-01763-2.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html. NeurIPS 2020.

Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *Proceedings of the Tenth International Conference on Learning Representations (ICLR)*, April 2022. URL https://openreview.net/forum?id=Lm8T39vLDTE. ICLR 2022 (poster).

Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 40(7):1114–1122, January 2022. ISSN 1546-1696. doi: 10.1038/s41587-021-01146-5. URL https://www.nature.com/articles/s41587-021-01146-5. Bandiera_abtest: a Cg_type: Nature Research Journals Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Machine learning;Protein design Subject_term_id: machine-learning;protein-design.

Chloe Hsu, Clara Fannjiang, and Jennifer Listgarten. Generative models for protein structures and sequences. *Nature Biotechnology*, 42:196–199, 2024.

Long-Kai Huang, Rongyi Zhu, Bing He, and Jianhua Yao. Steering protein language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26247–26260. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/huang25ba.html.

Moksh Jain, Emmanuel Bengio, Alex Hernandez-Garcia, Jarrid Rector-Brooks, Bonaventure F. P. Dossou, Chanakya Ekbote, Jie Fu, Tianyu Zhang, Michael Kilgour, Dinghuai Zhang, Lena Simine, Payel Das, and Yoshua Bengio. Biological sequence design with GFlowNets. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 9786–9801. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/jain22a.html.

Kaiyi Jiang, Zhaoqing Yan, Matteo Di Bernardo, Samantha R. Sgrizzi, Lukas Villiger, Alisan Kayabolen, B. J. Kim, Josephine K. Carscadden, Masahiro Hiraizumi, Hiroshi Nishimasu, Jonathan S. Gootenberg, and Omar O. Abudayyeh. Rapid in silico directed evolution by a protein language model with EVOLVEpro. *Science*, 387(6732):eadr6006, November 2024. doi: 10.1126/science.adr6006. URL https://www.science.org/doi/10.1126/science.adr6006. Publisher: American Association for the Advancement of Science.

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

Kadina E. Johnston, Patrick J. Almhjell, Ella J. Watkins-Dulaney, Grace Liu, Nicholas J. Porter, Jason Yang, and Frances H. Arnold. A combinatorially complete epistatic fitness landscape in an enzyme active site. *Proceedings of the National Academy of Sciences*, 121(32):e2400439121, August 2024. doi: 10.1073/pnas.2400439121. URL https://www.pnas.org/doi/10.1073/pnas.2400439121.

Kirthevasan Kandasamy, Akshay Krishnamurthy, Jeff Schneider, and Barnabás Póczos. Parallelised bayesian optimisation via thompson sampling. In *International conference on artificial intelligence and statistics*, pp. 133–142. PMLR, 2018.

Kazutaka Katoh and Daron M. Standley. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution*, 30(4):772–780, April 2013. ISSN 0737-4038. doi: 10.1093/molbev/mst010. URL https://doi.org/10.1093/molbev/mst010.

Hyeonah Kim, Minsu Kim, Taeyoung Yun, Sanghyeok Choi, Emmanuel Bengio, Alex Hernández-García, and Jinkyoo Park. Improved off-policy reinforcement learning in biological sequence design. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, Vancouver, Canada, July 2025. doi: 10.48550/arXiv.2410.04461. URL https://icml.cc/virtual/2025/poster/46683. ICML 2025 (poster), to appear.

Andrew Kirjner, Jason Yim, Raman Samusevich, Shahar Bracha, Tommi Jaakkola, Regina Barzilay, and Ila Fiete. Improving protein optimization with smoothed fitness landscapes. In *Proceedings of the 12th International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.00494.

Leo Klarner, Tim G. J. Rudner, Garrett M. Morris, Charlotte M. Deane, and Yee Whye Teh. Context-guided diffusion for out-of-distribution molecular and protein design. In *Proceedings of the 41st*

*International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 24770–24807. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/klarner24a.html.

Théophile Lambert, Amin Tavakoli, Gautham Dharuman, Jason Yang, Vignesh Bhethanabotla, Sukhvinder Kaur, Matthew Hill, Arvind Ramanathan, Anima Anandkumar, and Frances H. Arnold. Sequence-based generative AI-guided design of versatile tryptophan synthases. *bioRxiv*, August 2025. doi: 10.1101/2025.08.30.673177. URL https://www.biorxiv.org/content/10.1101/2025.08.30.673177v1. Preprint; version 1 posted Aug 30, 2025.

Cheuk Kit Lee, Paul Jeha, Jes Frellsen, Pietro Liò, Michael Samuel Albergo, and Francisco Vargas. Debiasing guidance for discrete diffusion with sequential monte carlo. In *ICLR 2025 Workshop on Frontiers in Probabilistic Inference: Learning Meets Sampling*, Singapore, April 2025a. doi: 10.48550/arXiv.2502.06079. URL https://iclr.cc/virtual/2025/workshop/23990. Oral; non-archival.

Seunghun Lee, Jinyoung Park, Jaewon Chu, Minseo Yoon, and Hyunwoo J. Kim. Latent bayesian optimization via autoregressive normalizing flows. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, Singapore, April 2025b. URL https://openreview.net/forum?id=ZCOwwRAaEl. ICLR 2025 (Oral).

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

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*, August 2025b. doi: 10.48550/arXiv.2508.10875. URL https://arxiv.org/abs/2508.10875. Preprint.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM improves controllable text generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/1be5bc25d50895ee656b8c2d9eb89d6a-Abstract-Conference.html. NeurIPS 2022.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gökcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. In *NeurIPS 2024 Workshop on AI for New Drug Modalities*, New Orleans, LA, USA, December 2024a. URL https://neurips.cc/virtual/2024/102888. Workshop poster; non-archival.

Zihao Li, Hui Yuan, Kaixuan Huang, Chengzhuo Ni, Yinyu Ye, Minshuo Chen, and Mengdi Wang. Diffusion model for data-driven black-box optimization. *arXiv preprint arXiv:2403.13219*, March 2024b. doi: 10.48550/arXiv.2403.13219. URL https://arxiv.org/abs/2403.13219. Preprint.

Hocheol Lim, Geon-Ho Lee, and Kyoung Tai No. Scoring-assisted generative exploration for proteins (sage-prot): A framework for multi-objective protein optimization via iterative sequence generation and evaluation. *arXiv preprint arXiv:2505.01277*, May 2025. doi: 10.48550/arXiv.2505.01277. URL https://arxiv.org/abs/2505.01277. Preprint.

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

Sidney Lyayuga Lisanza, Jacob Merle Gershon, Samuel W. K. Tipps, Jeremiah Nelson Sims, Lucas Arnoldt, Samuel J. Hendel, Miriam K. Simma, Ge Liu, Muna Yase, Hongwei Wu, Claire D. Tharp, Xinting Li, Alex Kang, Evans Brackenbrough, Asim K. Bera, Stacey Gerben, Bruce J. Wittmann, Andrew C. McShan, and David Baker. Multistate and functional protein design using RoseTTAFold

sequence space diffusion. *Nature Biotechnology*, 43(8):1288–1298, August 2025. doi: 10.1038/s41587-024-02395-w. URL https://www.nature.com/articles/s41587-024-02395-w. Epub 2024-09-25; Publisher Correction: Nat Biotechnol 43(8):1384, doi:10.1038/s41587-024-02456-0.

Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. Think while you generate: Discrete diffusion with planned denoising. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, Singapore, April 2025. URL https://openreview.net/forum?id=MJNywBdSDy. Poster.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32819–32848, Vienna, Austria, 21–27 Jul 2024. PMLR. URL https://proceedings.mlr.press/v235/lou24a.html.

Morteza Mardani, Jiaming Song, Jan Kautz, and Arash Vahdat. A variational perspective on solving inverse problems with diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=1YO4EE3SPB.

Natalie Maus, Haydn Jones, Juston Moore, Matt J Kusner, John Bradshaw, and Jacob Gardner. Local latent space bayesian optimization over structured inputs. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 34505–34518. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/ded98d28f82342a39f371c013dfb3058-Paper-Conference.pdf.

Viacheslav Meshchaninov, Pavel Strashnov, Andrey Shevtsov, Fedor Nikolaev, Nikita Ivanisenko, Olga Kardymon, and Dmitry Vetrov. Diffusion on language model encodings for protein sequence generation. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, Vancouver, Canada, July 2025. doi: 10.48550/arXiv.2403.03726. URL https://icml.cc/virtual/2025/poster/43588. Poster; to appear.

Richard Michael, Jacob Kæstel-Hansen, Peter Mørch Groth, Simon Bartels, Jesper Salomon, Pengfei Tian, Nikos S. Hatzakis, and Wouter Boomsma. A systematic analysis of regression models for protein engineering. *PLOS Computational Biology*, 20(5):e1012061, May 2024. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1012061. URL https://dx.plos.org/10.1371/journal.pcbi.1012061.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, San Diego, CA, USA, December 2025. URL https://neurips.cc/virtual/2025/poster/118608.

Erik Nijkamp, Jeffrey A. Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978.e3, November 2023. ISSN 24054712. doi: 10.1016/j.cels.2023.10.002. URL https://linkinghub.elsevier.com/retrieve/pii/S2405471223002727.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, Singapore, April 2025. URL https://openreview.net/forum?id=XsgHl54yO7. ICLR 2025 — Poster.

C. Anders Olson, Nicholas C. Wu, and Ren Sun. A Comprehensive Biophysical Description of Pairwise Epistasis throughout an Entire Protein Domain. *Current Biology*, 24(22):2643–2651, November 2014. ISSN 09609822. doi: 10.1016/j.cub.2014.09.072. URL https://linkinghub.elsevier.com/retrieve/pii/S0960982214012688.

Michael S. Packer and David R. Liu. Methods for the directed evolution of proteins. *Nature Reviews Genetics*, 16(7):379–394, July 2015. ISSN 1471-0056, 1471-0064. doi: 10.1038/nrg3927. URL http://www.nature.com/articles/nrg3927.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Jarrid Rector-Brooks, Sherwood Yao, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. In *ICLR 2025 DeLTa Workshop*, Singapore, April 2025. doi: 10.48550/arXiv.2502.03540. URL https://openreview.net/forum?id=fFuVPKpSt0. Workshop poster; non-archival.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html. NeurIPS 2023.

Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Zachary Quinn, Chenghao Liu, Sarthak Mittal, Nouha Dziri, Michael Bronstein, Yoshua Bengio, Pranam Chatterjee, Alexander Tong, and Avishek Joey Bose. Steering masked discrete diffusion models via discrete denoising posterior prediction. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR)*, Singapore, April 2025. doi: 10.48550/arXiv.2410.08134. URL https://openreview.net/forum?id=Ombm8S40zN. Poster.

Zhizhou Ren, Jiahan Li, Fan Ding, Yuan Zhou, Jianzhu Ma, and Jian Peng. Proximal exploration for model-guided protein sequence design. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18520–18536. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/ren22a.html.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), April 2021. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2016239118. URL https://www.pnas.org/content/118/15/e2016239118. Publisher: National Academy of Sciences Section: Biological Sciences.

Philip A Romero and Frances H Arnold. Exploring protein fitness landscapes by directed evolution. *Nat Rev Mol Cell Biol*, 10:866–876, 2009. doi: 10.1038/nrm2805.

Jeffrey A Ruffolo and Ali Madani. Designing proteins with language models. *Nature Biotechnology*, 42:200–202, February 2024.

Daniel J Russo, Benjamin Van Roy, Abbas Kazerouni, Ian Osband, Zheng Wen, et al. A tutorial on thompson sampling. *Foundations and Trends® in Machine Learning*, 11(1):1–96, 2018.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T. Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37 of *Advances in Neural Information Processing Systems*, Vancouver, Canada, December 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/eb0b13cc515724ab8015bc978fdde0ad-Abstract-Conference.html.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P. de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple Guidance Mechanisms for Discrete Diffusion Models, December 2024. URL http://arxiv.org/abs/2412.10193. arXiv:2412.10193 [cs].

Kosuke Seki, Amy B. Guo, Deniz Akpinaroglu, and Tanja Kortemme. A combinatorial mutational map of active non-native protein kinases by deep learning guided sequence design. *bioRxiv*, pp. 2025.08.03.668353, August 2025. doi: 10.1101/2025.08.03.668353. URL https://www.biorxiv.org/content/10.1101/2025.08.03.668353. Preprint; version 1 posted Aug 3, 2025.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. 2024. URL https://arxiv.org/abs/2406.04329.

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

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, Vancouver, Canada, July 2025. doi: 10.48550/arXiv.2501.06848. URL https://icml.cc/virtual/2025/poster/45673. ICML 2025 (poster).

Diogo Soares, Leon Hetzel, Paulina Szymczak, Fabian Theis, Stephan Günnemann, and Ewa Szczurek. Targeted AMP generation through controlled diffusion with efficient embeddings, April 2025. URL https://arxiv.org/abs/2504.17247. Preprint.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*, May 2021. URL https://openreview.net/forum?id=PxTIG12RRHS. ICLR 2021 — Outstanding Paper Award.

Zhenqiao Song and Lei Li. Importance weighted expectation-maximization for protein sequence design. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32349–32364, Honolulu, Hawaii, USA, Jul 2023. PMLR. URL https://proceedings.mlr.press/v202/song23g.html. ICML 2023.

Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating Bayesian Optimization for Biological Sequence Design with Denoising Autoencoders. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20459–20478. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/stanton22a.html.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to DNA sequence design. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46495–46513, Vienna, Austria, 21–27 Jul 2024. PMLR. URL https://proceedings.mlr.press/v235/stark24b.html.

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

Filippo Stocco, Maria Artigues-Lleixà, Andrea Hunklinger, Talal Widatalla, Marc Güell, and Noelia Ferruz. Guiding generative protein language models with reinforcement learning. In *NeurIPS 2024 Workshop on Machine Learning in Structural Biology (MLSB)*, Vancouver, Canada, December 2024. doi: 10.48550/arXiv.2412.12979. URL https://slideslive.com/39031185/guiding-protein-language-models-with-reinforcement-learning. Workshop presentation (non-archival).

Kiera H. Sumida, Reyes Núñez-Franco, Indrek Kalvet, Samuel J. Pellock, Basile I. M. Wicky, Lukas F. Milles, Justas Dauparas, Jue Wang, Yakov Kipnis, Noel Jameson, Alex Kang, Joshmyn De La Cruz, Banumathi Sankaran, Asim K. Bera, Gonzalo Jiménez-Osés, and David Baker. Improving Protein Expression, Stability, and Function with ProteinMPNN. *Journal of the American Chemical Society*, 146(3):2054–2061, January 2024. ISSN 0002-7863, 1520-5126. doi: 10.1021/jacs.3c10941. URL https://pubs.acs.org/doi/10.1021/jacs.3c10941.

Haoran Sun, Liang He, Pan Deng, Guoqing Liu, Zhiyu Zhao, Yuliang Jiang, Chuan Cao, Fusong Ju, Lijun Wu, Haiguang Liu, Tao Qin, and Tie-Yan Liu. Accelerating protein engineering with fitness landscape modelling and reinforcement learning. *Nature Machine Intelligence*, September 2025. ISSN 2522-5839. doi: 10.1038/s42256-025-01103-w. URL https://doi.org/10.1038/s42256-025-01103-w.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. Peptune: De novo generation of therapeutic peptides with multi-objective-guided discrete diffusion. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, Vancouver, Canada, July 2025a. doi: 10.48550/arXiv.2412.17780. URL https://icml.cc/virtual/2025/poster/45889. Poster; ICML 2025.

Sophia Tang, Yinuo Zhang, Alexander Tong, and Pranam Chatterjee. Gumbel-softmax flow matching with straight-through guidance for controllable biological sequence generation. In *ICLR 2025 Workshop on Integrating Generative and Experimental Platforms for Biomolecular Design (GEM)*, 2025b. URL https://arxiv.org/abs/2503.17361. Workshop poster (non-archival); also available as arXiv:2503.17361.

Neil Thomas, David Belanger, Chenling Xu, Hanson Lee, Kathleen Hirano, Kosuke Iwai, Vanja Polic, Kendra D. Nyberg, Kevin G. Hoff, Lucas Frenz, Charlie A. Emrich, Jun W. Kim, Mariya Chavarha, Abi Ramanan, Jeremy J. Agresti, and Lucy J. Colwell. Engineering highly active nuclease enzymes with machine learning and high-throughput screening. *Cell Systems*, 16(3):101236, March 2025. ISSN 2405-4712. doi: 10.1016/j.cels.2025.101236. URL https://www.sciencedirect.com/science/article/pii/S2405471225000699.

Marcelo D. T. Torres, Yimeng Zeng, Fangping Wan, Natalie Maus, Jacob Gardner, and Cesar de la Fuente-Nunez. A generative artificial intelligence approach for antibiotic optimization. *bioRxiv*, November 2024. doi: 10.1101/2024.11.27.625757. URL https://www.biorxiv.org/content/10.1101/2024.11.27.625757v1. preprint.

Marcelo D. T. Torres, Tianlai Chen, Fangping Wan, Pranam Chatterjee, and Cesar de la Fuente. Generative latent diffusion language modeling yields anti-infective synthetic peptides. *Cell Biomaterials*, 1:100183, October 2025. doi: 10.1016/j.celbio.2025.100183. URL https://www.cell.com/cell-biomaterials/fulltext/S3050-5623(25)00174-6. Published version of bioRxiv preprint 10.1101/2025.01.31.636003; online early Sept 2, 2025.

Masatoshi Uehara, Xingyu Su, Yulai Zhao, Xiner Li, Aviv Regev, Shuiwang Ji, Sergey Levine, and Tommaso Biancalani. Reward-guided iterative refinement in diffusion models at test-time with applications to protein and dna design. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, Vancouver, Canada, July 2025. PMLR. URL https://icml.cc/virtual/2025/poster/46204. ICML 2025 (poster).

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, Alexandre Adam, Jarrid Rector-Brooks, Yoshua Bengio, Glen Berseth, and Nikolay Malkin. Amortizing intractable inference in diffusion models for vision, language, and control. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, Vancouver, Canada, 2024. URL https://neurips.cc/virtual/2024/poster/95348. NeurIPS 2024.

Siddarth Venkatraman, Mohsin Hasan, Minsu Kim, Luca Scimeca, Marcin Sendera, Yoshua Bengio, Glen Berseth, and Nikolay Malkin. Outsourced diffusion sampling: Efficient posterior inference in latent spaces of generative models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 1–28, Vancouver, BC, Canada, July 2025. PMLR. doi: 10.48550/arXiv.2502.06999. URL https://openreview.net/forum?id=94c9hu6Fsv. ICML 2025 (poster). Also available as arXiv:2502.06999.

Tobias Vornholt, Mojmír Mutný, Gregor W. Schmidt, Christian Schellhaas, Ryo Tachibana, Sven Panke, Thomas R. Ward, Andreas Krause, and Markus Jeschek. Enhanced Sequence-Activity Mapping and Evolution of Artificial Metalloenzymes by Active Learning. *ACS Central Science*, 10 (7):1357–1370, May 2024. ISSN 2374-7943, 2374-7951. URL https://pubs.acs.org/doi/10.1021/acscentsci.4c00258.

Chenyu Wang, Masatoshi Uehara, Yichun He, Amy Wang, Tommaso Biancalani, Avantika Lal, Tommi Jaakkola, Sergey Levine, Hanchen Wang, and Aviv Regev. Fine-tuning discrete diffusion models via reward optimization with applications to dna and protein design. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR 2025)*, Singapore, April 2025a. URL https://openreview.net/forum?id=G328D1xt4W. ICLR 2025 Poster.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*, December 2025b. URL https://neurips.cc/virtual/2025/poster/118818. Poster.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 52309–52333, Vienna, Austria, Jul 2024. PMLR. URL https://proceedings.mlr.press/v235/wang24ct.html.

Ziwen Wang, Jiajun Fan, Ruihan Guo, Thao Nguyen, Heng Ji, and Ge Liu. Proteinzero: Self-improving protein generation via online reinforcement learning, jun 2025c. URL https://arxiv.org/abs/2506.07459. arXiv preprint.

Talal Widatalla, Rafael Rafailov, and Brian Hie. Aligning protein generative models with experimental fitness via Direct Preference Optimization. *bioRxiv*, 2024. doi: 10.1101/2024.05.20.595026. URL https://doi.org/10.1101/2024.05.20.595026.

Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 370–378, Cádiz, Spain, 2016. PMLR. URL https://proceedings.mlr.press/v51/.

Bruce Wittmann, Tessa Alexanian, Craig Bartling, Jacob Beal, Adam Clore, James Diggans, Kevin Flyangolts, Bryan T. Gemler, Tom Mitchell, Steven T. Murphy, Nicole E. Wheeler, and Eric Horvitz. Toward AI-Resilient Screening of Nucleic Acid Synthesis Orders: Process, Results, and Recommendations. *bioRxiv*, December 2024. doi: 10.1101/2024.12.02.626439. URL https://www.biorxiv.org/content/10.1101/2024.12.02.626439. Preprint.

Bruce J. Wittmann, Kadina E. Johnston, Zachary Wu, and Frances H. Arnold. Advances in machine learning for directed evolution. *Current Opinion in Structural Biology*, 69:11–18, 2021a. ISSN 1879033X. doi: 10.1016/j.sbi.2021.01.008. URL https://doi.org/10.1016/j.sbi.2021.01.008. Publisher: Elsevier Ltd.

Bruce J. Wittmann, Yisong Yue, and Frances H. Arnold. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Systems*, 12(11):1026–1045.e7, 2021b. ISSN 24054712. doi: 10.1016/j.cels.2021.07.008. URL https://doi.org/10.1016/j.cels.2021.07.008. Publisher: Elsevier Inc.

Dongxia Wu, Nikki Lijing Kuang, Ruijia Niu, Yi-An Ma, and Rose Yu. Diffusion-BBO: Diffusion-Based Inverse Modeling for Online Black-Box Optimization, 2025. URL https://arxiv.org/abs/2407.00610. arXiv preprint; also presented as a poster at NeurIPS 2024 Workshop on Bayesian Decision-making and Uncertainty.

Luhuan Wu, Brian L. Trippe, Christian A. Naesseth, David M. Blei, and John P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models, 2024. URL https://arxiv.org/abs/2306.17775.

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

Zachary Wu, Kadina E. Johnston, Frances H. Arnold, and Kevin K. Yang. Protein sequence design with deep generative models. *Current Opinion in Chemical Biology*, 65:18–27, 2021. ISSN 13675931. doi: 10.1016/j.cbpa.2021.04.004. URL http://arxiv.org/abs/2104.04457%0Ahttp://dx.doi.org/10.1016/j.cbpa.2021.04.004. arXiv: 2104.04457 Publisher: Elsevier Ltd.

Junhao Xiong, Hunter Nisonoff, Ishan Gaur, Maria E. Lukarska, Luke M. Oltrogge, David F. Savage, and Jennifer Listgarten. Guide your favorite protein sequence generative model, May 2025. URL https://arxiv.org/abs/2505.04823. arXiv:2505.04823.

Jason Yang, Francesca-Zhoufan Li, and Frances H. Arnold. Opportunities and Challenges for Machine Learning-Assisted Enzyme Engineering. *ACS Central Science*, 10(2):226–241, February 2024. ISSN 2374-7943, 2374-7951. doi: 10.1021/acscentsci.3c01275. URL https://pubs.acs.org/doi/10.1021/acscentsci.3c01275.

Jason Yang, Aadyot Bhatnagar, Jeffrey A. Ruffolo, and Ali Madani. Function-guided conditional generation using protein language models with adapters. *arXiv*, June 2025a. doi: 10.48550/arXiv.2410.03634. URL https://arxiv.org/abs/2410.03634. arXiv:2410.03634 [q-bio.BM], v2, last revised 2025-06-11.

Jason Yang, Ravi G. Lal, James C. Bowden, Raul Astudillo, Mikhail A. Hameedi, Sukhvinder Kaur, Matthew Hill, Yisong Yue, and Frances H. Arnold. Active learning-assisted directed evolution. *Nature Communications*, 16(1):714, January 2025b. ISSN 2041-1723. doi: 10.1038/s41467-025-55987-8. URL https://www.nature.com/articles/s41467-025-55987-8. Publisher: Nature Publishing Group.

Jason Yang, Francesca-Zhoufan Li, Yueming Long, and Frances H. Arnold. Illuminating the universe of enzyme catalysis in the era of artificial intelligence. *Cell Systems*, pp. 101372, August 2025c. ISSN 2405-4712. doi: 10.1016/j.cels.2025.101372. URL https://www.sciencedirect.com/science/article/pii/S2405471225002054.

Kevin Yang and Dan Klein. FUDGE: Controlled Text Generation With Future Discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535, 2021. doi: 10.18653/v1/2021.naacl-main.276. URL http://arxiv.org/abs/2104.05218. arXiv:2104.05218 [cs].

Kevin K. Yang, Zachary Wu, and Frances H. Arnold. Machine-learning-guided directed evolution for protein engineering. *Nature Methods*, 16(8):687–694, 2019. ISSN 15487105. doi: 10.1038/s41592-019-0496-6. URL http://dx.doi.org/10.1038/s41592-019-0496-6. arXiv: 1811.10775 Publisher: Springer US.

Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2025. doi: 10.1109/CVPR52734.2025.01946. URL https://arxiv.org/abs/2407.01521. CVPR 2025; arXiv:2407.01521.

Junming Zhao, Chao Zhang, and Yunan Luo. Contrastive Fitness Learning: Reprogramming Protein Language Models for Low-N Learning of Protein Fitness Landscape. preprint, Bioinformatics, February 2024a. URL http://biorxiv.org/lookup/doi/10.1101/2024.02.11.579859.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Baker Grosse. Probabilistic Inference in Language Models via Twisted Sequential Monte Carlo. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 60704–60748. PMLR, July 2024b. URL https://proceedings.mlr.press/v235/zhao24c.html.

Hongkai Zheng, Wenda Chu, Bingliang Zhang, Zihui Wu, Austin Wang, Berthy T. Feng, Caifeng Zou, Yu Sun, Nikola Borislavov Kovachki, Zachary E. Ross, Katherine L. Bouman, and Yisong Yue. Inversebench: Benchmarking plug-and-play diffusion priors for inverse problems in physical sciences. In *Proceedings of the Thirteenth International Conference on Learning Representations (ICLR 2025)*, Singapore, may 2025. OpenReview. URL https://openreview.net/forum?id=U3PBITXNG6. ICLR 2025 Spotlight.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract summarizes the claims that we explain further in the Results and the Discussion.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: A detailed discussion of the limitations is provided in the Discussion section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All proofs have cited their sources.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The methods needed to reproduce the experimental results are provided in the appendix. The code will also be released to the public.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The codebase will be linked in the text upon de-anonymization.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: These details are provided in the appendix methods section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Standard deviations between multiple repetitions of the same experiment are provided in relevant figures.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The compute resources used to perform the experiments is provided.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes, the research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have cautioned the potential for negative societal impacts in the discussion.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our models only apply to designing proteins with minimal risk of negative impacts.

    Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The assets used comply with their corresponding licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The license will be posted with the code when it is released.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

Justification: No human subjects were used.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subjects were used.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were not a part of the core methods of the research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

# A Appendix

## A.1 Data for pretraining generative priors

The first step in our pipeline involves learning a generative prior on naturally occurring protein sequences to capture the distribution of those with high evolutionary likelihood. This prior is unconditional in the sense that no labeled fitness data is used for training. However, because we are optimizing protein *variants* for a desired fitness, we pretrained our generative prior on sequences homologous to the parent protein to be optimized (known as a multiple sequence alignment or MSA): TrpB, CreiLOV, or GB1. Likelihoods from MSAs have been captured by statistical models and have been shown to offer good zero-shot approximations of fitness. In other words, they capture mutational substitutions that are more favorable, based on the precedent of natural evolution.

We focused on the TrpB (Johnston et al., 2024) and CreiLOV (Chen et al., 2023c) datasets due to the extensive number of sequences in their MSAs and compared to GB1 (Olson et al., 2014), which has comparatively fewer sequences. MSAs were obtained by running jackhmmer (Johnson et al., 2010) against Uniref90 for two iterations with the parent sequence of the fitness dataset as target. For the MSA, we only used sequences where the aligned portion was at least 75% the length of the parent sequence. We used the MSA that was aligned to the parent sequence, with gap tokens replaced by the corresponding amino acid found in the parent sequence, resulting in full, fixed-length pseudo-natural sequences. For GB1, we augmented the training set with synthetic data, namely all proteins with a single mutation to sequences in the MSA. For the language models on TrpB and CreiLOV, some sequences were randomly mutated by a single position near the beginning of the sequence, to prevent mode collapse during autoregressive generation.

We performed sequence clustering using mmseqs2 (Steinegger & Söding, 2017) at 80% identity and resampled the dataset by weighting each sample with $\frac{1}{1+\ln(n)}$ relative probability of being sampled, where $n$ is the size of the cluster associated with that sequence. Afterward, we removed 5% of the clusters and their associated sequences as a validation set.

## A.2 Protein fitness optimization task

**An oracle as a proxy for protein fitness.** We studied fitness optimization across three different protein-fitness datasets, TrpB, CreiLOV, and GB1 (Table 2). TrpB is 389 residues in length, but based on available fitness data, we limited design to 15 residues: 117, 118, 119, 162, 166, 182, 183, 184, 185, 186, 227, 228, 230, 231, and 301. Namely, we combined the fitness data from 6 combinatorially complete 3-site libraries (D-I from Johnston et al. (2024)) and the 4-site library across residues 183, 184, 227, and 228. We normalized the parent fitness to 1 in each dataset and rounded all negative fitness values up to zero. The fitness here is the catalytic rate of a native reaction, the formation of tryptophan from indole and serine. To obtain a proxy fitness for all variants in the design space ($20^{15}$ possibilities) we trained an oracle inspired by the dataset splitting and model architecture used in Blalock et al. (2025). Namely, we used all of the single, double, and triple mutants in the library for training, with 10% and 20% of the quadruple mutants being used for validation and testing, respectively. Our model consists of an ensemble of 20 MLPs for TrpB, and each was trained on one-hot encodings of the designed residues for 1000 epochs.

Differently, the CreiLOV dataset (length $N = 119$) contains experimental fitnesses for all single mutations in the protein and certain higher order mutations at 15 selected positions with beneficial single mutations. Fitness here refers to associated fluorescence. To obtain a proxy fitness for all variants in the design space ($20^{119}$ possibilities), we trained an oracle similar to the procedure above, using similar splits to those in Blalock et al. (2025) and were able to reproduce their high performance on the test set. Before model training, we scaled the fitnesses of the single mutants to the fitnesses of multi-mutants by adding a normalization factor to all single mutants such that the parent sequence in both datasets had the same fitness. Our model consists of an ensemble of 10 MLPs for CreiLOV, and each was trained on onehot encodings of sequences for 1000 epochs.

For GB1, the experimental finesses for nearly all double mutations across the entire protein were available, where fitness refers to binding affinity of a domain of the G protein. To train the oracle, we held out 10% and 20% the sequences with two mutations as a validation and test set, respectively, with remaining sequences being used for training. Our model consists of an ensemble of 10 MLPs for GB1, and each was trained on one-hot encodings of the designed residues for 50 epochs.

Our oracles show high Pearson correlation on the train and test sets (Fig. A1). As the generalization ability of our oracle was only been tested on variants that are similar to the parent, we penalized the fitness of protein sequences by a factor of 0.99 for every mutation accumulated beyond a threshold of 60% sequence identity to the parent sequence. From here forth, we treated ground truth fitness as outputs from the oracle.

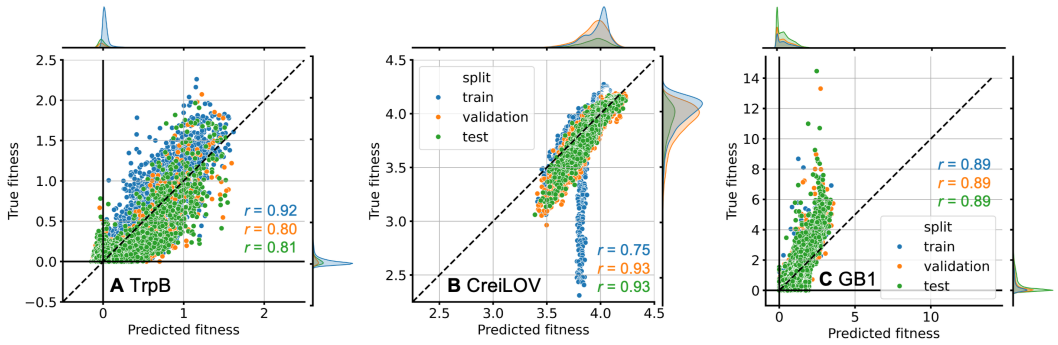


Figure A1: Oracles trained on available labeled fitness data for TrpB, CreiLOV, and GB1 extrapolate well to higher order combinations of mutations within the design space, as measured by Pearson correlation.

**Processing generated sequences.** Our primary method for evaluation involved examining the distribution of sampled sequences and their corresponding fitness values, diversities, and novelties. The processing pipeline for generated sequences in shown in Fig. A2. In diffusion models, sequences were generated with fixed length equal to the parent length. For the language models, nearly all generated sequences had length equal to the parent sequence length. Still, sequences were aligned with the parent sequence using mafft (Katoh & Standley, 2013), and gaps were replaced with the corresponding amino acid in the parent sequence to generate complete pseudo-sequences of a fixed length. Special tokens, which occurred rarely in generation, were replaced by a random amino acid. For TrpB, residues outside of the design space of 15 residues were naively mapped to the original amino acid type in the parent sequence at the end of generation. We did not test inpainting, although this could be accomplished with masked (diffusion) language models.

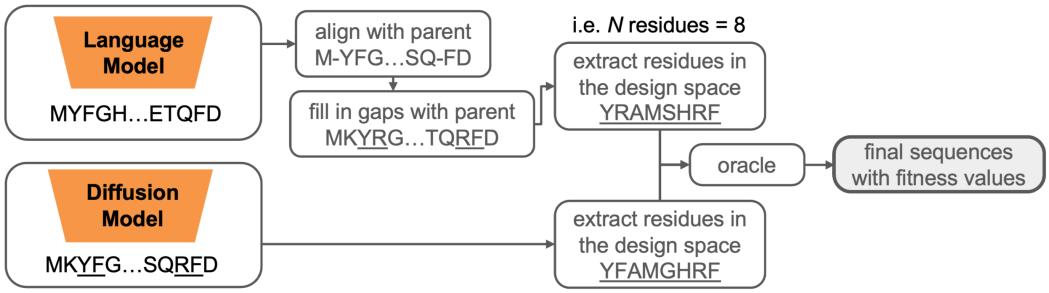


Figure A2: Example pipeline for generating protein sequences for evaluation, based on a hypothetical parent sequence: MKKFG...SQRFD (length=100), with 8 residues being optimized (3, 4, 26, 27, 28, 29, 98, 99), corresponding to a design space combo of KFDEACRF.

**Comparison to existing protein engineering methods.** There are several reasons why we did not directly compare the performance of SGPO methods to existing methods used in protein engineering, such as directed evolution and MLDE. In the case of directed evolution (such as random mutagenesis): (1) It is not obvious which parent sequences to use as the starting points for directed evolution for a fair comparison. (2) It is unclear if the oracle captures the true nature of the protein fitness landscape or extrapolates well to sequences with many mutations relative to the original fitness dataset from which the oracle was trained. (3) Overall, our method enables the accumulation of many mutations in a single round of experimentation, whereas directed evolution is largely limited to one mutation at a time. For example, on the CreiLOV dataset, the generated sequences with the highest fitness had on average 66 mutations from the parent reference sequence from which the original dataset was

generated, which would not be achievable with directed evolution. We also did not directly compare our method to supervised approaches in smaller design spaces, such as 4-site combinatorial libraries (Yang et al., 2025b), as we focus here on design in larger design spaces, where existing methods are lacking. Overall, traversing large swaths of sequence space will be important for faster engineering and enabling improvements to fitness that would normally be slow with directed evolution.

### A.3 Generative models for sequences

Table A1: **Summary of training details for generative priors in this work.** Reference refers to the codebase that was modified for our implementation and where the model architecture was adapted from. For all models, we retained the model with the lowest validation loss. When using the ESM encoder, we used the 35M-parameter ESM2 model Lin et al. (2023).

| Model | Max Epochs | Learning Rate | Batch Size | Warmup Steps | Noise Schedule | Diffusion Timesteps | Model Architecture | Reference |
|---|---|---|---|---|---|---|---|---|
| **Continuous** | 5 | $1\times10^{-4}$ | 64 | 10 | cosine | 500 | BERT | Gruver et al. (2023) |
| Continuous-ESM | 25 | $1\times10^{-4}$ | 64 | 10 | cosine | 500 | BERT | Gruver et al. (2023) |
| D3PM-Baseline | 5 | $1\times10^{-4}$ | 64 | 10 | Sohl-Dickstein | 500 | ByteNet | Alamdari et al. (2024) |
| **D3PM** | 5 | $1\times10^{-4}$ | 64 | 10 | Sohl-Dickstein | 500 | ByteNet | Alamdari et al. (2024) |
| UDLM | 5 | $3\times10^{-5}$ | 64 | 2500 | loglinear | 500 | DiT | Schiff et al. (2024) |
| **MDLM** | 50 | $3\times10^{-4}$ | 64 | 2500 | loglinear | 500 | DiT | Schiff et al. (2024) |
| **ARLM** | 10 | $1\times10^{-4}$ | 32 | 10 | n/a | n/a | GPT-J | Nijkamp et al. (2023) |

#### A.3.1 Diffusion over continuous space

Diffusion models construct samples by reversing a diffusion process that maps clean data points $\mathbf{x}_0$ to samples from a prior distribution $\pi(\mathbf{x})$. The forward process ($\mathbf{x}_0 \to \mathbf{x}_T$) is composed of conditional distributions $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, which admit closed-form expressions for the conditional distributions $p(\mathbf{x}_t|\mathbf{x}_0)$ and $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$. The reverse process ($\mathbf{x}_T \to \mathbf{x}_0$) converts samples from the prior into samples from the learned data distribution $p_\theta(\mathbf{x}_0)$ by repeatedly predicting the denoised variable $\hat{\mathbf{x}}_0$ from noisy values $\mathbf{x}_t$, using the conditional distribution $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}}_0)$ to derive a transition distribution $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$.

**Continuous noise forward process.** Similarly to Gruver et al. (2023), we define a protein sequence as $\mathbf{w} \in \mathcal{A}^L$, where $\mathcal{A}$ is the alphabet of amino acids and $L$ is the fixed length of the sequence. To learn a distribution $p(\mathbf{w})$, we first embed $\mathbf{w}$ into a continuous variable $\mathbf{x}_0$ using an embedding matrix $U_\theta$ or encoder from the ESM2 language model (Lin et al., 2023), transforming discrete tokens into a continuous latent space. Gaussian noise is then applied to this embedding space. The prior distribution is defined as:

$$\pi(\mathbf{x}) = \mathcal{N}(0, I), \tag{1}$$

while the forward process follows a Gaussian corruption schedule:

$$p(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)I), \quad \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i, \quad \alpha_t = 1 - \beta_t. \tag{2}$$

The variance schedule $\{\beta_t\}$ follows the cosine schedule proposed by Nichol & Dhariwal (2021), which is commonly used to stabilize training.

**Reverse process.** The reverse process aims to recover the original sequence by learning a function $p_\theta(\hat{\mathbf{w}}|\mathbf{x}_t, t)$ that predicts the sequence from noised points $\mathbf{x}_t$. This is done by minimizing the following objective:

$$L(\theta) = \mathbb{E}_{\mathbf{w}_0,t}\left[-\log p_\theta(\mathbf{w}_0|\mathbf{x}_t)\right], \quad \mathbf{x}_t \sim p(\mathbf{x}_t|\mathbf{x}_0 = U_\theta \mathbf{w}_0). \tag{3}$$

By learning $p_\theta(\hat{\mathbf{w}}|\mathbf{x}_t, t)$, we construct the reverse transition distribution:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \sum_{\hat{\mathbf{w}}} p(\mathbf{x}_{t-1}|\mathbf{x}_t, \hat{\mathbf{x}}_0 = U_\theta \hat{\mathbf{w}}) p_\theta(\hat{\mathbf{w}}|\mathbf{x}_t, t), \tag{4}$$

where the posterior $p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$ follows:

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_{t-1}; \mu_t, \sigma_t^2 I), \tag{5}$$

with mean $\mu_t$ and variance $\sigma_t^2$ given by:

$$\mu_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}\mathbf{x}_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\mathbf{x}_t, \tag{6}$$

$$\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t. \tag{7}$$

**Inference and sampling.** At inference time, the learned reverse process is used to generate protein sequences from the prior $\pi(x)$. This is done by iteratively sampling:

$$\mathbf{x}_{t-1} \sim p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \tag{8}$$

and then reconstructing $\mathbf{w}$ by sampling:

$$\mathbf{w} \sim p_\theta(\hat{\mathbf{w}}|\mathbf{x}_0). \tag{9}$$

This denoising process iteratively refines noisy embeddings back into structured sequences.

### A.3.2 Diffusion over discrete space.

Discrete diffusion models (Austin et al., 2021; Campbell et al., 2022; Lou et al., 2024) generate data in discrete spaces by reversing a predefined forward Markov process. Specifically, a family of distributions $p_t$ evolves according to the Markov chain

$$\frac{\mathrm{d}p_t}{\mathrm{d}t} = \boldsymbol{Q}_t p_t, \tag{10}$$

where $p_0 = p_{\mathrm{data}}$ is the data distribution and $\boldsymbol{Q}_t \in \mathbb{R}^{N\times N}$ are predefined transition matrices.

This Markov process can be reversed with the help of a concrete score function, $s(\mathbf{x}, t) := [\frac{p_t(\tilde{\mathbf{x}})}{p_t(\mathbf{x})}]_{\tilde{\mathbf{x}}\neq\mathbf{x}}$, as its time reversal is given by

$$\frac{\mathrm{d}p_{T-t}}{\mathrm{d}t} = \bar{\boldsymbol{Q}}_{T-t} p_{T-t}, \tag{11}$$

where $\bar{\boldsymbol{Q}}_t[\tilde{\mathbf{x}}, \mathbf{x}] = s(\mathbf{x}, t)_{\tilde{x}} \boldsymbol{Q}_t[\mathbf{x}, \tilde{\mathbf{x}}]$ for $\tilde{\mathbf{x}} \neq \mathbf{x}$, and $\bar{\boldsymbol{Q}}_t[\mathbf{x}, \mathbf{x}] = -\sum_{\tilde{\mathbf{x}}\neq\mathbf{x}} \bar{\boldsymbol{Q}}_t[\tilde{\mathbf{x}}, \mathbf{x}]$. To generate data $\mathbf{x}_0 \sim p_{\mathrm{data}}$, we start with sampling $\mathbf{x}_T$ from a uniform distribution and then evolve through Eq. 11 by the Euler method.

**Uniform discrete language models.** Both D3PM (Austin et al., 2021) and UDLM (Schiff et al., 2024) implement a uniform transition matrix $\boldsymbol{Q}_t = \frac{1}{N}\mathbf{1}\mathbf{1}^T - \boldsymbol{I}$. When $T \to \infty$, the probability distribution $p_T$ converges to a uniform distribution.

**Masked diffusion language models.** Masked diffusion language models (MDLM) (Sahoo et al., 2024) utilize an absorbing transition matrix $\boldsymbol{Q}_t$ that converts tokens in a sequence to [MASK] states. The corresponding transition matrix can be written as $\boldsymbol{Q}_t \in \mathbb{R}^{(N+1)\times(N+1)}$, $\boldsymbol{Q}_t = -\boldsymbol{I} + \mathbf{e}_{N+1}\mathbf{1}^T$. When $T \to \infty$, the limiting distribution $p_T$ converges to a completely masked sequence.

### A.3.3 Autoregressive language models.

In this work, we finetuned the ProGen2-small decoder-only transformer (151 million parameters) based on the code and parameters used in Yang et al. (2025a). Models were trained based on next token prediction and cross entropy loss. However, we did not use adapter layers, and we did not group batches based on sequence length. During inference from the autoregressive model, we used a temperature of 1 and a Top-$p$ value of 1.

## A.4 Steering methods

Table A2: **Summary of supervised value functions used to predict fitness in this work, to guide diffusion models.** All "classifiers" were trained as regressors to predict fitness. For DAPS methods, only clean data was used for training, whereas other classifiers are trained on clean and noised samples from various timesteps.

| Model | Guidance Strategy | Max Epochs | Learning Rate | Batch Size | Architecture | Hidden Dimension |
|---|---|---|---|---|---|---|
| Continuous Diffusion | CG | 1000 | $1 \times 10^{-3}$ | 128 | 4-layer MLP | 256 |
| | DAPS | 1000 | $1 \times 10^{-3}$ | 128 | 4-layer MLP | 256 |
| Continuous Diffusion | NOS | 100 | $1 \times 10^{-3}$ | 128 | 1-layer MLP | 256 |
| Discrete Diffusion | CG | 1000 | $1 \times 10^{-3}$ | 64 | 4-layer MLP | 64 |
| | DAPS | 200 | $1 \times 10^{-3}$ | 64 | 4-layer MLP | 64 |
| Discrete Diffusion | NOS | 200 | $3 \times 10^{-4}$ | 64 | linear layer | n/a |

### A.4.1 Classifier guidance

Classifier guidance (Song et al., 2021) is a technique used to steer samples generated by diffusion models toward desired attributes. The primary goal is to sample from a conditional distribution $p(\mathbf{x}|\mathbf{y})$, where $\mathbf{y}$ is a guiding signal of interest. In continuous space, this can be achieved by replacing the unconditional score function $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t)$ at time $t$ by a conditional score function,

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{y}) = \nabla_{\mathbf{x}_t} \log p_t(\mathbf{x}_t) + \nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) \tag{12}$$

To obtain the conditional score function, one only needs to train a time-dependent predictor, which predicts the probability of $p_t(\mathbf{y}|\mathbf{x}_t)$ given $\mathbf{x}_t$ and time $t$.

**Continuous guidance.** Classifier guidance modifies the reverse diffusion process to steer generated samples toward a desired property, represented by a conditioning variable $y$. The guided sampling process modifies the update rule for $\mathbf{x}_t$ by incorporating a classifier score $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)$ into the model's learned score function based on the relation in Eq. 12. Following Song et al. (2021), the classifier guidance term modifies the predicted $\hat{\mathbf{x}}_0$ in the denoising process:

$$\hat{\mathbf{x}}_0 = \mathbf{x}_t + \sigma^2(s_\theta(\mathbf{x}_t, t) + \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t)). \tag{13}$$

Since our diffusion model directly predicts logits rather than the score function $s_\theta(\mathbf{x}_t, t)$, adding classifier guidance requires modifying the predicted $\hat{\mathbf{x}}_0$.

Instead of predicting the score function explicitly, our model predicts logits over the vocabulary, from which the denoised representation $\hat{\mathbf{x}}_0$ is obtained. We modify $\hat{\mathbf{x}}_0$ by incorporating classifier gradients as follows:

- Compute the unmodified $\tilde{\mathbf{x}}_0$ using the model's predicted logits:

$$\tilde{\mathbf{x}}_0 = \sum_{\hat{\mathbf{w}}} p(\hat{\mathbf{w}}|\mathbf{x}_t, t) U_\theta \hat{\mathbf{w}} \tag{14}$$

Table A3: **Hyperparameters used to tune the guidance/steering process.** The **bolded** parameter was chosen as the ideal parameter for the iterative "Bayesian optimization" experiment (Fig. 6). Larger guidance parameter typically implements stronger guidance strength.

| Guidance Strategy | Hyperparameters |
|---|---|
| Continuous CG | $1/\beta = 64, 128, 256, 512, 1024$ |
| Discrete CG | $1/\beta = 1, 2.5, 6.25, \mathbf{15.625}, 39.0625$ |
| Continuous DAPS | $1/\beta = 0.25, 0.5, 1, 2, 4 \times 10^4$<br>$K = 50$<br>Euler method steps = 10<br>Langevin dynamics steps = 100 |
| Discrete DAPS | $1/\beta = 16, 32, 64, \mathbf{128}, 256$<br>$K = 50$<br>Euler method steps = 20<br>Metropolis Hastings steps = 1000 |
| Continuous NOS | $\lambda = 0.1, 1, 10, 100, \mathbf{1000}$<br>$\eta = 0.5, \mathbf{2}, 5$<br>$K = 5, \mathbf{10}$<br>optimizer = AdaGrad |
| Discrete NOS | $\lambda = 0.1, 1, 10, 100, 1000$<br>$\eta = 0.5, 2, 5$<br>$K = 5, 10$<br>optimizer = AdaGrad |
| DPO | $\beta = 0.02, 0.1, 0.5, \mathbf{2}, 4$<br>lr = $1 \times 10^{-6}$<br>epochs = 5<br>batch size = 8 |

where $U_\theta$ is the embedding matrix mapping discrete tokens to continuous space.

- If a time-dependent classifier $f$ is available, compute the classifier guidance term:

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} f(\mathbf{x}_t, t)/\beta. \tag{15}$$

- Modify $\tilde{\mathbf{x}}_0$ using the classifier gradient:

$$\hat{\mathbf{x}}_0 = \tilde{\mathbf{x}}_0 + \sigma^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{y}|\mathbf{x}_t). \tag{16}$$

This allows the diffusion model to generate samples that are more likely to satisfy the desired condition $\mathbf{y}$.

Further details on training the classifier are provided in Table A2 and Table A3.

**Discrete guidance.** Nisonoff et al. (2025) extend classifier guidance to discrete state-space diffusion models. In analogy to classifier guidance for continuous diffusion models, they modify the unconditional rate matrix $\bar{\boldsymbol{Q}}_t$ (as defined in Eq. 11) to be a conditional rate matrix $\boldsymbol{R}_t^{\mathbf{y}}$ with

$$\boldsymbol{R}_t^{\mathbf{y}}[\mathbf{x}, \tilde{\mathbf{x}}] = \frac{p(\mathbf{y}|\tilde{\mathbf{x}}, t)}{p(\mathbf{y}|\mathbf{x}, t)} \bar{\boldsymbol{Q}}_t[\mathbf{x}, \tilde{\mathbf{x}}], \ \forall \tilde{\mathbf{x}} \neq \mathbf{x}. \tag{17}$$

For classifier guidance on both continuous and discrete diffusion models, we train a time-dependent predictor (classifier) $f$ that predicts the fitness $\mathbf{y}$ given $\mathbf{x}_t$ at time $t$. We define $p(\mathbf{y}|\mathbf{x}) \propto \exp(f(\mathbf{x})/\beta)$, where $f(\cdot)$ is a surrogate predictor of the fitness, and $\beta$ is the guidance temperature and governs the strength of guidance. Therefore, $\nabla_{\mathbf{x}_t} \log p_t(\mathbf{y}|\mathbf{x}_t) = \frac{1}{\beta} \nabla_{\mathbf{x}_t} f(\mathbf{x}_t, t)$, and $\boldsymbol{R}_t^{\mathbf{y}}[\mathbf{x}, \tilde{\mathbf{x}}] = \exp\left(\frac{1}{\beta}\big(f(\tilde{\mathbf{x}}, t) - f(\tilde{\mathbf{x}}, t)\big)\right) \bar{\boldsymbol{Q}}_t[\mathbf{x}, \tilde{\mathbf{x}}]$.

To obtain a classifier $f$ for discrete diffusion models, we trained an MLP regressor to predict the fitness of a one-hot encoded sequence given $\mathbf{x}_t$ and uniformly random time $t \in [0, T]$. Further details are provided in Table A2 and Table A3.

### A.4.2 Posterior sampling

Another line of guidance work (Chung et al., 2023; Mardani et al., 2024; Zhang et al., 2025) focuses on drawing samples from the posterior distribution $p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{x})p(\mathbf{y}|\mathbf{x})$, where the prior distribution is modeled by a pretrained diffusion model. The conditional distribution $p(\mathbf{y}|\mathbf{x})$ can either be the likelihood function of a forward model (i.e., when $\mathbf{y}$ is an incomplete measurement of $\mathbf{x}$) or an exponential distribution with respect to a reward function (i.e., $p(\mathbf{y}|\mathbf{x}) \propto \exp(f(\mathbf{x})/\beta)$). The major difference between posterior sampling and classifier guidance is that it requires the reward function to be trained only on clean data $\mathbf{x}$.

While many works have studied posterior sampling in Euclidean space with continuous diffusion models, posterior sampling for discrete data has been less explored. We modified DAPS (Zhang et al., 2025) to enable diffusion posterior sampling in discrete-state spaces. Suppose $\mathbf{x}$ lies in a finite support $\mathcal{X}^D$, we follow the following steps:

- Initialize $\mathbf{x}_T \sim p_T(\mathbf{x}_T)$
- for $i = 1, \ldots, K$
  1. Sample $\hat{\mathbf{x}}_0^{(i)} \sim p(\mathbf{x}_0|\mathbf{x}_{t_{i-1}})$ by a discrete diffusion model.
  2. Run Metropolis Hastings to sample $\mathbf{x}_0^{(i)} \sim p(\mathbf{x}_0|\mathbf{x}_{t_{i-1}}, \mathbf{y})$ as defined in Eq. 18.
  3. Sample $\mathbf{x}_{t_i} \sim p(\mathbf{x}_{t_i}|\mathbf{x}_0)$ following the forward Markov process.
- Return $\mathbf{x}_K$.

Specifically, $t_0, t_1, \ldots, t_K$ are mono-decreasing time steps with $t_0 = T$ and $t_K \approx 0$. $p(\mathbf{x}_0|\mathbf{x}_t, \mathbf{y})$ is defined as

$$\begin{aligned} p(\mathbf{x}_0|\mathbf{x}_t, \mathbf{y}) &\propto p(\mathbf{y}|\mathbf{x}_0)p(\mathbf{x}_0|\mathbf{x}_t) \\ &\approx p(\mathbf{y}|\mathbf{x}_0) \exp(-\|\mathbf{x}_0 - \hat{\mathbf{x}}_0(\mathbf{x}_t)\|_0/\sigma_t), \end{aligned} \tag{18}$$

where $\hat{\mathbf{x}}_0(\mathbf{x}_t) \sim p(\mathbf{x}_0|\mathbf{x}_t)$ is a point estimate of the conditional distribution, and we approximate $p(\mathbf{x}_0|\mathbf{x}_t)$ by an exponential distribution over Hamming distance. Following Proposition 1 in Zhang et al. (2025), $\hat{\mathbf{x}}_0^{(i)}$, $\mathbf{x}_0^{(i)}$, and $\mathbf{x}_{t_i}$ converge to the posterior distribution as $t_i$ goes to 0.

For posterior sampling with DAPS, we obtained the value function $f$ using the same model architecture and training parameters as classifier guidance but only trained on clean data $\mathbf{x}$ (no noisy $\mathbf{x}_t$). We set $K = 50$ using the time scheduler for the original model. Further details are provided in Table A2 and Table A3.

### A.4.3 NOS

Diffusion optimized sampling (NOS) (Gruver et al., 2023) is a guidance method for both continuous and discrete diffusion models, which utilizes gradient information of the continuous latent representations of protein sequences. In pretrained discrete diffusion models, noisy sequences $\mathbf{w}_t$ always have a continuous embedding in the form of hidden states of the neural network. Specifically, the denoising model that predicts $\mathbf{w}_0$ from $\mathbf{w}_t$ can be written as $p_\theta(\mathbf{w}_0|g(\mathbf{w}_t), t)$, where $\mathbf{h}_t = g(\mathbf{w}_t)$ is a continuous hidden states of the model.

Instead of training a value function on discrete sequences $\mathbf{w}_t$, NOS proposes to train the value function on the hidden states $\mathbf{h}_t$. In each diffusion step, NOS samples from the posterior distribution,

$$p(\mathbf{w}_0|\mathbf{h}_t, \mathbf{y}) \propto p_\theta(\mathbf{w}_0|\mathbf{h}_t) \exp(f(\mathbf{h}_t)). \tag{19}$$

To sample from this distribution, NOS runs Langevin dynamics on $\mathbf{h}_t$, i.e.,

$$\mathbf{h}_t' \leftarrow \mathbf{h}_t' - \eta \nabla_{\mathbf{h}_t'} (\lambda D_{KL}(p_\theta(\mathbf{w}_0|\mathbf{h}_t') \| p_\theta(\mathbf{w}_0|\mathbf{h}_t)) - f(\mathbf{h}_t)) + \sqrt{2\eta\tau}\epsilon, \; \epsilon \sim \mathcal{N}(0, I). \tag{20}$$

After $K$ iterations, we denoise $\mathbf{w}_t$ following the guided hidden state, i.e., $p(\mathbf{w}_{t-1}|\mathbf{w}_t, \mathbf{y}) = p_\theta(\mathbf{w}_{t-1}|\mathbf{h}_t', t)$.

To train the value function used for guidance in NOS, following the method from Gruver et al. (2023), we trained a very shallow neural network on the final layer hidden embeddings of the diffusion model. Further details are provided in Table A2 and Table A3.

### A.4.4 Direct preference optimization

For DPO with language models, we used the weighted loss function from Widatalla et al. (2024) and Stocco et al. (2024) (Eq. 21). $\pi_\theta$ is the policy to be updated, $\pi_{\text{ref}}$ is the original model, and $\beta$ is a tunable parameter describing the extent of drift from the reference model. The loss therefore describes the cross entropy of the ratio $\beta \log \frac{\pi_\theta(\mathbf{x})}{\pi_{\text{ref}}(\mathbf{x})}$ and the fitness value $w$. Following Stocco et al. (2024), we calculated the ratio $r$ as the difference of the log likelihood of the sequence from the updated model minus the log likelihood of the reference model, and softmax was applied to all of the fitness values $w$. We used the default parameters from (Stocco et al., 2024) and tested increasing the learning rate to $10^{-4}$ but found that generation quality broke down above the levels used in Table A3 with finetuning for 5 epochs. We also tested ranked loss with other types of models, but the performance was similar.

$$L_{\text{DPO}_{\text{weighted}}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_D \sum_{k=1}^{K} w^k \left[ \beta \log \frac{\pi_\theta(x)}{\pi_{\text{ref}}(x)} - \log \sum_{j=k}^{K} \exp\left( \beta \log \frac{\pi_\theta(x)}{\pi_{\text{ref}}(x)} \right) \right] \tag{21}$$

### A.5 Adaptive optimization algorithm

**Algorithm 1** Adaptive Optimization with Guided Generative Models

**Input:** Pretrained generative prior $p(\mathbf{x})$, initial empty labeled dataset $\mathcal{D}_0 = \emptyset$, number of rounds $T$, batch size $B$, ensemble size M
**for** $t = 1$ to $T$ **do**
  Initialize batch $\mathcal{X}_t \leftarrow \emptyset$
  **if** $t > 1$ **then** Train ensemble of value functions $\{f_{\theta_{t,m}}\}_{m=1}^{M}$ on $\mathcal{D}_{t-1}$
  **while** $|\mathcal{X}_t| < B$ **do**
    **if** $t > 1$ **then**
      Sample value function $f_\theta \sim \text{Uniform}(\{f_{\theta_{t,m}}\}_{m=1}^{M})$ ▷ Thompson-style sampling
      Sample sequence $\mathbf{x}_b \sim \text{GuidedSample}(p(\mathbf{x}), f_\theta, \text{GuidanceStrategy})$
    **else**
      Sample sequence $\mathbf{x}_b \sim \text{UnconditionalSample}(p(\mathbf{x}))$
    **end if**
    **if** $\mathrm{x}_b \notin \mathcal{D}_{t-1}$ **then** Add $\mathbf{x}_b$ to batch $\mathcal{X}_t$
  **end while**
  Evaluate true fitness $y_b = f_{\text{true}}(\mathbf{x}_b)$ for all $\mathbf{x}_b \in \mathcal{X}_t$
  Update dataset: $\mathcal{D}_t \leftarrow \mathcal{D}_{t-1} \cup \{(\mathbf{x}_b, y_b)\}_{b=1}^{B}$
**end for**
**Return:** Best observed sequence in $\mathcal{D}_T$

We used an ensemble size of $M = 10$ models, each trained with a different random initialization of neural network weights. In practice, to speed up sampling, we sampled ($B = 100$ samples)/($M = 10$ models) = 10 sequences in each GPU batch using the same Thompson-sampled value function, rather than using a GPU batch size of 1. Alternatively, for the Gaussian process model, we trained the model with the radial basis function kernel, and we sub-sampled the total amount of training pairs (when using noisy samples) to 5000 samples.

### A.6 Latent space Bayesian optimization with

We utilized the APEXGo codebase (Torres et al., 2024), a package for training generative variational autoencoders over peptide sequences and then optimizing those sequences with latent space Bayesian optimization to maximize certain properties. We used the training code out-of-the-box to train variational autoencoders over the same MSA sequences used to train priors for discrete diffusion models in SGPO. We trained until losses plateaued, to 542, 241, and 391 epochs for TrpB, CreiLOV, and GB1, respectively. Overall, the reconstruction losses were low, and generated sequences had low perplexity and high fitness, comparable to the generative models used in SGPO. We then used this latent space and the APEXGo optimization algorithm to maximize the fitness of sequences as measured by the oracle used in SGPO benchmarking. Specifically, in our configuration, we set

the number of initialization points to 100, the number of desired diverse solutions to 1, the max number of oracle calls to 800, and the batch size to 100–to mimic the iterative Bayesian optimization experiments performed in Fig. 6.

### A.7 Additional results

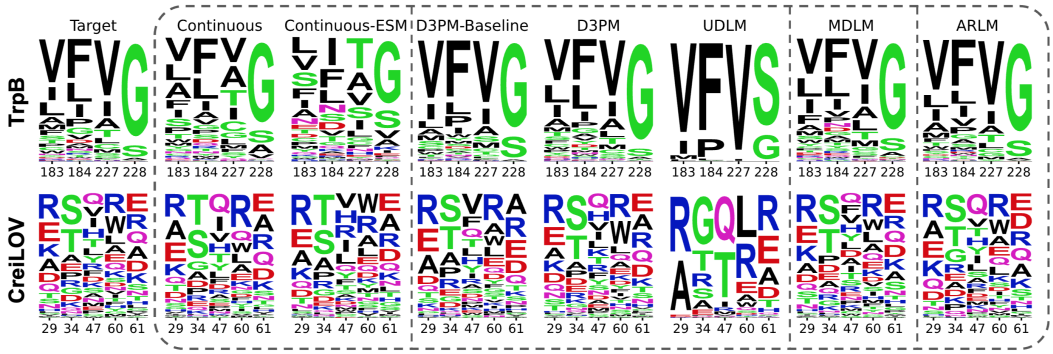


Figure A3: **The distributions of sequences sampled from pretrained generative priors largely match those of the target distribution.** The target distribution shows all sequences in the MSA, and the distributions of generative models are approximated by sampling 1000 sequences. Model definitions can be found in Table 3. The residues shown for TrpB are 4 out of 15 positions studied in the dataset (parent is VFVS), and 5 out of 119 residues for CreiLOV are shown as they correspond to those harboring favorable mutations in the original dataset (parent is AGQRD). Note that the target distribution for training the ARLM is slightly different than that shown here.

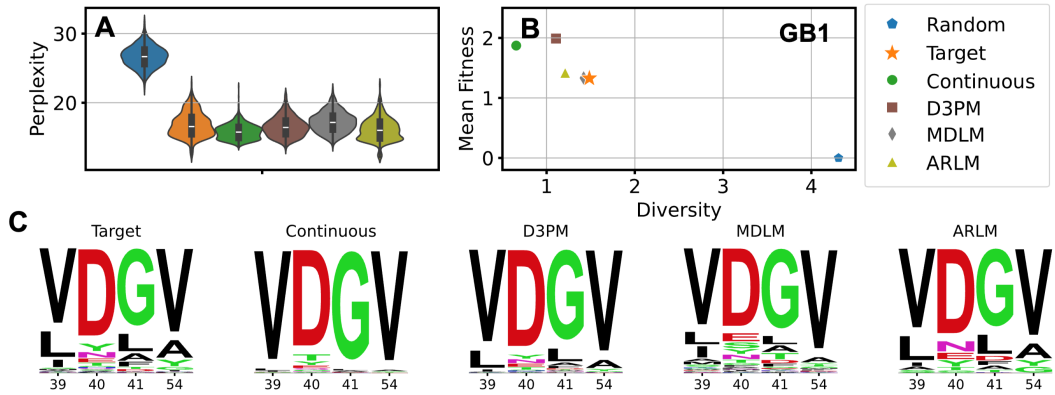


Figure A4: Additional results for GB1, corresponding to Fig. 4 and Fig. A3. The 4 positions shown correspond to the parent sequence of VDGV.

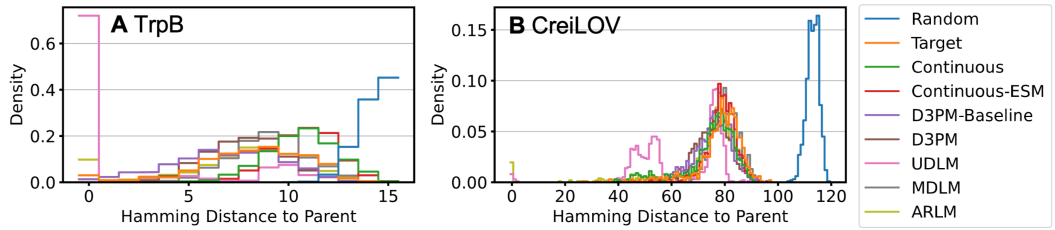


Figure A5: Generated sequences from pretrained priors are more similar to parent than random for (**A**) TrpB and (**B**) CreiLOV, measured by the Hamming (or edit) distance. UDLM models exhibit mode collapse onto consensus sequence(s) in the training distribution. The parent sequence refers to the starting sequence used to generate variants in the original protein fitness dataset.

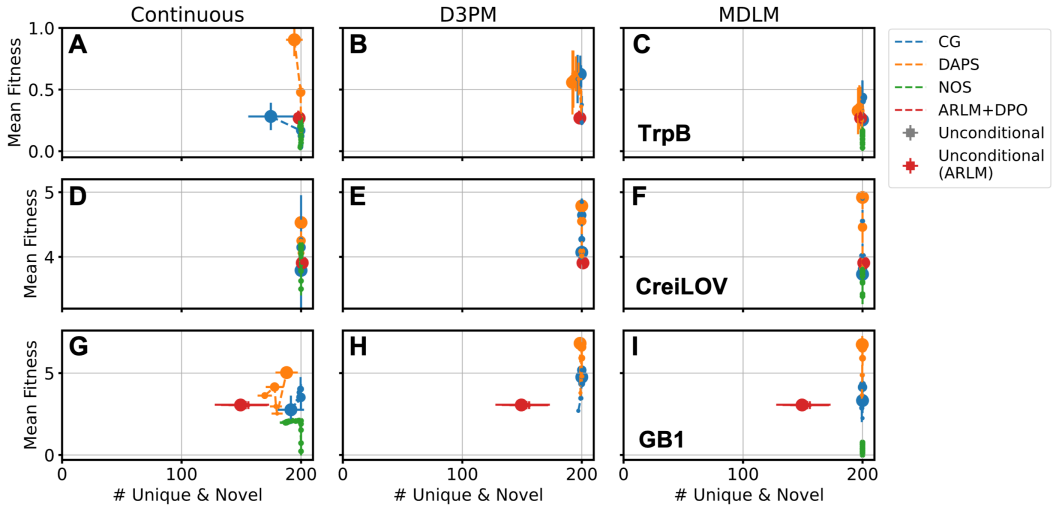


Figure A6: Pareto boundaries demonstrate the trade-off between generating sequences with high fitness and high diversity for TrpB (**A-C**), CreiLOV (**D-F**), and GB1 (**G-I**) – showing the same experiment as Fig. 5. Error bars show standard deviation. Mean fitness and diversity were calculated based on 200 generated samples, with diversity calculated as the total number of unique and novel (previously unseen) samples in the generated batch, out of 200. Larger circles indicate a stronger guidance strength, specified in Table. A3.

Table A4: Adaptive optimization with an ensemble of 10 value functions and Thompson sampling, compared to using a single model for guidance. Max fitness refers to the mean max fitness achieved at the end of the campaign using the same experimental setup as Fig. 6, over 5 different random initializations.

| Protein | Model | Guidance | Max Fitness (Ensemble) | Max Fitness (Single Model) |
|---|---|---|---|---|
| TrpB | D3PM | CG | **1.551** | 1.542 |
| | | DAPS | **1.595** | 1.568 |
| | MDLM | CG | **1.551** | 1.542 |
| | | DAPS | **1.595** | 1.568 |
| CreiLOV | D3PM | CG | **5.608** | 5.552 |
| | | DAPS | **5.522** | 5.520 |
| | MDLM | CG | **5.608** | 5.552 |
| | | DAPS | **5.530** | 5.520 |

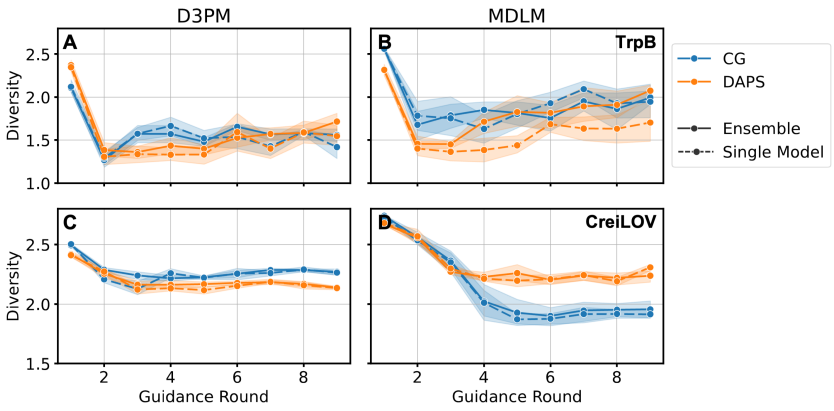


Figure A7: Diversity of generated sequences, measured by average Shannon entropy of mutated positions, during each round of guidance. Using an ensemble of value functions and Thompson sampling generally shows higher diversity than using a single model. Experimental setup is the same as Fig. 6, and experiments were repeated over 5 random initializations.