Fast non-autoregressive inverse folding with discrete diffusion

John J. Yang
Massachusetts Institute of Technology
johnyang@mit.edu

Jason Yim
Massachusetts Institute of Technology
jyim@csail.mit.edu

Regina Barzilay
Massachusetts Institute of Technology
regina@csail.mit.edu

Tommi Jaakkola
Massachusetts Institute of Technology
tommi@csail.mit.edu

Abstract

Generating protein sequences that fold into a intended 3D structure is a fundamental step in *de novo* protein design. De facto methods utilize autoregressive generation, but this eschews higher order interactions that could be exploited to improve inference speed. We describe a non-autoregressive alternative that performs inference using a constant number of calls resulting in a *23 times speed up* without a loss in performance on the CATH benchmark. Conditioned on the 3D structure, we fine-tune ProteinMPNN to perform discrete diffusion with a purity prior over the index sampling order. Our approach gives the flexibility in trading off inference speed and accuracy by modulating the diffusion speed. Code: https://github.com/johnyang101/pmpnndiff

1 Introduction

De novo protein design aims to design proteins from first principles without modifying an existing protein [7]. This involves designing protein 3D structures for a desired function such as binding then determining the sequence that would fold *in-vivo* into the designed structure. The first step is the protein structure generation problem for which RFdiffusion has become state-of-the-art [15]. The aim of this work is to develop a discrete diffusion model towards the second step of sampling sequences conditioned to fold into a desired 3D structure, referred to as *inverse (protein) folding* [6].

Generative models have already become the preferred tool for inverse folding: ProteinMPNN [3] is a widely used method with successful experimental validation. However, ProteinMPNN has two limitations. The first is its autoregressive decoding which scales linearly and can be prohibitively slow for large proteins. The second is the uniformly random decoding order which is likely suboptimal given substantial evidence of higher-order interactions occurring in protein evolution [13].

Inspired by RFdiffusion, in which a pre-trained protein folding model is fine-tuned with diffusion, we use a pre-trained ProteinMPNN and fine-tune it with diffusion. We explore multiple variants of discrete diffusion and find the best configuration to result in equivalent performance as ProteinMPNN on foldability while using 23 times less compute. Our best diffusion model is depicted in Figure 1. Our contribution enables exciting possibilities to extend ProteinMPNN using conditional diffusion models for improvement in both speed and controllable generation.

The paper is structured as follows. Related work is provided in Appendix A. Section 2 describes our method of fine-tuning ProteinMPNN with discrete diffusion. Section 3 evaluates each diffusion

Machine Learning for Structural Biology Workshop, NeurIPS 2023.

variant on the CATH benchmark [12]. We focus on analyzing tradeoffs in diversity, speed, and designability of sequence designs. Section 4 concludes with limitations and future directions.

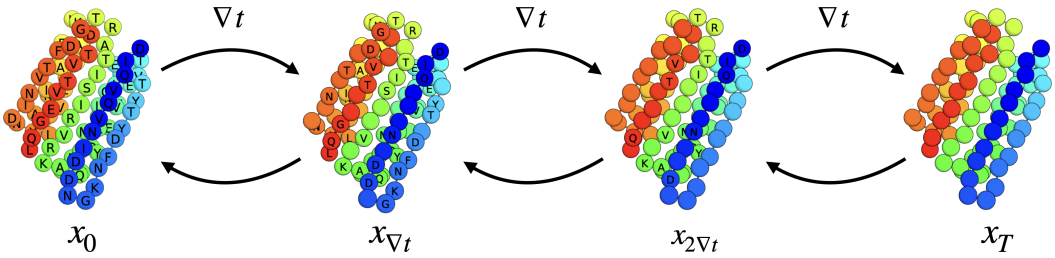

Figure 1: Overview. Starting from a sequence and structure, the forward process masks residues on each step. The reverse process unmasks on each step given only the structure and all residues initializd to `[MASK]`. Our framework allows for flexible decoding strategies by *striding* over intervals of time ∇t in each reverse step, i.e. the diagram uses 3 steps to decode the whole sequence.

2 Method

We first provide background on the problem formulation and ProteinMPNN in section 2.1. Next, section 2.2 describes discrete diffusion and its variants we implement for fine-tuning ProteinMPNN.

2.1 Background

Problem formulation. The task of inverse folding is to design a sequence $\boldsymbol{s} \in \mathcal{V}^L$ that folds into a given backbone structure $\boldsymbol{x} \in \mathbb{R}^{L\times 4\times 3}$ where $\mathcal{V}$ is the vocabulary of 20 amino acids plus a mask token `[MASK]` and L is the number of residues. Superscripts $\boldsymbol{s}^{(t)}$ will be used to refer to time $t \in [1, \dots, T]$ while subscripts s_i refer to residues i (i.e. index). Here, $\boldsymbol{x}$ refers to the atomic coordinates of the 5 canonical backbone atoms $\{N, C_\alpha, C, C_\beta, O\}$. It has been well-documented that sequences with as low as 30% sequence similarity can fold into the same structure [10]. Therefore, the goal is to learn a distribution of possible sequences that fold into a structure, $p(\boldsymbol{s}|\boldsymbol{x})$.

ProteinMPNN. First described in Dauparas et al. [3], we provide a brief summary. Each residue is represented as a node in an attributed graph $\mathcal{G}(\boldsymbol{x}) = (\mathcal{V}(\boldsymbol{x}), \mathcal{E}(\boldsymbol{x}))$. $\mathcal{V}(\boldsymbol{x})$ encodes geometric features such as orientation and sequence index of each residue; $\mathcal{E}(\boldsymbol{x})$ constructs a k-nearest neighbor graph based on C_α distance and encodes relative geometric features. The neural network uses message passing to learn embeddings of protein geometry through a masked language modeling objective that was first described in Ingraham et al. [8]. Inference is performed by sampling a uniformly random decoding order then autoregressively decoding residues one by one.

2.2 Discrete diffusion

We provide a overview of discrete diffusion models (D3PM) described in Austin et al. [1]. Discrete diffusion models are a class of latent variable generative models that are defined by a fixed forward Markov process $q(\boldsymbol{s}^{(1:T)}|\boldsymbol{s}^{(0)}) = \prod_{t=1}^{T} q(\boldsymbol{s}^{(t)}|\boldsymbol{s}^{(t-1)})q(\boldsymbol{s}^{(0)})$ from a starting sequence $\boldsymbol{s}^{(0)}$ and sequence of increasingly noisy latent variables $\boldsymbol{s}^{(1:T)} = (\boldsymbol{s}^{(1)}, \dots, \boldsymbol{s}^{(T)})$. The goal is to learn a model that parameterizes the reverse process $p_\theta(\boldsymbol{s}^{(0:T)}, \boldsymbol{x}) = \prod_{t=1}^{T} p_\theta(\boldsymbol{s}^{(t-1)}|\boldsymbol{s}^{(t)})$ where,

$$p_\theta(\boldsymbol{s}^{(t-1)}|\boldsymbol{s}^{(t)}) = \sum_{\boldsymbol{s}^{(0)}} q(\boldsymbol{s}^{(t-1)}|\boldsymbol{s}^{(t)}, \boldsymbol{s}^{(0)})p_\theta(\boldsymbol{s}^{(0)}|\boldsymbol{s}^{(t)}). \tag{1}$$

Thus, a model with weights θ learns how to *denoise* by predicting $p_\theta(\boldsymbol{s}^{(0)}|\boldsymbol{s}^{(t)})$. During training, we optimize the evidence-weighted lower bound (ELBO) between $q(\boldsymbol{s}^{(t-1)}|\boldsymbol{s}^{(t)}, \boldsymbol{s}^{(0)})$ and $p(\boldsymbol{s}_{t-1}|\boldsymbol{s}_t)$. For absorbing state diffusion, the ELBO reduces to the cross-entropy loss over masked residues,

$$\mathcal{L} = \mathbb{E}_{q(\boldsymbol{s}^{(0)})}\left[\sum_{t=1}^{T} \frac{T-t+1}{T} \mathbb{E}_{q(\boldsymbol{s}^{(t)}|\boldsymbol{s}^{(0)})}\left[\sum_{i=1}^{L} \mathbb{1}\{s_i^{(t)} = \texttt{[MASK]}\} \log p_\theta\left(\hat{s}_i^{(0)}|\boldsymbol{s}^{(t)}\right)\right]\right]. \tag{2}$$

Following Bond-Taylor et al. [2], each inner expectation is re-weighted to take into account the difficulty of denoising steps. Discrete diffusion depends on the noising process using a categorical distribution over each residue $q(s_i^{(t)}|s_i^{(t-1)}) = \text{Cat}(s_i^{(t)}; \boldsymbol{p} = s_i^{(t-1)}\boldsymbol{Q}_t)$ that is parameterized by the probabilities $\boldsymbol{p}$ of each category where $\boldsymbol{Q}_t$ is a doubly stochastic transition matrix with $s_i^{(t-1)}$ represented as a one-hot encoding. Each residue is noised independently. We can derive a closed form for sampling the t-step marginal and reverse step,

$$q(s_i^{(t)}|s_i^{(0)}) = \text{Cat}\left(s_i^{(t)}; \boldsymbol{p} = s_i^{(0)}\overline{\boldsymbol{Q}}_t\right), \quad \text{where} \quad \overline{\boldsymbol{Q}}_j = \prod_{j=1}^{t} \boldsymbol{Q}_j \tag{3}$$

$$q(s_i^{(t-1)}|s_i^{(t)}, s_i^{(0)}) = \text{Cat}\left(s_i^{(t-1)}; \boldsymbol{p} = \frac{s_i^{(t)}\boldsymbol{Q}_t^\top \odot s_i^{(0)}\overline{\boldsymbol{Q}}_{t-1}}{s_i^{(0)}\overline{\boldsymbol{Q}}_t(s_i^{(t)})^\top}\right). \tag{4}$$

We choose to use the **absorbing state** transition matrix,

$$\boldsymbol{Q}_t = (1-\beta_t)\mathbb{I} + \beta_t \mathbb{1} e_m^\top \tag{5}$$

where β_t controls the masking rate, the vocabulary is extended with a mask token, and e_m is a vector with 1 on the index of the mask token and 0 elsewhere. On each forward step, a percentage of the unmasked tokens transition to mask and stay masked until the final step, where all tokens are masked. The reverse process performs the opposite with unmasking starting with residue as `[MASK]`.

2.3 Diffusion improvements to ProteinMPNN

ProteinMPNN's autoregressive sampling can be seen as an absorbing state diffusion that unmasks one residue on each reverse step with a uniformly random decoding order. Given this perspective, we sought to extend ProteinMPNN to be more efficient with discrete diffusion. In the context of inverse folding, diffusion is conditioned on the structure, $p_\theta(\boldsymbol{s}_{t-1}|\boldsymbol{s}_t, \boldsymbol{x})$. Borrowing techniques from Bond-Taylor et al. [2], we first fine-tune ProteinMPNN with non-autoregressive diffusion training and strided sampling. Second, we utilize more informed sampling orders based on a purity prior [14].

Non-autoregressive diffusion. We fine-tune ProteinMPNN with absorbing state diffusion where a subset of the residues are masked/unmasked on each step. We choose a linear schedule for masking where, for a length L protein, approximately $\lfloor \frac{t}{T} L \rfloor$ will be masked at time t. The t-step marginal probability $\boldsymbol{p} = s_i^{(0)}\overline{\boldsymbol{Q}}_t$ in eq. (3) is then,

$$s_i^{(0)}\overline{\boldsymbol{Q}}_t = \prod_{j=1}^{t}(1-\beta_j)s_i^{(0)} + \prod_{j=1}^{t}\beta_j \mathbb{1} e_m^\top = \left(1 - \frac{t}{T}\right) s_i^{(0)} + \frac{t}{T}\mathbb{1} e_m^\top \tag{6}$$

where β_t is set such that $t/T = \prod_{j=1}^{t}\beta_j$. Training proceeds using eq. (3) to noise and $\mathcal{L}$ eq. (2) as the loss. To sample, all residues are initially masked then $\lfloor \frac{1}{T} L \rfloor$ residues are unmasked on each step according to a decoding order (to be discussed soon). Using T steps during sampling can still be prohibitively slow. We explored efficient sampling where on each step we decode $\lfloor \frac{\nabla t}{T} L \rfloor$ residues and only require $\lceil \frac{\nabla t}{T} \rceil$ steps where ∇t is an integer value representing the *stride interval*. A depiction of strided sampling is in Figure 1.

Purity Prior. Since structure is correlated with evolutionary couplings [5], we sought to bias the masking/unmasking order based on couplings strengths, but our preliminary attempts were not successful. Instead, we utilize a purity prior to bias the order. The *purity* of residue index $i \in \{1, \ldots, L\}$ at step t is defined as

$$\mathfrak{p}(i,t) = \max_{j \in \{1,\ldots,L\}} p_\theta((\boldsymbol{s}_0)_i = j|\boldsymbol{s}_t) \tag{7}$$

Purity can be thought of as the model's confidence in its prediction at a given index relative to the other indices. We hypothesized ProteinMPNN would be more confident about jointly predicting coupled residues. On each forward/reverse step, we perform importance sampling based on $\mathfrak{p}$ to determine the next location to mask/unmask (see Tang et al. [14]).

3 Experiments

To evaluate ProteinMPNN diffusion, we report results on the CATH 4.2 Single Chain benchmark previously reported in ProteinMPNN.

Metrics. Following ProteinMPNN, we sample 8 sequences for each structure in the CATH test set and calculate the following metrics. The most common metric is *sequence recovery*, defined as percentage of correct amino acids relative to the native sequence, as the primary metrics [6, 4, 19]. *Designability* is defined as the RMSD error of a pre-trained protein folding model (we use ESMFold [11]) to predict the intended structure from the sampled sequence. Multiple works have found designability to correlate with experimental success [3, 15, 16], such evidence does not exist for sequence recovery. Therefore, we designate designability as the main metric to optimize.

In addition, we report *diversity* as the average pairwise Levenshtein distance between sequences for a structure. Lastly, *speed* is the average wall clock time (in seconds) to sample one sequence per protein in the test set using a single NVIDIA V100.

Diffusion fine-tuning. We set $T = 100$ and use $\nabla t = 20$ when using strided sampling for a total of 5 model calls to sample any sequence. Depending on the protein length, each model call will have variable wall clock time. Dropout is set to 0.1 during training. The Adam optimizer is initialized with a learning rate of 10^{-4}. We first pre-train ProteinMPNN with the standard next-token prediction objective as done in Dauparas et al. [3] on the CATH single chain training set for 200 epochs with batch size of 10,000 tokens. ProteinMPNN is then fine tuned using the diffusion objective (Equation (2)) for 1000 epochs with a batch size of 5k tokens. Models were trained on NVIDIA RTX A6000-48GB.

3.1 Results

Our results are presented in Table 1. Using the same model, we evaluate three different sampling variants based on whether purity order and strided sampling ($\nabla t > 1$) are used. Our baseline does not include purity or strided sampling. Using purity to determine the decoding order results in improvement over recovery and designability but a drop in diversity. Including strided sampling gives a 10 times improvement in speed. In comparison to ProteinMPNN, we find sequence recovery is worse with diffusion but designability remains on par with a slight drop in diversity. The advantage with diffusion is clear: a 23 times speed up with a small drop in diversity and designability.

Table 1: Performance of discrete diffusion variants.

	Purity	∇t	Recovery (↑)	Diversity (↑)	Designability (↓)	Speed (↓)
Diffusion	✗	1	33.6 %	0.624	2.855	355.7
	✓	1	39.6 %	0.424	2.158	329.8
	✓	20	39.9 %	0.420	2.112	33.7
ProteinMPNN			47.9 %	0.386	2.007	768.3

4 Discussion

The development of efficient and accurate methods for inverse protein folding remains a cornerstone task in *de novo* protein design. We introduced a diffusion approach that fine-tunes a state-of-the-art inverse folding model to significantly accelerates the inference process. Our work has several limitations. First, training ProteinMPNN diffusion without pre-training performs worse than fine-tuning. The reasons for this are unclear. Second, our non-autoregressive generation approach does not edit previously generated residues, making it prone to error accumulation. Ideally, the model would be able to iteratively refine the whole sequence like in other non-autoregressive sequence generation works. Third, we utilize a discrete time diffusion process that is incompatible with continuous time stochastic differential equations (SDE). An interesting direction would to explore the continuous time formulations. Lastly, we were not able to find correlations between purity decoding order with

structural characteristics, i.e. residues close in space should decode together. We plan to investigate further into optimal decoding orders and analyze its relation with structure.

References

[1] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[2] Sam Bond-Taylor, Peter Hessey, Hiroshi Sasaki, Toby P. Breckon, and Chris G. Willcocks. Unleashing transformers: Parallel token prediction with discrete absorbing diffusion for fast high-resolution image generation from vector-quantized codes, 2021.

[3] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615): 49–56, 2022. doi: 10.1126/science.add2187. URL `https://www.science.org/doi/abs/10.1126/science.add2187`.

[4] Zhangyang Gao, Cheng Tan, and Stan Z Li. Pifold: Toward effective and efficient protein inverse folding. *arXiv preprint arXiv:2209.12643*, 2022.

[5] Thomas A Hopf, Charlotta PI Schärfe, João PGLM Rodrigues, Anna G Green, Oliver Kohlbacher, Chris Sander, Alexandre MJJ Bonvin, and Debora S Marks. Sequence co-evolution gives 3d contacts and structures of protein complexes. *elife*, 3:e03430, 2014.

[6] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International Conference on Machine Learning*, pages 8946–8970. PMLR, 2022.

[7] Po-Ssu Huang, Scott E Boyken, and David Baker. The coming of age of de novo protein design. *Nature*, 537(7620):320–327, 2016.

[8] John Ingraham, Vikas Garg, Regina Barzilay, and Tommi Jaakkola. Generative models for graph-based protein design. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/f3a4ff4839c56a5f460c88cce3666a2b-Paper.pdf`.

[9] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael J. L. Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons, 2021.

[10] Evgeny Krissinel. On the relationship between sequence and structure similarities in proteomics. *Bioinformatics*, 23(6):717–723, 2007.

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

[12] Christine A Orengo, Alex D Michie, Susan Jones, David T Jones, Mark B Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5 (8):1093–1109, 1997.

[13] Tyler N Starr and Joseph W Thornton. Epistasis in protein evolution. *Protein science*, 25(7): 1204–1218, 2016.

[14] Zhicong Tang, Shuyang Gu, Jianmin Bao, Dong Chen, and Fang Wen. Improved vector quantized diffusion models, 2023.

[15] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, pages 1–3, 2023.

[16] BIM Wicky, LF Milles, A Courbet, RJ Ragotte, J Dauparas, E Kinfu, S Tipps, RD Kibler, M Baek, F DiMaio, et al. Hallucinating symmetric protein assemblies. *Science*, 378(6615): 56–61, 2022.

[17] Kevin E. Wu, Kevin K. Yang, Rianne van den Berg, James Y. Zou, Alex X. Lu, and Ava P. Amini. Protein structure generation via folding diffusion, 2022.

[18] Kevin K. Yang, Hugh Yeh, and Niccolò Zanichelli. Masked inverse folding with sequence transfer for protein representation learning. *bioRxiv*, 2023. doi: 10.1101/2022.05.25.493516. URL `https://www.biorxiv.org/content/early/2023/03/19/2022.05.25.493516`.

[19] Kai Yi, Bingxin Zhou, Yiqing Shen, Pietro Liò, and Yu Guang Wang. Graph denoising diffusion for inverse protein folding. *arXiv preprint arXiv:2306.16819*, 2023.

A Related work

Advances in inverse folding can be broadly categorized into improving structure representation or sequence generation. On the structure side, graph-based architectures combined with geometric features led to significant improvements over MLP and CNN based models [8, 9, 4]. On the sequence side, Transformer-based architectures have dominated [8, 6, 18]. However, these Transformer-based architectures generate sequences autoregressively, leading to slow inference times and high variance in generation quality. Non-autoregressive alternatives have shown strong performance on benchmarks such as perplexity and sequence recovery [4, 19]. However, these benchmarks have proved unreliable to protein designers. Independent of us, Wu et al. [17] found that ESM-IF [6] achieving SOTA sequence recovery of 72% but poor designability results compared to methods with lower sequence recovery. Multiple works have found designability has been shown to correlate with experimental validation [3, 15, 16]. Our method improves on Protein-MPNN, a popular model among protein designers that has demonstrated strong experimentally validated results, by generating sequences faster and non-autoregressively [3].