Steering Generative Models with Experimental Data for Protein Fitness Optimization

Jason Yang* Caltech
Wenda Chu* Caltech
Daniel P. Khalil Caltech
Raul Astudillo Caltech
Bruce J. Wittmann Microsoft Corporation
Frances H. Arnold Caltech
Yisong Yue Caltech

Abstract

Protein fitness optimization involves finding an ideal protein sequence satisfying desired quantitative properties in an astronomically large design space of possible sequences, where it is often only possible to measure real-world fitness for few (hundreds of) sequences. Existing machine learning approaches for efficiently navigating the protein design space broadly fall into two categories–discriminative (often supervised) modeling and generative modeling–each with their own strengths and weaknesses. Supervised models can be used to identify promising variants, but require predicting fitness values for all possible sequences in a design space. Generative models, on the contrary, are not hampered by the size of a design space, but historically it has been difficult to direct these models toward specific fitness goals. To address these limitations, we propose a framework for protein sequence optimization in which generative priors on natural sequences are steered with assay-labeled fitness data, taking advantage of both unlabeled and labeled data. Specifically, we evaluate discrete diffusion and language models in combination with various steering techniques such as guidance and reinforcement learning. Our computational studies on the TrpB and CreiLOV protein fitness datasets show that various methods, particularly guidance with discrete diffusion models, are effective strategies for 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, the protein optimization problem is challenging: the design space of proteins is vast, as a protein of length M can be constructed in 20^M different ways; only a negligible fraction of possible protein sequences are functional (Romero & Arnold, 2009); and most experimental wet-lab assays can only provide $10^2 - 10^3$ fitness labels per round. As a result, researchers often rely on directed evolution, an iterative process that incrementally improves protein fitness (Packer & Liu, 2015). In each round of directed evolution, a protein is mutated, the fitness of the variants is measured, and the most beneficial mutation is retained for the next iteration. However, this approach can be slow (often only one mutation is accumulated in each round), and it may be ineffective–as it is limited to a local search of very similar protein sequences.

Thus, in recent years, there has been strong interest in developing adaptive machine learning (ML)-assisted methods to more efficiently optimize protein fitness (Yang et al., 2019; Wittmann et al., 2021a; Yang et al., 2024b; Hie & Yang, 2022). For example, in machine learning-assisted directed evolution (MLDE) approaches (Wu et al., 2019; Wittmann et al., 2021b; Yang et al., 2025; Li et al., 2024), unlabeled data (zero-shot likelihoods based on natural sequences) and labeled fitness data are used to find a sequence with optimal fitness

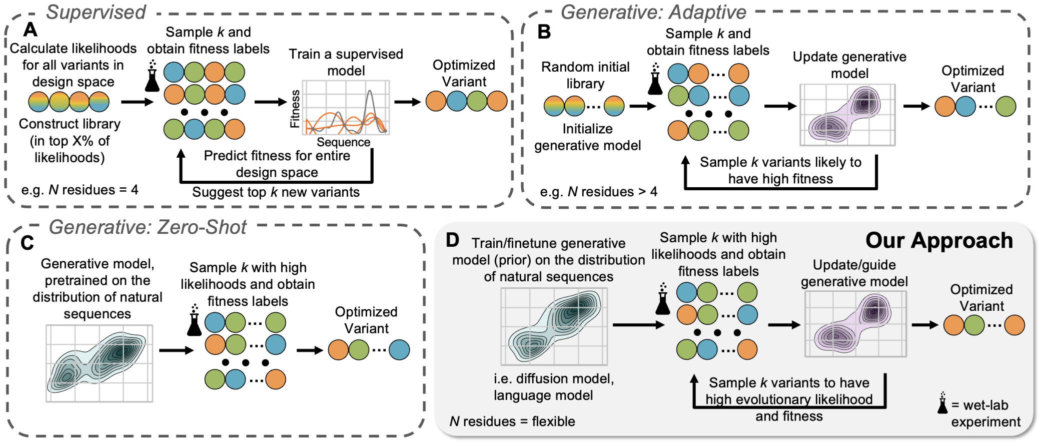
Supervised
A
Calculate likelihoods for all variants in design space
Construct library (in top X% of likelihoods)
Sample k and obtain fitness labels
Train a supervised model
Fitness
Sequence
Optimized Variant
Predict fitness for entire design space
Suggest top k new variants
e.g. N residues = 4
Generative: Adaptive
B
Random initial library
Initialize generative model
Sample k and obtain fitness labels
Update generative model
Optimized Variant
Sample k variants likely to have high fitness
e.g. N residues > 4
Generative: Zero-Shot
C
Generative model, pretrained on the distribution of natural sequences
Sample k with high likelihoods and obtain fitness labels
Optimized Variant
D
Our Approach
Train/finetune generative model (prior) on the distribution of natural sequences
Sample k with high likelihoods and obtain fitness labels
Update/guide generative model
Optimized Variant
i.e. diffusion model, language model
Sample k variants to have high evolutionary likelihood and fitness
N residues = flexible
= wet-lab experiment

Figure 1: **Our approach offers advantages compared to existing approaches for ML-assisted protein fitness optimization.** Optimization is difficult because the the design space is massive and the throughput of wet-lab fitness assays is low. **(A)** Supervised approaches involve enumerating to calculate fitness predictions for all variants design space, limiting them to optimizing on constrained number of residues. **(B)** Generative approaches for finding variants with high fitness do not take advantage of the likelihood of sequences given the natural distribution of proteins (prior); by contrast, **(C)** fully zero-shot methods that sample highly natural sequences do not utilize labeled fitness data. **(D)** Our approach involves initializing a generative model to sample sequences with high likelihoods and iteratively refining or guiding that model with assay labeled fitness data.

by introducing mutations at N residues (*Supervised*, Fig. 1A). While MLDE and related methods (Jiang et al., 2024; Hsu et al., 2022; Ding et al., 2024; Hawkins-Hooker et al., 2024; Zhao et al., 2024) work well in practice for relatively small design spaces ($N \leq 5$), a major limitation of supervised approaches is that they require enumerating predictions of the supervised model and zero-shot likelihoods across the design space of 20^N variants (Table 1), thus becoming computationally intractable as N increases beyond ~ 9.

In contrast, generative methods do not face this limitation; rather than predicting fitness values *in silico*, they learn to sample from a distribution of sequences with high fitness (Wu et al., 2021; Hsu et al., 2024). However, many generative models used in adaptive protein fitness optimization do not use unlabeled data on natural proteins (Brookes et al., 2019; Brookes & Listgarten, 2020; Song & Li, 2024; Stanton et al., 2022; Gupta & Zou, 2019; Hie & Yang, 2022) (*Generative: Adaptive*, Fig. 1B). Alternatively, protein language models (PLMs) trained on the natural distribution of unlabeled sequences can be used as priors to sample sequences with high evolutionary likelihoods (*Generative: Zero-Shot*, Fig. 1C). Although sampling from these priors can yield sequences with generally higher fitness (*e.g.* for antibody binding, Hie et al. 2023), these models do not incorporate any assay-labeled fitness data, which means that the prior may be less informative if fitness deviates from natural function–*e.g.* enzymes being engineered for non-native activities (Arnold, 2018; Yang et al., 2025).

Recent works have aimed to address the need for a generative method utilizing both unlabeled and labeled data (Fig. 1D, Table 1), but applicability to real-world protein fitness optimization is still limited. Broadly, these methods aim to guide or align generative priors of protein *sequences*, such as discrete diffusion models (Alamdari et al., 2023; Wang et al., 2024b) and PLMs (Ruffolo & Madani, 2024), with fitness data. Reinforcement learning (RL) with different generative models has been demonstrated (Widatalla et al., 2024; Stocco et al., 2024; Blalock et al., 2024), and guiding discrete diffusion models has begun to show promise for this task (Nisonoff et al., 2024; Stark et al., 2024; Klarner et al., 2024; Gruver et al., 2023; Lisanza et al., 2024). However, few previous studies have explored alignment with few ($10^2 - 10^3$) labeled sequences (Lisanza et al., 2024; Stocco et al., 2024) for protein *variant* optimization based on real fitness data, *e.g.* activity or fluorescence (Lisanza et al., 2024; Blalock et al., 2024). Moreover, most studies only evaluate one type of generative prior and steering strategy, so a comprehensive comparison of state-of-the-art methods is needed.

Table 1: **Aligning generative models with labeled data offers several advantages compared to existing approaches.** Many current approaches face at least one shortcoming, limiting their broad applicability, but our approach aims to provide a general method for protein fitness optimization addressing individual limitations in other methods. Namely, our approach 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. Note that 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
Supervised	✓	✓	×	Wittmann et al. (2021b); Ding et al. (2024); Hawkins-Hooker et al. (2024); Zhao et al. (2024)
Generative: Adaptive	×	✓	✓	Brookes et al. (2019); Brookes & Listgarten (2020); Song & Li (2024); Stanton et al. (2022)
Generative: Zero-Shot	✓	×	✓	Hie et al. (2023)
Our Approach	✓	✓	✓	Widatalla et al. (2024); Stocco et al. (2024); Nisonoff et al. (2024); Rector-Brooks et al. (2024)

Our study builds upon existing work, and **our main contribution is comparing different types of models and steering strategies for protein variant fitness optimization in a setting that is reflective of real-world engineering** (Fig. 2, Section 2). On the TrpB and CreiLOV protein fitness datasets, we find that our methods enable efficient discovery of protein variants with high fitness, showing potential for future integration into adaptive optimization workflows (Hie & Yang, 2022). We also introduce posterior sampling (Zhang et al., 2024) for this task, a method which shows some of the highest performance. Our results also suggest that guidance with diffusion models may be better than fine-tuning with RL; the latter may not be ideal as it can cause the model to forget prior information and can be computationally expensive. Our code will be made publicly available.

2 Strategies to Steer Generative Models with Labels

2.1 Guidance with Sequence-Based Diffusion Models

Increasingly, various diffusion model (Ho et al., 2020) architectures (Fig. 2) have shown their potential for modeling discrete data, such as sequences (x), approaching the performance of language models and leveraging many similar learning techniques such as masking or autoregressive decoding (Sahoo et al., 2024; Lou et al., 2024). These models can broadly be categorized into two types: those that perform diffusion in a continuous latent space (Li et al., 2022; Chen et al., 2023a; Dieleman et al., 2022; Torres et al., 2025; Meshchaninov et al., 2025) and those that diffuse directly over discrete space. Those performing diffusion in discrete space use a transition matrix to update all discrete states in each timestep (D3PM) (Austin et al., 2023), which has later been formulated as continuous-time Markov chains (Lou et al., 2024; Campbell et al., 2022; 2024), followed by simplified frameworks involving progressive unmasking of the discrete state (such as masked diffusion language models, MDLMs) (Sahoo et al., 2024; Hoogeboom et al., 2022). We elaborate more on these methods in Section A.2.

An advantage of diffusion models is the ability to perform guidance based on fitness labels (y) without finetuning the prior model weights, resulting in reduced training costs and potentially strong signal despite having few ($\sim 10^2$) labels. Sampling (inference) from a con-

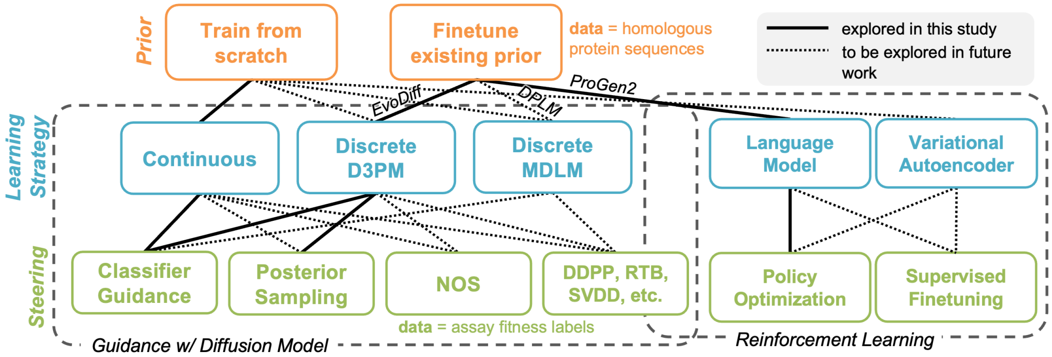

Figure 2: **Non-exhaustive landscape of generative models for natural protein *sequences* and methods to steer/align them with experimental labeled data.** Three major types of diffusion models for sequences include those perform diffusion over continuous space, or those perform diffusion over discrete space, as a continuous time markov chain (D3PM) or with an iterative masking process (MDLM). Various types of guidance strategies are compatible with certain diffusion models, in green (NOS: diffusion optimization sampling, DDPP: discrete denoising posterior prediction, RTB: relative trajectory balance, SVDD: soft value-based decoding in diffusion models). Differently, language models and variational autoencoders can be aligned with labeled data via reinforcement learning such as preference optimization or supervised finetuning.

tinuous diffusion model involves following gradients of the learned denoising function, and these gradients can be updated or modified in various ways. In classifier guidance (Nisonoff et al., 2024), posterior sampling (Chung et al., 2024; Zhang et al., 2024), and diffusion optimized sampling (NOS) (Gruver et al., 2023), the denoising process is guided via related gradients from a supervised predictor model that can predict fitness from sequence representations, $p(y|x_t, t)$. There are many other variations on this guidance process, explained in more detail in Section A.4. In this study, we focus on continuous (trained from scratch) and discrete D3PM (finetuned EvoDiff (Alamdari et al., 2023)) models with classifier guidance and posterior sampling as guidance techniques (Fig. 2). Future work on MDLMs could also utilize the pretrained diffusion protein language model (DPLM) (Wang et al., 2024b) and other recent guidance techniques (Tang et al., 2024; Rector-Brooks et al., 2024).

2.2 Reinforcement Learning

We consider RL broadly here as finetuning generative models such as language models with labeled data about which generations are favorable vs unfavorable. There are emerging RL techniques applied to discrete diffusion models including discrete denoising posterior prediction (DDPP) (Rector-Brooks et al., 2024), relative trajectory balance (RTB) (Venkatraman et al., 2025), and direct reward backpropagation with gumbel softmax trick (DRAKES) (Wang et al., 2024a). 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 applied generally to generative models such as language models (Fig. 2). Policy optimization generally shows better performance than SFT (Stocco et al., 2024; Blalock et al., 2024); direct preference optimization (DPO), while not technically defined as RL, 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., 2024; Stocco et al., 2024), but these methods can have high computational costs of finetuning and may require large amounts of labels ($> 10^3$) to effectively steer generations. In this work, we include DPO with an autoregressive PLM (finetuned ProGen2 (Nijkamp et al., 2023)) as a baseline.

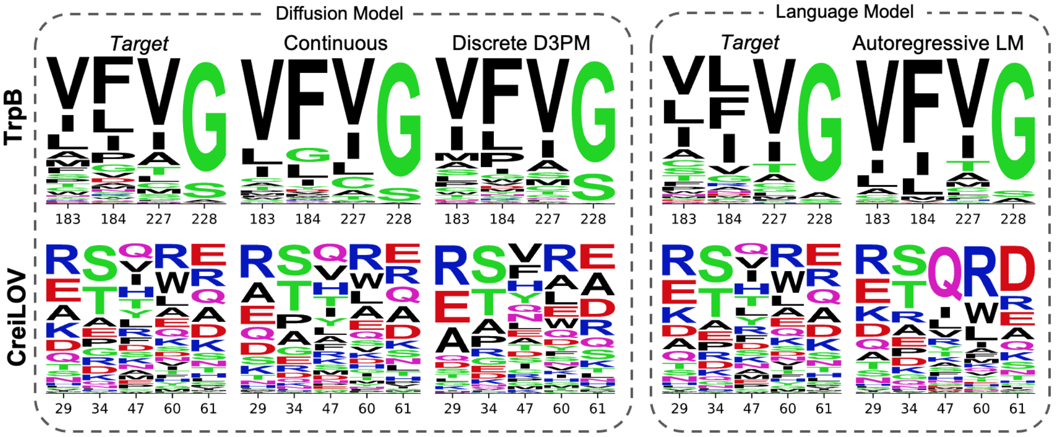

Figure 3: **The distributions of sequences from pretrained generative priors largely match those of the target distributions.** The target distribution shows all sequences in the multiple sequence alignment of the parent, and the distributions of generative models are approximated by sampling 1000 sequences. The continuous model was trained from scratch while the D3PM diffusion model and language model (LM) were finetuned from EvoDiff and ProGen2, respectively. 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). The target distributions for the diffusion model and language model are slightly different due to differences in the usage of gaps during pretraining (Section A.1).

3 Results

Model pretraining recapitulates the distribution of evolutionarily related protein sequences. We focus our fitness optimization studies on two proteins, the TrpB enzyme (length 389) (Johnston et al., 2024) and the CreiLOV fluorescent protein (length 119) (Chen et al., 2023b) due to the availability of fitness data across many residues and the large number of homologous protein sequences found in their multiple sequence alignments. Based on the methods explained in Section A.1, we trained generative priors on these natural sequences for each family, focusing on continuous diffusion models, D3PM diffusion models, and autoregressive language models. Overall, these models capture the natural distribution of protein sequences, with the D3PM models seeming to match the distribution the most closely while also generating sequences with high diversity (Fig. 3).

Pretrained priors and steered/aligned models generate sequences with high fitness. We focused protein fitness optimization to a design space of 15 residues in TrpB and all 119 residues in CreiLOV (details in Section A.3); for each protein's variants, we evaluated fitness by approximating it via a supervised oracle trained on a large amount of real data (Section A.3). We evaluated fitness optimization under two different scenarios: (1) limiting generated sequences to having a maximum of 4 mutations from the parent sequence and (2) unconstrained optimization of all residues in the design space. The former is a more conservative approach to fitness optimization, as our oracles show good performance in this domain (Fig. A1), but we manually satisfy this constraint after unconstrained generation (Section A.3). The latter is ultimately a more generalizable approach, but most mutations are known to be deleterious, meaning that many of these sequences may not be functional in the real world, and the oracle may not accurately extrapolate to these outcomes.

Overall, we found that pretrained priors sample protein variants that are enriched in high fitness, which corroborates previous studies finding that sequences with higher evolutionary likelihood are also likely to have higher fitness (Li et al., 2024; Hie et al., 2023). Impressively, steering/alignment with modest amounts of labeled data (200 sequence-fitness pairs) enabled most models and methods to generate sequences with even higher fitness (Fig. 4). In this regime, guidance with diffusion models (particularly D3PM models) seems to out-

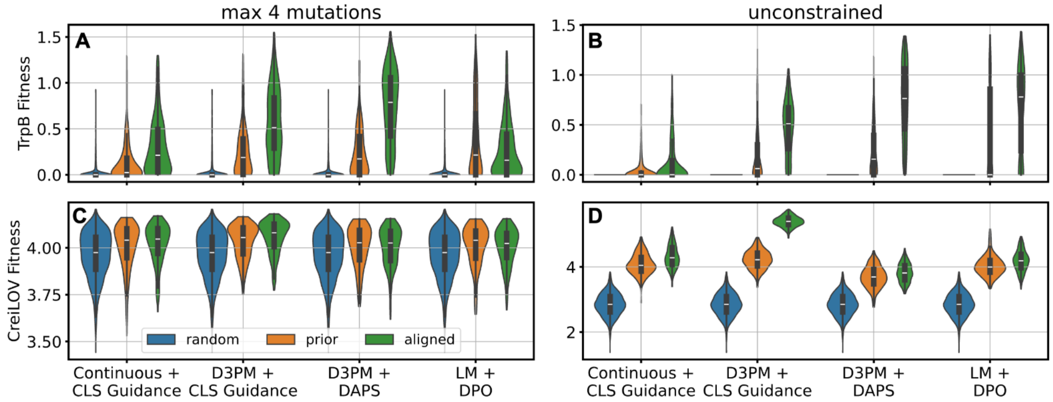

Figure 4: **Sequences sampled from our generative models, before and after steering/alignment with labeled fitness data, are enriched in high-fitness protein variants.** 200 unique sequences were sampled in each round, and models were steered/aligned based on samples drawn from the prior. Variants with negative predicted fitness were rounded up to zero. D3PM: Discrete Diffusion, CLS: Classifier, DAPS: Posterior Sampling, LM: Language Model, DPO: Direct Preference Optimization.

perform DPO with language models, but additional computational experiments are needed to understand this.

4 Discussion

There are many directions for future work. One promising avenue is exploring alternative guidance methods for diffusion models (Fig. 2), particularly MDLM and other models, which have recently achieved state-of-the-art performance in sequence modeling (Sahoo et al., 2024; Peng et al., 2025; Liu et al., 2024). Additionally, we conducted preliminary testing for our approach in an iterative setting, with ten iterations of 100 samples each–similar to batched active learning (Yang et al., 2025; Lisanza et al., 2024). Future work will investigate how increased training samples and increased iterations affects performance (Hie & Yang, 2022). To this end, repeated experiments are needed to conclude which strategies are most effective, and other baselines will be tested. So far, we have focused on proteins with many homologous sequences and fitness as mostly native function, but we will need to test our approach on other protein fitness optimization tasks where the pretrained prior may not provide as much utility. While diffusion models seem to be more amenable and effective for the task explored in this work, further comparisons with different language model architectures, such as masked language models (Blalock et al., 2024), are needed. Furthermore, we manually constrained generations to have few mutations at the end of generation, but implicitly building this guidance into predictor models or sampling techniques (such as inpainting in masked models) may lead to improved performance.

In short, guiding generative models with labeled data is a satisfying and general protein fitness optimization framework, as it takes advantage of knowledge from both the natural protein universe and specific fitness objectives. Overall, we have demonstrated that several implementations of this approach (diffusion and language models with different steering strategies) could be practically useful for protein fitness optimization, laying the groundwork for further exploration.

References

Sarah Alamdari, Nitya Thakkar, Rianne Van Den Berg, Neil Tenenholtz, Bob Strome, Alan Moses, Alex Xijie Lu, Nicolo Fusi, Ava Pardis Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need, September 2023. URL `http://biorxiv.org/lookup/doi/10.1101/2023.09.11.556673`.

Frances H. Arnold. Directed Evolution: Bringing New Chemistry to Life. *Angewandte Chemie International Edition*, 57(16):4143–4148, April 2018. ISSN 1433-7851, 1521-3773. 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, February 2023. URL http://arxiv.org/abs/2107.03006. arXiv:2107.03006 [cs].

Nathaniel Blalock, Srinath Seshadri, Agrim Babber, Sarah Fahlberg, and Philip Romero. Functional Alignment of Protein Language Models via Reinforcement Learning with Experimental Feedback. 2024.

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. pp. 18, 2019.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A Continuous Time Framework for Discrete Denoising Models, October 2022. URL http://arxiv.org/abs/2205.14987. arXiv:2205.14987 [stat].

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, February 2024. URL http://arxiv.org/abs/2402.04997. arXiv:2402.04997 [cs, q-bio, stat].

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog Bits: Generating Discrete Data using Diffusion Models with Self-Conditioning, March 2023a. URL http://arxiv.org/abs/2208.04202. arXiv:2208.04202 [cs].

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 2023b. ISSN 2161-5063, 2161-5063. doi: 10.1021/acssynbio.2c00662. URL https://pubs.acs.org/doi/10.1021/acssynbio.2c00662.

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.

Hyungjin Chung, Jeongsol Kim, Michael T. Mccann, Marc L. Klasky, and Jong Chul Ye. Diffusion Posterior Sampling for General Noisy Inverse Problems, May 2024. URL http://arxiv.org/abs/2209.14687. arXiv:2209.14687 [stat].

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, December 2022. URL http://arxiv.org/abs/2211.15089. arXiv:2211.15089 [cs].

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

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. May 2023. URL http://arxiv.org/abs/2305.20009. arXiv:2305.20009 [cs, q-bio].

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, June 2024. URL http://biorxiv.org/lookup/doi/10.1101/2024.05.28.596156.

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, December 2020. URL http://arxiv.org/abs/2006.11239. arXiv:2006.11239 [cs].

Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive Diffusion Models, February 2022. URL http://arxiv.org/abs/2110.02037. arXiv:2110.02037 [cs].

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.

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. 121(32):e2400439121, June 2024. doi: 10.1101/2024.06.23.600144. URL http://biorxiv.org/lookup/doi/10.1101/2024.06.23.600144.

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.

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, July 2024. URL http://arxiv.org/abs/2407.11942. arXiv:2407.11942 [cs, q-bio, stat].

Francesca-Zhoufan Li, Jason Yang, Kadina E Johnston, Emre Guersoy, Yisong Yue, and Frances H Arnold. Evaluation of Machine Learning-Assisted Directed Evolution Across Diverse Combinatorial Landscapes. *bioRxiv*, 2024.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B. Hashimoto. Diffusion-LM Improves Controllable Text Generation, May 2022. URL http://arxiv.org/abs/2205.14217. arXiv:2205.14217 [cs].

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*, September 2024. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-024-02395-w. URL https://www.nature.com/articles/s41587-024-02395-w.

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, October 2024. URL http://arxiv.org/abs/2410.06264. arXiv:2410.06264 [cs].

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution, June 2024. URL http://arxiv.org/abs/2310.16834. arXiv:2310.16834 [stat].

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.

Viacheslav Meshchaninov, Pavel Strashnov, Andrey Shevtsov, Fedor Nikolaev, Nikita Ivanisenko, Olga Kardymon, and Dmitry Vetrov. Diffusion on language model encodings for protein sequence generation, February 2025. URL http://arxiv.org/abs/2403.03726. arXiv:2403.03726 [cs].

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.

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. June 2024. URL http://arxiv.org/abs/2406.01572. arXiv:2406.01572 [cs].

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, February 2025. URL http://arxiv.org/abs/2502.03540. arXiv:2502.03540 [cs].

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, December 2023. URL http://arxiv.org/abs/2305.18290. arXiv:2305.18290 [cs].

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, October 2024. URL http://arxiv.org/abs/2410.08134. arXiv:2410.08134 [cs].

Philip A Romero and Frances H Arnold. Exploring protein fitness landscapes by directed evolution. 10:866–876, 2009. doi: 10.1038/nrm2805. Publisher: Nature Publishing Group.

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

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, November 2024. URL http://arxiv.org/abs/2406.07524. arXiv:2406.07524 [cs].

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Zhenqiao Song and Lei Li. Importance Weighted Expectation-Maximization for Protein Sequence Design, December 2024. URL http://arxiv.org/abs/2305.00386. arXiv:2305.00386 [q-bio].

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, July 2022. URL http://arxiv.org/abs/2203.12742. arXiv:2203.12742 [cs, q-bio, stat].

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet Flow Matching with Applications to DNA Sequence Design, May 2024. URL http://arxiv.org/abs/2402.05841. arXiv:2402.05841 [cs, q-bio].

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. 2024.

Sophia Tang, Yinuo Zhang, and Pranam Chatterjee. PepTune: De Novo Generation of Therapeutic Peptides with Multi-Objective-Guided Discrete Diffusion, December 2024. URL http://arxiv.org/abs/2412.17780. arXiv:2412.17780 [q-bio].

Marcelo D. T. Torres, Tianlai Chen, Fangping Wan, Pranam Chatterjee, and Cesar de la Fuente-Nunez. Generative latent diffusion language modeling yields anti-infective synthetic peptides, February 2025. URL https://www.biorxiv.org/content/10.1101/2025.01.31.636003v1. Pages: 2025.01.31.636003 Section: New Results.

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, January 2025. URL http://arxiv.org/abs/2405.20971. arXiv:2405.20971 [cs].

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, October 2024a. URL http://arxiv.org/abs/2410.13643. arXiv:2410.13643 [cs].

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion Language Models Are Versatile Protein Learners. *arXiv*, February 2024b. URL http://arxiv.org/abs/2402.18567. arXiv:2402.18567 [cs, q-bio].

Talal Widatalla, Rafael Rafailov, and Brian Hie. Aligning protein generative models with experimental fitness via Direct Preference Optimization. 2024.

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.

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.

Jason Yang, Aadyot Bhatnagar, Jeffrey A. Ruffolo, and Ali Madani. Conditional Enzyme Generation Using Protein Language Models with Adapters. *arXiv*, October 2024a. URL http://arxiv.org/abs/2410.03634. arXiv:2410.03634 null.

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 2024b. ISSN 2374-7943, 2374-7951. URL https://pubs.acs.org/doi/10.1021/acscentsci.3c01275.

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

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, July 2024. URL http://arxiv.org/abs/2407.01521. arXiv:2407.01521 [cs].

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 2024. URL http://biorxiv.org/lookup/doi/10.1101/2024.02.11.579859.

Table A1: **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. While the TrpB dataset has a lot more training labels, it may be more difficult to learn due to 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	389	117, 118, 119, 162, 166, 182, 183, 184, 185, 186, 227, 228, 230, 231, 301	N=15	57,000	75,618	23,313	(Johnston et al., 2024)
CreiLOV	119	All	N=119	370,000	6,842	2,401	(Chen et al., 2023b)

A Appendix

A.1 Data for Pretraining of Priors

The first step in our pipeline involves learning a generative prior on naturally occurring protein sequences to learn 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), either TrpB or CreiLOV. 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 chose the TrpB (Johnston et al., 2024) and CreiLOV (Chen et al., 2023b) datasets due to the extensive number of sequences in their MSAs, which were obtained by running jackhmmer (Johnson et al., 2010) against Uniref90 for two iterations with the parent sequence as target. For the MSA, we only used sequences where the aligned portion was at least 75% the length of the parent sequence. For the diffusion model priors, 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 the language model, the sequences used were the portions of the MSA that was aligned to the parent sequence, with gaps removed, thus resulting in variable length sequence fragments.

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 Generative Models for Sequences

A.2.1 Diffusion over Continuous Space

Diffusion models construct samples by reversing a diffusion process that maps clean data points x_0 to samples from a prior distribution $\pi(x)$. The forward process ($x_0 \rightarrow x_T$) is composed of conditional distributions $p(x_t|x_{t-1})$, which admit closed-form expressions for the conditional distributions $p(x_t|x_0)$ and $p(x_{t-1}|x_t, x_0)$. The reverse process ($x_T \rightarrow x_0$) converts samples from the prior into samples from the learned data distribution $p_\theta(x_0)$ by repeatedly predicting the denoised variable $\hat{x}_0$ from noisy values x_t, using the conditional distribution $p(x_{t-1}|x_t, \hat{x}_0)$ to derive a transition distribution $p_\theta(x_{t-1}|x_t)$.

Continuous Noise Forward Process. Similarly to Gruver et al. (2023), we define a protein sequence as $w \in \mathcal{A}^L$, where $\mathcal{A}$ is the alphabet of amino acids and L is the fixed length of the sequence. To learn a distribution $p(w)$, we first embed w into a continuous variable x_0 using an embedding matrix U_θ, transforming discrete tokens into a continuous latent space. Gaussian noise is then applied to this embedding space. The prior distribution is defined as:

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

while the forward process follows a Gaussian corruption schedule:

$$p(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)I), \quad \bar{\alpha}_t = \prod_{i=1}^{t}\alpha_i, \quad \alpha_t = 1-\beta_t. \tag{2}$$

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 $p_\theta(\hat{w}|x_t, t)$ that predicts the sequence from noised points x_t. This is done by minimizing the following objective:

$$L(\theta) = \mathbb{E}_{w_0,t}\left[-\log p_\theta(w_0|x_t)\right], \quad x_t \sim p(x_t|x_0 = U_\theta w_0). \tag{3}$$

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

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

where the posterior $p(x_{t-1}|x_t, x_0)$ follows:

$$p(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \mu_t, \sigma_t^2 I), \tag{5}$$

with mean μ_t and variance σ_t^2 given by:

$$\mu_t = \frac{\sqrt{\bar{\alpha}_{t-1}}\beta_t}{1-\bar{\alpha}_t}x_0 + \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}x_t, \tag{6}$$

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

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:

$$x_{t-1} \sim p_\theta(x_{t-1}|x_t), \tag{8}$$

and then reconstructing w by sampling:

$$w \sim p_\theta(\hat{w}|x_0). \tag{9}$$

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

Training and Hyperparameters. The continuous diffusion model was trained using a cosine noise schedule with $T = 500$ diffusion steps, following the improved variance schedule from Nichol & Dhariwal (2021). The embedding dimension was set to 64, and the denoising network was a transformer encoder with 12 layers and 8 attention heads, based on the NOS-C model from Gruver et al. (2023). We used the AdamW optimizer with a learning rate of 10^{-4}, linear learning rate schedule, and weight decay of 0.01 for 5 epochs of training, taking the model with the lowest validation loss.

A.2.2 Diffusion over Discrete Space.

Discrete diffusion models (Austin et al., 2023; 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 $p_0 = p_{\text{data}}$ is the data distribution and $\boldsymbol{Q}_t \in \mathbb{R}^{N\times N}$ are predefined transition matrices. In this work, we use uniform transition matrices $\boldsymbol{Q}_t = \frac{1}{N}\boldsymbol{1}\boldsymbol{1}^T - \boldsymbol{I}$. When $T \to \infty$, the probability distribution p_T converges to a uniform distribution.

This Markov process can be reversed with the help of a concrete score function, $s(x,t) := [\frac{p_t(\tilde{x})}{p_t(x)}]_{\tilde{x}\neq x}$, 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 $\bar{\boldsymbol{Q}}_t[\tilde{x},x] = s(x,t)_{\tilde{x}}\boldsymbol{Q}_t[x,\tilde{x}]$ for $\tilde{x} \neq x$, and $\bar{\boldsymbol{Q}}_t[x,x] = -\sum_{\tilde{x}\neq x}\bar{\boldsymbol{Q}}_t[\tilde{x},x]$. To generate data $x_0 \sim p_{\text{data}}$, we start with sampling x_T from a uniform distribution and then evolve through Eq. 11 by the Euler method.

Training and Hyperparameters. We used the 38 million parameter ByteNet architecture from Alamdari et al. (2023) (EvoDiff) to model the score function and finetuned from their final checkpoint. This model uses a hidden dimension of 1024, 16 layers, and a kernel size of 5. Diffusion was performed over 500 timesteps using the uniform noise scheduler from EvoDiff. We used the AdamW optimizer with a learning rate of 10^{-4}, and linear learning rate schedule for 5 epochs of training, taking the model with the lowest validation loss.

A.2.3 Autoregressive Language Models.

In this work, we finetuned the ProGen2-base decoder-only transformer (780 million parameters) based on the code and parameters used in Yang et al. (2024a). Models were trained based on next token prediction and cross entropy loss. We used the AdamW optimizer with a learning rate of 10^{-4}, and linear learning rate schedule, for 10 epochs of training, taking the model with the lowest validation loss. However, we did not use adapter layers, and we did not group batches based on sequence length.

A.3 Protein Fitness Optimization Task

We studied fitness optimization across two different protein-fitness datasets, TrpB and CreiLOV (Table A1). 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 rate catalytic activity 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. (2024). 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 using for validation and testing, respectively. Our model consists of an ensemble of 20 MLPs for TrpB, and each was trained on onehot encodings of the designed residues for 1000 epochs. From here forth, we treated ground truth fitness for TrpB as outputs from the oracle.

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. (2024) 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. From here forth, we treated ground truth fitness for CreiLOV as outputs from the oracle. Our oracles have high Pearson correlation on the test set (Fig. A1)

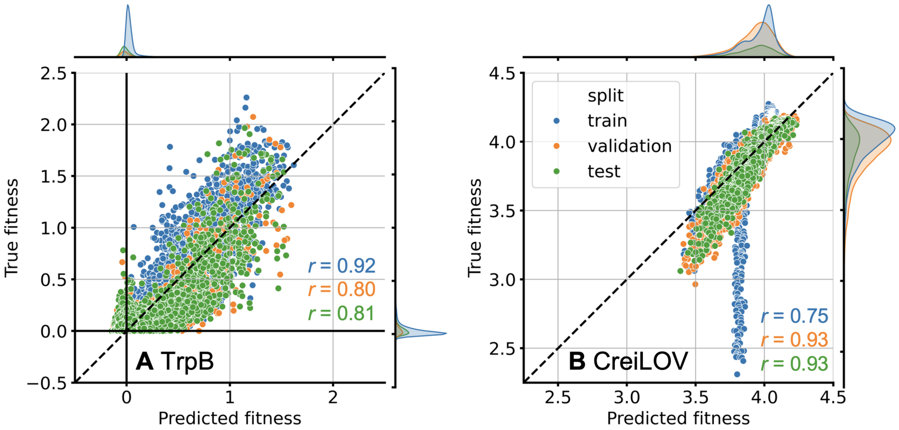

Figure A1: Oracles trained on available labeled data for TrpB and CreiLOV extrapolate well to higher order combinations of mutations within the design space, as measured by Pearson correlation.

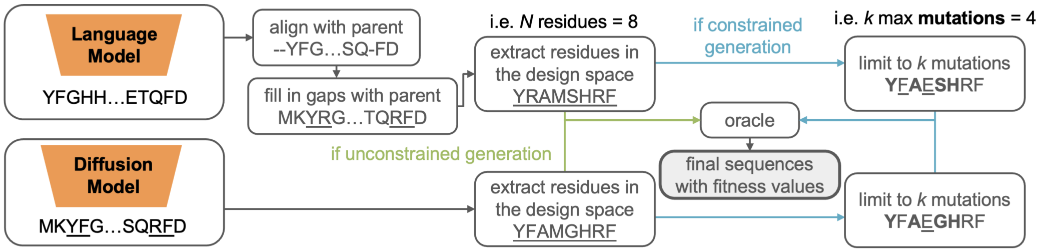

Figure A2: Pipeline for generating protein sequences for evaluation, based on a hypothetical parent sequence: MKKFG...SQRFD (length=100), with 8 residues being optimized (3, 4, 26, 27, 28, 29, 98, 99), corresponding to a design space combo of KFDEACRF.

Our primary method for evaluation involved examining the distribution of sampled sequences and their corresponding fitness values. 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 model, sequence fragments of variable length were generated and 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. 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. For unconstrained generation, we allowed the entire design space to be mutated. In constrained generation, we allowed for a maximum of 4 mutations in any generated sequence, relative to parent, as the generalization ability of our oracle has only been tested on variants that are similar to the parent. We enforced this constraint at the end of generation by choosing 4 mutated residues randomly to keep, and other residues were mapped back to their corresponding identities in the parent sequence.

For the plots in Fig. 3, 1000 sequences were generated and duplicates were not removed. For the language model, we used a temperature of 1.0 and a top-p of 1.0. For Fig. 4, to evaluate the fitness distribution of generated sequences, duplicate sequences were removed, and unique sequences were generated until a certain threshold was met: 200 sequences. We used the same generation parameters for generation from the aligned (conditional) models, which were aligned with sequence-fitness pairs obtained from an initial round of 200 sequences from the unconditional prior.

A.4 Steering Methods

A.4.1 Classifier Guidance

Classifier guidance (Song et al., 2020) is a technique used to steer samples generated by diffusion models toward desired attributes. The primary goal is to sample from a conditional distribution $p(x|y)$, where y is a guiding signal of interest. In continuous space, this can be achieved by replacing the unconditional score function $\nabla_{x_t} \log p_t(x_t)$ at time t by a conditional score function,

$$\nabla_{x_t} \log p(x_t|y) = \nabla_{x_t} \log p_t(x_t) + \nabla_{x_t} \log p_t(y|x_t). \tag{12}$$

To obtain the conditional score function, one only needs to train a time-dependent predictor, which predicts the probability of $p_t(y|x_t)$ given 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 x_t by incorporating a classifier score $\nabla_{x_t} \log p(y|x_t)$ into the model's learned score function. This is based on the fact that the conditional score function can be rewritten as:

$$\nabla_{x_t} \log p(x_t|y) = \nabla_{x_t} \log p_t(x_t) + \lambda \nabla_{x_t} \log p_t(y|x_t), \tag{13}$$

where λ is the guidance scale controlling the influence of the classifier.

Following Song et al. (2020), the classifier guidance term modifies the predicted $\hat{x}_0$ in the denoising process:

$$\hat{x}_0 = x_t + \sigma^2(s_\theta(x_t, t) + \lambda \nabla_{x_t} \log p(y|x_t)). \tag{14}$$

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

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

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

  $$\hat{x}_0 = \sum_{\hat{w}} p(\hat{w}|x_t, t) U_\theta \hat{w} \tag{15}$$

  where U_θ is the embedding matrix mapping discrete tokens to continuous space.
- If a classifier D is available, compute the classifier guidance term:

  $$\nabla_{x_t} \log p(y|x_t) = \frac{1}{\tau} \nabla_{x_t} D(x_t, t). \tag{16}$$

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

  $$\hat{x}_0 = \hat{x}_0 + \lambda \sigma^2 \nabla_{x_t} \log p(y|x_t). \tag{17}$$

This allows the diffusion model to generate samples that are more likely to satisfy the desired condition y.

To obtain a classifier D for continuous diffusion models, we trained an MLP predictor with a hidden dimension of 264 to predict the fitness of a continuously embedded sequences given x_t and time t. We trained the classifier with 1000 epochs of Adam optimization with a learning rate of 10^{-3}. For classifier guidance with continuous models, we used a guidance weight of $\lambda = 100$.

Discrete Guidance. Nisonoff et al. (2024) 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 $\boldsymbol{R}_t^y$ with

$$\boldsymbol{R}_t^y[x, \tilde{x}] = \frac{p(y|\tilde{x}, t)}{p(y|x, t)} \bar{\boldsymbol{Q}}_t[x, \tilde{x}], \ \forall \tilde{x} \neq x. \tag{18}$$

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

To obtain a classifier D for discrete diffusion models, we trained an MLP predictor with a hidden dimension of 64 to predict the fitness of a one-hot encoded sequence given x_t and time t. We trained the classifier with a uniformly random $t \in [0, T]$ over 1000 epochs of Adam optimization with a learning rate of 10^{-3}. For classifier guidance with D3PM models, we used a guidance temperature of $\tau = 0.01$.

A.4.2 Posterior Sampling

Another line of guidance work (Chung et al., 2023; Mardani et al., 2024; Zhang et al., 2024) focuses on drawing samples from the posterior distribution $p(x|y) \propto p(x)p(y|x)$, where the prior distribution is modeled by a pretrained diffusion model. The conditional distribution $p(y|x)$ can either be the likelihood function of a forward model (i.e., when y is an incomplete measurement of x) or an exponential distribution with respect to a reward function (i.e., $p(y|x) \propto \exp(r(x)/\tau)$). The major difference between posterior sampling and classifier guidance is that it requires the reward function to be trained only on clean data 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., 2024) to enable diffusion posterior sampling in discrete-state spaces. Suppose x lies in a finite support $\mathcal{X}^D$, we follow the following steps:

- Initialize $x_T \sim \text{Uniform}(\mathcal{X})^D$
- for $i = 1, \ldots, K$
    1. Sample $\hat{x}_0^{(i)} \sim p(x_0|x_{t_{i-1}})$ by a discrete diffusion model.
    2. Run Metropolis Hasting algorithm to sample $x_0^{(i)} \sim p(x_0|x_{t_{i-1}}, y)$ as defined in Eq. 19.
    3. Sample $x_{t_i} \sim p(x_{t_i}|x_0)$ following the forward Markov process.
- Return x_K.

Specifically, $t_0, t_1, \ldots, t_K$ are mono-decreasing time steps with $t_0 = T$ and $t_K \approx 0$. $p(x_0|x_t, y)$ is defined as

$$\begin{aligned} p(x_0|x_t, y) &\propto p(y|x_0)p(x_0|x_t) \\ &\approx p(y|x_0)\exp(-\|x_0 - \hat{x}_0(x_t)\|_0/\sigma_t), \end{aligned} \tag{19}$$

where $\hat{x}_0(x_t) \sim p(x_0|x_t)$ is a point estimate of the conditional distribution, and we approximate $p(x_0|x_t)$ by an exponential distribution over Hamming distance. Following Proposition 1 in Zhang et al. (2024), $\hat{x}_0^{(i)}$, $x_0^{(i)}$, and x_{t_i} converge to the posterior distribution as t_i goes to 0.

For posterior sampling with the D3PM model and DAPS, we obtained the reward model D using the same model architecture and training parameters as discrete guidance but only trained on clean data x (no noised x_t). We used a guidance temperature $\tau = 0.1$ and 1000 Metropolis Hasting steps in each iteration. We set $K = 500$ using the same time scheduler $\{t_k\}_{k=0}^K$ as D3PM.

A.4.3 Policy Optimization

For DPO with language models, we used the ranked loss function from Widatalla et al. (2024) and Stocco et al. (2024) (Eq. 20). π_θ is the policy to be updated, π_{ref} is the original model, and β is a tunable parameter describing the extent of drift from the reference model. The loss therefore describes the cross entropy of the ratio $\beta \log \frac{\pi_\theta(x)}{\pi_{\text{ref}}(x)}$ and the fitness value w. Following Stocco et al. (2024), we calculated the ratio r as the difference of the log likelihood

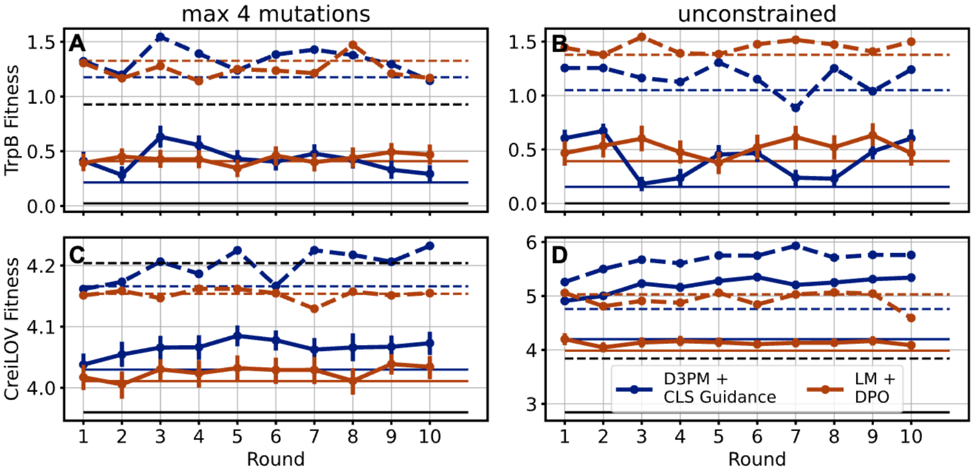

Figure A3: Performance generally improves over multiple iterations of alignment, but this trend is more obvious for the CreiLOV dataset. 100 sequences were sampled in each round, and the model was aligned with all accumulated previous samples. Solid lines represent the mean fitness of samples in the round, while dashed lines represent the maximum fitness of a sequences sampled during the round. The horizontal lines show baselines achieved by the prior before alignment, and the black lines are random. This task will be explored more in future work.

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) but increased the learning rate to 10^{-4} and the β parameter to 1 with finetuning for 5 epochs. In future work, we will explore other DPO approaches such as paired and weighted loss with other types of models.

$$L_{\text{DPO}_{\text{ranked}}}(\pi_\theta;\pi_{\text{ref}}) = -\mathbb{E}_D \sum_{k=1}^{K} \left[\beta \log \frac{\pi_\theta(x)}{\pi_{\text{ref}}(x)} - \log \sum_{j=k}^{K} \exp\left(\beta \log \frac{\pi_\theta(x)}{\pi_{\text{ref}}(x)} \right) \right] \tag{20}$$

A.5 Additional Results