Plug & Play Directed Evolution of Proteins with Gradient-based Discrete MCMC

Patrick Emami[1,*] **Aidan Perreault**[1,2] **Jeffrey Law**[1] **David Biagioni**[3] **Peter C. St. John**[1]

[1]National Renewable Energy Lab [2]Stanford University [3]Maplewell Energy

Abstract

A long-standing goal of machine-learning-based protein engineering is to accelerate the discovery of novel mutations that improve the function of a known protein. We introduce a plug and play framework for evolving proteins *in silico* that supports mixing and matching a variety of unsupervised evolutionary models with supervised models to help constrain search to regions likely to contain functional proteins. Our framework achieves this by sampling from a product of experts distribution defined in discrete protein space and does not require any model fine-tuning or re-training. Instead of resorting to brute force or random search, as is typical of previous plug and play algorithms for protein engineering, we derive a fast discrete sampler that uses gradients to identify promising mutations. Our *in silico* directed evolution experiments on wide fitness landscapes show that we efficiently discover variants with high evolutionary likelihood and estimated activity that are multiple mutations away from the wild type protein. Our framework is also analyzed across a range of different unsupervised evolutionary models including a 650M parameter protein language model.

1 Introduction

Engineering proteins to improve their productivity or catalyze new reactions requires scientists to navigate the complex landscape mapping a protein's amino acid sequence to its structure and function (Li et al., 2020). *Directed evolution* is a classic approach inspired by natural evolution where random mutations to a protein's sequence are screened until higher-performing variants are found, at which point the process repeats starting from these variants (Kuchner & Arnold, 1997). However, this becomes impractical when a protein's activity cannot be assessed in a high-throughput fashion. Brute force search only explores a limited (single or double) mutation window size; for a protein with 400 amino acids, there are $\sim 10^{19}$ ways to make five single substitutions assuming the standard vocabulary of 20 amino acids. It is also remarkably difficult to find proteins with improved function. Most of the protein space is non-functional and beneficial mutations are rare (Arnold, 1998).

As supervised machine learning for predicting protein function from primary sequence improves (Dallago et al., 2021; Hsu et al., 2022), *machine-learning-based directed evolution* has emerged as a way to improve candidate selection between design rounds, with aims of reducing time spent in the wet lab (Yang et al., 2019; Wu et al., 2021; Biswas et al., 2021). We argue that the promising performance of recent unsupervised evolutionary sequence models for mutation effect prediction (Meier et al., 2021; Hsu et al., 2022; Weinstein et al., 2022) has motivated a reconsideration of simple black-box algorithms as a way to mix and match unsupervised and supervised models for candidate selection without requiring any fine-tuning or retraining. Combining both types of models is advantageous because unsupervised evolutionary models learn information that, for e.g., is useful for steering search away from adversarial sequences that can fool supervised models due to

[*]Corresponding email: pemami@nrel.gov

Machine Learning for Structural Biology Workshop, NeurIPS 2022.

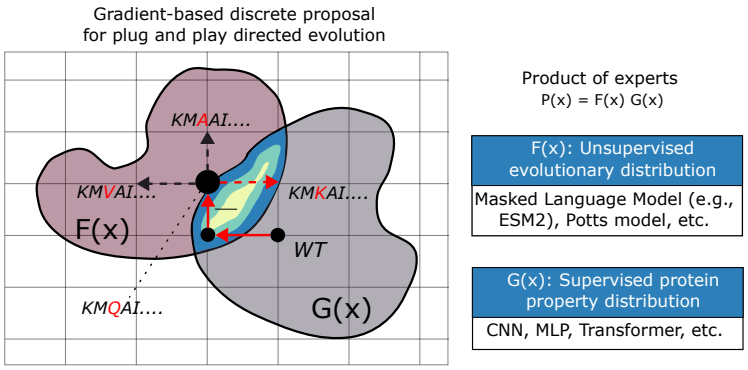


Figure 1: Our goal is to improve the candidate selection step between design rounds of directed evolution. To that end, we introduce a plug and play sampler for discovering high-fitness variants near a wild-type (WT) protein. Our framework assigns probabilities to proteins with a product of experts distribution, which allows us to easily compose unsupervised evolutionary protein models with supervised models of protein fitness. We derive a fast gradient-based discrete MCMC sampler to efficiently sample from this distribution. An example MCMC trajectory of single amino acid mutations is shown in red. Each protein is a sequence of amino acids, e.g., "KMQAI".

overestimation errors (Szegedy et al., 2014). Supervised models contain specific information about beneficial mutations gleaned from assay-labeled data. Black-box algorithms are particularly appealing for "plug and play" search due to their simplicity, flexibility, compatibility with discrete spaces, and use of interpretable mutation operators. However, they converge extremely slowly to variants with high predicted fitness. A simple plug and play framework for directed evolution has remained elusive due to the difficulty of fast search in discrete, high-dimensional spaces.

To that end, this paper introduces Plug and Play Directed Evolution (PPDE), a method for searching for high-fitness variants of a wild-type (WT) protein directly in discrete protein space. PPDE flexibly combines unsupervised evolutionary models and supervised models by using a *product of experts* distribution (Hinton, 2002) $p(x) = \prod_i^M p_i(x)$. Combining both types of protein models in this manner encourages variants to have both high evolutionary likelihood and high predicted activity. We sample efficiently from the high-dimensional, discrete, and unnormalized distribution $p(x)$ by deriving a fast Markov chain Monte Carlo (MCMC) sampler that uses *gradients* of $p(x)$ to propose mutations (Figure 1). PPDE nearly maintains all of the characteristics of black-box algorithms—it additionally assumes that $p(x)$ is differentiable at each discrete point of protein space. We characterize the efficiency of this sampler and empirically show that our framework works with a variety of pre-trained unsupervised evolutionary models *without continuous relaxations and without retraining or fine-tuning*. Although our formulation can be applied to a broad class of biological sequence design tasks, we focus on proteins due to the current widespread availability of pre-trained models.

1.1 Background

Before describing our sampler for plug and play directed evolution, we first introduce gradient-based discrete MCMC. Let $f(x)$ be the unnormalized log probability of x such that $\log p(x) = f(x) - \log Z$ where $Z = \sum_{x \in X} \exp(f(x))$ is a normalizing constant. Uninformed Metropolis Hastings (MH) (Metropolis et al., 1953; Hastings, 1970) proposals such as a uniform distribution are often inefficient for sampling from high-dimensional discrete distributions since candidate states x' are proposed "blindly". In general, the efficiency of MH algorithms is highly dependent on the choice of proposal distribution. MH with locally-balanced informed proposals (Zanella, 2020) uses distributions of the form

$$q(x'|x) \propto \exp(f(x') - f(x))^{\frac{1}{2}} \mathbf{1}(x' \in \mathcal{N}(x)). \tag{1}$$

This proposal is biased towards local state transitions that incur an increase in likelihood[2]. Since enumerating all local moves in $\mathcal{N}(x)$ in discrete high-dimensional spaces is infeasible, an effi-

[2]We take the square root of the exponential in this proposal as the choice of local balancing function $w(t) = tw(1/t), \forall t > 0$. This function "balances" the acceptance and rejection probabilities in the local neighborhood $\mathcal{N}(x)$ to achieve a high acceptance rate. The square root $w(t) = \sqrt{t}$ was empirically validated as a good default option in Zanella (2020).

cient alternative is available for functions $f(x)$ whose gradient can be evaluated at the discrete state x (Grathwohl et al., 2021). A gradient-based locally-balanced informed proposal for discrete MCMC uses a first-order Taylor-series approximation around each x' in the neighborhood $\mathcal{N}(x)$ to bias the proposal towards promising next states:

$$\tilde{q}(x'|x) \propto \exp\left(\frac{1}{2}\nabla_x f(x)^T(x' - x)\right)\mathbf{1}(x' \in \mathcal{N}(x)). \tag{2}$$

When $\mathcal{N}(x)$ is the 1-Hamming ball, this proposal amounts to a tempered softmax over single changes to one dimension of x. One forward pass and one backwards pass is required to compute the forward $\tilde{q}(x'|x)$ and reverse $\tilde{q}(x|x')$ approximate proposals.

2 Plug and Play Directed Evolution

We consider the problem of searching for mutations that improve a target property of a given WT protein. This search problem is defined over discrete protein sequences $x := \{x_0, \ldots, x_{L-1}\}$, $x \in X$, of length L with each x_i taking on a value in a vocab of size V (typically $V = 20$ for the 20 standard amino acids). We assume that each x_i is one-hot encoded.

We now present our probabilistic framework for sampling from a combination of unsupervised and supervised sequence models without requiring any re-training or fine-tuning. We construct a distribution that has its probability mass on proteins that have both high *evolutionary density* $f(x)$ (i.e., high likelihood of being a naturally occurring protein) and high predicted activity $g(x)$ by taking the product of multiple pre-trained "expert" distributions:

$$\log p(x) \propto \sum_i f_i(x) + \lambda \sum_j g_j(x). \tag{3}$$

Each $f_i(x)$ is an unsupervised evolutionary model and each $g_j(x)$ is a supervised model. Typically, $f_i(x)$ has been trained to do density estimation on unlabeled yet aligned sequences (e.g., multiple sequence alignments (MSAs)) or unaligned sequences, while $g_j(x)$ may be an ensemble of nonlinear regressors trained to predict activity on a labeled dataset of mutants. The unsupervised experts $p_f(x) \propto \prod_i \exp\left(f_i(x)\right)$ act as a soft constraint that keeps the sampler near regions of high evolutionary density and away from, e.g., adversarial optima of the supervised models. Examples include the EVmutation Potts model (Hopf et al., 2017) and the ESM protein language models (Rives et al., 2021; Lin et al., 2022). The supervised experts $p_g(x)$ are soft constraints that guide sampling towards proteins that have high activity, where $p_g(x)$ is a Boltzmann distribution $p_g(x) \propto \prod_j \exp\left(\lambda g_j(x)\right)$ that assigns high probability to sequences with high activity. The hyperparameter λ allows us to balance the contribution of the unsupervised and supervised experts.

2.1 Product of experts gradient-based discrete MCMC

Sampling from the product of experts (Equation 3) is difficult since $\log p(x)$ is unnormalized. We derive a fast gradient-based discrete MCMC sampler for $\log p(x)$ by assuming that each expert is a continuous function that is differentiable at each discrete $x \in X$ (e.g., as is the case when the experts are neural networks). During each step of MCMC, we use the gradient of each expert to approximate the likelihood change in the local neighborhood $\mathcal{N}(x)$ to bias the mutation proposal towards promising mutations.

We adapt the gradient-based *path proposal* from Sun et al. (2022), which is capable of performing large jumps in protein space per step of MCMC. In detail, our path proposal with path length $R \sim \text{Unif}(1, U)$ is $\tilde{q}_R(x'|x) = \prod_{r=1}^{R} \tilde{q}\left(x^r|x^{r-1}\right)$ where $\tilde{q}\left(x^r|x^{r-1}\right)$ is

$$\exp\left(\frac{1}{2}\sum_{i=1}^{M}\nabla_x f_i(x)^T(x^r - x^{r-1}) + \lambda\sum_{j=1}^{N}\nabla_x g_j(x)^T(x^r - x^{r-1})\right)\mathbf{1}(x^r \in \mathcal{N}(x^{r-1})). \tag{4}$$

We use this path proposal to sample R single amino acid substitutions, which we apply to the current protein $x := x^0$ at each step of MCMC. The terminal state of a single path $x' := x^R$ is the variant that results from the accumulation of the R substitutions. To avoid computing extra forward and backwards passes at intermediate path states, following Sun et al. (2022) we re-use the gradient

Algorithm 1 Plug and Play Directed Evolution (PPDE)

input one-hot encoded wild-type protein x^{WT}, unsupervised experts f_i, supervised experts g_j, scale λ, max path length U

output evolved protein x

define $x := x^0$, $x' := x^{R-1}$, $\pi(x) := \sum_i f_i(x) + \lambda \sum_j g_j(x)$

while *still searching* **do**

`// compute the forward path proposal distribution`

sample path length $R \sim \text{Unif}(1, U)$

for $r \in \{1, \ldots, R-1\}$ **do**

$$d(x^r|x^{r-1}) = \sum_i \nabla_x f_i(x)^T (x^r - x^{r-1}) + \lambda \sum_j \nabla_x g_j(x)^T (x^r - x^{r-1})$$

$$\tilde{q}(x^r|x^{r-1}) = \text{categorical}\left(\text{softmax}\left(\frac{d(x^r|x^{r-1})}{2}\right)\right)$$

`// sample a single amino acid substitution and apply`

$x^r \sim \tilde{q}(x^r|x^{r-1})$

`// compute the reverse path proposal distribution for` $r = R-1, \ldots, 1$

$$d(x^{r-1}|x^r) = \nabla_{x'} \sum_i f_i(x')^T (x^{r-1} - x^r) + \lambda \nabla_{x'} \sum_j g_j(x')^T (x^{r-1} - x^r)$$

$$\tilde{q}(x^{r-1}|x^r) = \text{categorical}\left(\text{softmax}\left(\frac{d(x^{r-1}|x^r)}{2}\right)\right)$$

`// accept` x' `with probability`

$$\min\left\{1, \exp(\pi(x') - \pi(x)) \frac{\prod_{r=R-1}^{1} \tilde{q}(x^{r-1}|x^r)}{\prod_{r=1}^{R-1} \tilde{q}(x^r|x^{r-1})}\right\}$$

taken with respect to the path origin x^0. The same is done for the reverse path proposals $\tilde{q}(x^{r-1}|x^r)$ with respect to the terminal state x^R. Algorithm 1 shows pseudo-code for our fast MCMC sampler for plug and play directed evolution of proteins. The sampler follows the basic structure of MH MCMC. At each sampler step, we first compute the forward path proposal distribution $\tilde{q}_R(x'|x)$ which we use to sample the proposed protein x'. Then, we compute the reverse path proposal distribution $\tilde{q}_R(x|x')$. We use these probabilities to compute an acceptance probability for determining whether to accept or reject x', after which the process repeats until termination.

2.2 Sampler Analysis

The following corollary to Theorem 3 from Sun et al. (2022) relates the smoothness of each expert to the sampler's ability to efficiently explore protein space.

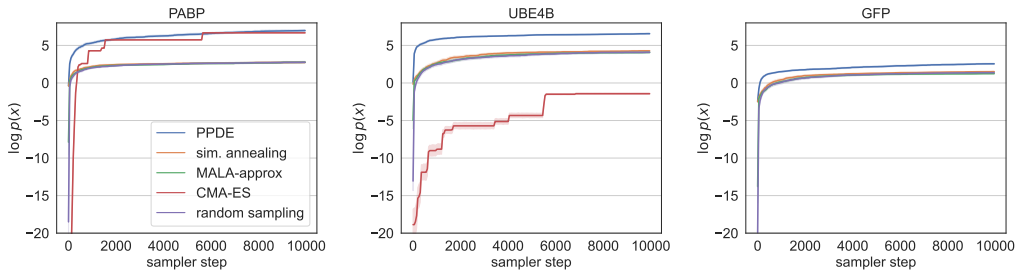
Corollary 1 *Assume each expert f_i is differentiable, $\nabla_x f_i(x)$ is K_i-Lipschitz, the max path length is U, and a 1-Hamming ball neighborhood $\mathcal{N}(x)$. Let $Q_R(x, x')$ and $\tilde{Q}_R(x, x')$ be the Markov transition kernels induced by our sampler with the product of experts proposal $q_R(x'|x)$ and with its approximation $\tilde{q}_R(x'|x)$, respectively. These transition kernels are related by*

$$\tilde{Q}_R(x, x') \geq \left(\prod_{i=1}^{M} e^{-K_i \frac{U(U+1)}{2}}\right) Q_R(x, x'). \tag{5}$$

See Appendix A.1 for the proof. This result tells us that a single expert whose gradient has a large Lipschitz constant could greatly reduce the overall efficiency of the sampler.

3 *In silico* Directed Evolution Experiments

We use the three benchmark proteins from Hsu et al. (2022) with higher-order mutants as well as their provided MSAs for our *in silico* directed evolution experiments. The Poly(A)-binding protein (PABP) dataset of variants measuring binding activity (95 residue subsequence, each variant has $\leq$ 2 mutations), the ubiquitination factor E4B (UBE4B) protein dataset measuring ligase activity (103 residue subsequence, each variant has ≤ 6 mutations), and GFP protein dataset measuring fluorescence (237 residues, each variant has ≤ 15 mutations). To emulate a realistic protein engineering



(a) PABP (b) UBE4B (c) GFP

Figure 2: Cumulative maximum product of experts log probability averaged across the population.

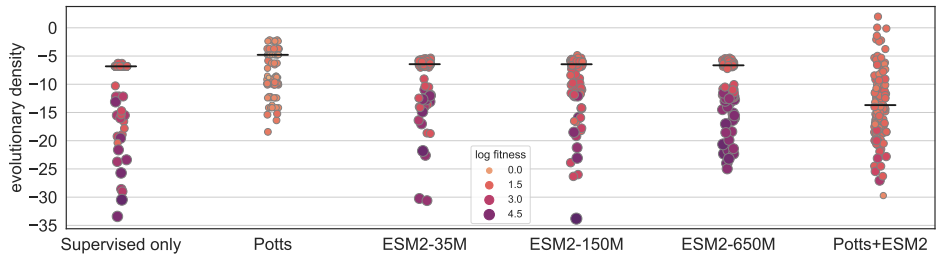


Figure 3: Plugging in different unsupervised experts into PPDE for UBE4B.

setup with reasonable amounts of data for training our supervised experts $g(x)$, we use the "2-vs-rest" mutation train/test split suggested in Dallago et al. (2021)—that is, after splitting each dataset 80/20, we keep only sequences with two or fewer mutations for training and subsample 10% of these. This amounts to 3K sequences for PABP, 2K for UBE4B, and 1K for GFP. We use the EVmutation Potts and ESM2 family of protein language models as pre-trained unsupervised experts.

As baselines, we use `simulated annealing` with random mutation proposals as in Biswas et al. (2021). We also consider a simpler variation (`random sampling`) without acceptance criteria. `CMA-ES` (Hansen & Ostermeier, 1996) is a black-box evolutionary algorithm for continuous spaces and `MALA-approx` (Nguyen et al., 2017) is gradient-based MCMC for continuous spaces. Both require continuous relaxations. We run each sampler from the WT 128 times for 10K steps and use the sample with the highest $\log p(x)$ to assemble the final population of 128 variants. Activity of proteins (log fitness) is scored relative to WT with the Augmented EVmutation Potts oracle from Hsu et al. (2022) trained on all mutants in each 80% train split. We use a different transformer, the MSA Transformer (Rao et al., 2021), conditioned on 500 randomly subsampled sequences from each protein's provided MSA as an evolutionary density score relative to WT. We also calculate the population diversity and average number of mutations from WT. Higher is better for all metrics.

Results: See Tables 1,2 in the appendix for numerical results. PPDE enables efficient discovery of promising mutants *farther away from WT* than random-walk-based baselines. Figure 2 shows that, across all three proteins, PPDE most rapidly and effectively explores the product of experts distribution. Figure 3 highlights the plug and play ability of PPDE.

4 Conclusions

In this study, we have shown how to flexibly combine unsupervised models of evolutionary density and supervised models of protein fitness and how to efficiently sample from the resulting distribution to discover proteins that maximize a desired function while avoiding poor local optima. This strategy leverages the vast amounts of unlabeled data that are available for unsupervised pre-training to improve generated sequences, even when relatively few labelled data are available for training the fitness function. Future work may extend this framework to larger problems in biological design. For instance, the simultaneous engineering of several sequences in multimeric enzyme complexes, or incorporating substrate structure in evaluating the likelihood of enzyme-substrate complexes.

Acknowledgments

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Bioenergy Technologies Office and the Laboratory Directed Research and Development (LDRD) Program at NREL. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

Arnold, F. H. Design by directed evolution. *Accounts of chemical research*, 31(3):125–131, 1998.

Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M., and Church, G. M. Low-n protein engineering with data-efficient deep learning. *Nat Methods*, 18(4):389–396, 2021. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-021-01100-y. URL http://www.nature.com/articles/s41592-021-01100-y.

Dallago, C., Mou, J., Johnston, K. E., Wittmann, B. J., Bhattacharya, N., Goldman, S., Madani, A., and Yang, K. K. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, 2021.

Grathwohl, W., Swersky, K., Hashemi, M., Duvenaud, D., and Maddison, C. J. Oops I took A gradient: Scalable sampling for discrete distributions. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3831–3841. PMLR, 2021. URL http://proceedings.mlr.press/v139/grathwohl21a.html.

Hansen, N. and Ostermeier, A. Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation. In *Proceedings of IEEE international conference on evolutionary computation*, pp. 312–317. IEEE, 1996.

Hastings, W. K. Monte carlo sampling methods using markov chains and their applications. 1970.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Hopf, T. A., Ingraham, J. B., Poelwijk, F. J., Schärfe, C. P. I., Springer, M., Sander, C., and Marks, D. S. Mutation effects predicted from sequence co-variation. *Nat Biotechnol*, 35(2):128–135, 2017. ISSN 1087-0156, 1546-1696. doi: 10.1038/nbt.3769. URL http://www.nature.com/articles/nbt.3769.

Hsu, C., Nisonoff, H., Fannjiang, C., and Listgarten, J. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, pp. 1–9, 2022.

Kuchner, O. and Arnold, F. H. Directed evolution of enzyme catalysts. *Trends in Biotechnology*, 15 (12):523–530, dec 1997. doi: 10.1016/s0167-7799(97)01138-4.

Li, C., Zhang, R., Wang, J., Wilson, L. M., and Yan, Y. Protein engineering for improving and diversifying natural product biosynthesis. *Trends in biotechnology*, 38(7):729–744, 2020.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function, 2021. URL http://biorxiv.org/lookup/doi/10.1101/2021.07.09.450648.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

Nesterov, Y. Introductory lectures on convex programming volume i: Basic course. pp. 1–212, 1998.

Nguyen, A., Clune, J., Bengio, Y., Dosovitskiy, A., and Yosinski, J. Plug & play generative networks: Conditional iterative generation of images in latent space. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 3510–3520. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.374. URL https://doi.org/10.1109/CVPR.2017.374.

Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T., and Rives, A. Msa transformer. In *International Conference on Machine Learning*, pp. 8844–8856. PMLR, 2021.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

Sun, H., Dai, H., Xia, W., and Ramamurthy, A. Path auxiliary proposal for mcmc in discrete space. In *International Conference on Learning Representations*, 2022.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6199.

Weinstein, E. N., Amin, A. N., Frazer, J., and Marks, D. S. Non-identifiability and the blessings of misspecification in models of molecular fitness and phylogeny, 2022. URL http://biorxiv.org/lookup/doi/10.1101/2022.01.29.478324.

Wu, Z., Johnston, K. E., Arnold, F. H., and Yang, K. K. Protein sequence design with deep generative models. *Current opinion in chemical biology*, 65:18–27, 2021.

Yang, K. K., Wu, Z., and Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat Methods*, 16(8):687–694, 2019. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-019-0496-6. URL http://www.nature.com/articles/s41592-019-0496-6.

Zanella, G. Informed proposals for local mcmc in discrete spaces. *Journal of the American Statistical Association*, 115(530):852–865, 2020.

A Appendix

A.1 Proof for Corollary 1

The basic idea of the proof is to use the triangle inequality to obtain a bound on the approximation error of a *sum* of experts which assumes that each expert has sufficiently smooth gradients.

Definition A function $f : \mathbb{R}^N \to \mathbb{R}$ has *K-Lipschitz continuous gradient* when

$$\|\nabla_{x'} f(x') - \nabla_x f(x)\| \leq L\|x' - x\| \tag{6}$$

for all $x, x' \in \mathbb{R}^N$.

For convenience, we reproduce a pertinent result here from Nesterov (1998) (Lemma 1.2.3).

Lemma 1.2.3 Nesterov (1998) *If $f : \mathbb{R}^N \to \mathbb{R}$ has an L-Lipschitz gradient, then for any $x, x' \in \mathbb{R}^N$ we have*:

$$|f(x') - f(x) - \langle \nabla_x f(x), x' - x\rangle| \leq \frac{L}{2}\|x' - x\|^2. \tag{7}$$

Lemma 1 *For functions f and g with K_f-Lipschitz and K_g-Lipschitz gradients respectively, the sum-composition $h = f + g$ has $(K_f + K_g)$-Lipschitz gradient.*

Proof: By the triangle inequality:

$$\begin{aligned}
\|\nabla_{x'} h(x') - \nabla_x h(x)\| &= \|\nabla_{x'}(f(x') + g(x')) - \nabla_x(f(x) + g(x))\| \\
&= \|\nabla_{x'} f(x') + \nabla_{x'} g(x') - \nabla_x f(x) - \nabla_x g(x)\| \\
&\leq \|\nabla_{x'} f(x') - \nabla_x f(x)\| + \|\nabla_{x'} g(x') - \nabla_x g(x)\| \\
&\leq (K_f + K_g)\|x' - x\|.
\end{aligned} \tag{8}$$

Lemma 2 *Suppose f_i, $i = 1, \ldots, M$ are functions with K_i-Lipschitz gradient. Then for any $x, x' \in \mathbb{R}^N$ we have*

$$\left|\sum_{i=1}^{M}\Big(f_i(x') - f_i(x)\Big) - \Big\langle \sum_{i=1}^{M} \nabla_x f_i(x), x' - x\Big\rangle\right| \leq \frac{\sum_{i=1}^{M} K_i}{2}\|x' - x\|^2. \tag{9}$$

Proof: Let $g = \sum_{i=1}^{M} f_i$ where each f_i, $i = 1, \ldots, M$ has K_i-Lipschitz gradient. By Lemma 1 we can see that g has a $\sum_{i=1}^{M} K_i$-Lipschitz gradient. Then Lemma 2 follows immediately by applying Lemma 1.2.3 from Nesterov (1998) to g.

The proof of Corollary 1 proceeds by bounding the approximation error between two consecutive states x^{r-1} and $x' \in \mathcal{N}(x^{r-1})$ in a path of length $R \sim \text{Unif}(1, U)$. For simplicity we assume $\mathcal{N}(x)$ is the 1-Hamming ball, i.e., $\|x^r - x^{r-1}\|^2 = 1$.

For $g = \sum_{i=1}^{M} f_i$ which has $K = \sum_{i=1}^{M} K_i$-Lipschitz gradient, Lemma 2 gives us that

$$-\frac{K}{2} \leq g(x') - g(x^{r-1}) - \langle \nabla g(x^{r-1}), x' - x^{r-1}\rangle \leq \frac{K}{2}.$$

Then an upper bound for $g(x') - g(x^{r-1})$ is

$$\begin{aligned}
g(x') - g(x^{r-1}) &\leq \langle \nabla g(x^{r-1}), x' - x^{r-1}\rangle + \frac{K}{2} \\
&= \langle \nabla g(x^0), x' - x^{r-1}\rangle + \langle \nabla g(x^{r-1}) - \nabla g(x^0), x' - x^{r-1}\rangle + \frac{K}{2} \\
&\leq \langle \nabla g(x^0), x' - x^{r-1}\rangle + Kr + \frac{K}{2} \\
&= \langle \nabla g(x^0), x' - x^{r-1}\rangle + K(r - \frac{1}{2}).
\end{aligned}$$

Following similar steps, we also have

$$g(x') - g(x^{r-1}) \geq \langle \nabla g(x^0), x' - x^{r-1} \rangle + K(r + \frac{1}{2}).$$

The remainder of the proof for Corollary 1 exactly follows Equations 64-72 in the proof of Theorem 3 in Sun et al. (2022) with $g(x)$ which has K-Lipschitz gradient.

Table 1: **50^{th} (100^{th}) percentile scores**. Population size is 128. Across all three proteins, `PPDE (Potts)` and `PPDE (Potts+ESM2)` in particular discover variants with higher predicted fitness and average mutations than the random-mutation-based samplers and `MALA-approx`. `PPDE (Potts)` achieves the highest evolutionary density scores on PABP; we attribute slightly lower evolutionary density scores compared to the baselines (except `CMA-ES`) on the more challenging UBE4B and GFP proteins because PPDE discovers variants with higher average mutations ($\sim 2-4+$ mutations vs. ~ 1 mutation) and higher predicted fitness. `CMA-ES` finds variants with high numbers of mutations ($\sim 10-17$) but low evolutionary density scores (3.47 on PABP, -94.76 on UBE4B and -62.43 on GFP compared to 6.92, -4.79, and -5.98 for `PPDE (Potts)`). `CMA-ES` seems to have significant difficulty with the larger proteins UBE4B and GFP; e.g., on GFP it achieves a log fitness of -2.50 compared to -0.04 for PPDE. We conclude that `CMA-ES` can be recommended for use only when a single protein variant is desired and the length of the protein is relatively small (e.g., ≤ 95 residues).

	Log fitness↑ (Augmented EVmutation)			Evolutionary density↑ (MSA Transformer)			Exploration (mean$_{\pm \text{std}}$ # muts)		
Potts expert	PABP	UBE4B	GFP	PABP	UBE4B	GFP	PABP	UBE4B	GFP
PPDE	0.27(0.86)	0.39(1.18)	-0.04(0.24)	6.92(13.43)	-4.79(-2.14)	-5.98(-0.76)	3.5± 0.9	2.7± 0.6	2.0± 0.3
Random search	0.09(0.82)	-0.19(0.34)	-0.04(0.04)	4.26(6.87)	-1.09(2.46)	-0.11(-0.11)	1.3± 0.5	1.1± 0.3	1.0± 0.2
Sim. annealing	0.09(0.44)	-0.19(0.28)	-0.04(0.10)	3.55(8.63)	-0.94(2.70)	-5.89(-0.99)	1.3± 0.5	1.0± 0.2	1.0 ±0.1
MALA-*approx*	0.09(0.56)	-0.19(0.58)	-0.04(0.10)	2.25(5.14)	-0.89(1.54)	-6.74(-1.88)	1.3± 0.5	1.03± 0.2	1.03± 0.2
CMA-ES	1.37(1.37)	2.54(2.54)	-2.50(-0.15)	3.47(3.47)	-94.76(0.0)	-62.43(0.0)	17.0± 0	15.5± 6.2	10.2± 9.4
Unsupervised experts									
Potts (No supervised)	0.70(1.47)	0.12(0.99)	-0.18(0.21)	9.17(18.54)	-4.24(-2.68)	-1.88(-1.59)	4.7 ± 1.2	2.6 ± 0.5	1.2 ±0.4
None (Supervised only)	0.14(0.44)	1.66(5.26)	-0.23(0.14)	-2.56(0.48)	-6.83(-6.29)	-9.28(-2.13)	1.9 ± 0.8	1.3 ± 0.7	1.7± 0.8
ESM2	0.14(0.63)	1.66(5.56)	-5.55(0.16)	-2.38(5.56)	-6.58(-3.83)	-126.82(5.90)	2.9± 1.3	2.2± 2.2	14.9± 12.6
Potts+ESM2	0.44(1.48)	1.30(3.33)	-0.04(0.33)	9.12(19.34)	-13.6(1.98)	-7.17(8.11)	5.3±1.8	4.3±0.7	2.1±0.3

Table 2: **Population diversity** (% unique sequences out of 128). PPDE samplers achieve the best diversity scores across all proteins.

	Random search	Simulated annealing	MALA-*approx*	CMA-ES	PPDE (Potts only)	PPDE (Super. only)	PPDE (Potts)	PPDE (ESM2)	PPDE (Potts+ESM2)
PABP	32.8	28.9	28.9	0.8	**85.2**	60.2	65.6	63.1	**85.2**
UBE4B	7.0	4.7	6.2	3.1	12.5	18.8	18.8	**36.5**	31.3
GFP	9.4	3.9	9.4	92.2	8.6	59.4	22.7	**92.9**	21.9