Diffusion Language Models Are Versatile Protein Learners

Xinyou Wang [*] [◇] [♡] **Zaixiang Zheng** [*] [♡] **Fei Ye** [♡] **Dongyu Xue** [♡] **Shujian Huang** [◇] **Quanquan Gu** [♡]

Abstract

This paper introduces **d**iffusion **p**rotein **l**anguage **m**odel (DPLM), a versatile protein language model that demonstrates strong generative and predictive capabilities for protein sequences. We first pre-train scalable DPLMs from evolutionary-scale protein sequences within a generative self-supervised discrete diffusion probabilistic framework, which generalizes language modeling for proteins in a principled way. After pre-training, DPLM exhibits the ability to generate structurally plausible, novel and diverse protein sequences for unconditional generation. We further demonstrate the proposed diffusion generative pre-training make DPLM possess a better understanding of proteins, making it a superior representation learner, which can be fine-tuned for various predictive tasks, comparing favorably to ESM2 (Lin et al., 2022). Moreover, DPLM can be tailored for various needs, which showcases its prowess of conditional generation in several ways: (1) conditioning on partial peptide sequences, *e.g.*, generating scaffolds for functional motifs with high success rate; (2) incorporating other modalities as conditioners, *e.g.*, structure-conditioned generation for inverse folding; and (3) steering sequence generation towards desired properties, *e.g.*, satisfying specified secondary structures, through a plug-and-play classifier guidance.

1 Introduction

Proteins, which are 3D-folded linear sequences of amino acids, play a pivotal role in regulating various biological functions, including transcription, translation, signaling, and the control of the cell cycle. Recently, the promise of learning to understand and design proteins via data-driven generative deep learning has initiated a significant paradigm shift apart from the long-established physics-based methods.

[*]Equal contribution [◇]Dept. of Computer Science, Nanjing University (this work was done during Xinyou's internship at ByteDance Research) [♡]ByteDance Research. Correspondence to: Quanquan Gu <quanquan.gu@bytedance.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

The analogies between protein sequences and human languages have long been recognized (Yang et al., 2019; Ferruz & Höcker, 2022). Drawing inspiration from the remarkable progress in NLP achieved by language models (LMs; Devlin et al., 2019; Radford et al., 2018; OpenAI, 2023) thanks to the *scalability* of Transformers (Vaswani et al., 2017) and the existence of large-scale text data, recent explorations in protein has also demonstrated the impressive capabilities of protein language models (Rives et al., 2019; Lin et al., 2022; Hu et al., 2022), learned from the universe of evolutionary-scale protein sequences. As a result, protein LMs have become one of the most important cornerstones in AI for protein research, serving a pivotal role not only in predictive tasks (*e.g.*, probing functional properties, and predicting protein structures from single sequences without explicit evolutionary homologs) but also in generative tasks (*e.g.*, redesigning sequences given protein backbone structures, or synthesizing completely new protein sequences).

While current protein LMs have made significant strides, they have not yet reached their fullest potential. One of the fundamental problems is rooted in the widely-used pretraining objectives, *i.e.*, masked prediction *vs*. autoregression:

(i) For masked prediction, masked language models (Masked-LMs, *e.g.*, ESM family; Rives et al., 2019; Lin et al., 2022) excel in sequence understanding for protein predictive tasks, thanks to their *bi-directional receptive field*. However, Masked-LMs are unable to perform protein sequence generation, due to the lack of a well-defined formulation for generative modeling. We further postulate that this could even cap their predictive power, since a powerful generative model that can *create* new samples by learning the underlying data distribution, is expected to simultaneously acquire a deep *understanding* of the data. As a famous quote, "*what you cannot create, you do not understand*."

(ii) For autoregression, autoregressive language models (AR-LMs, *e.g.*, ProGen; Nijkamp et al., 2022), albeit good at generation, often fall short in understanding sequence data (Radford et al., 2018) including proteins (Elnaggar et al., 2021). More importantly, proteins are structural macromolecules rather than simple linear strings. Consequently, while effective as an inductive bias for text, AR-LMs are constrained by their uni-directional receptive field, only accessing one-sided sequence context. This limitation stems from

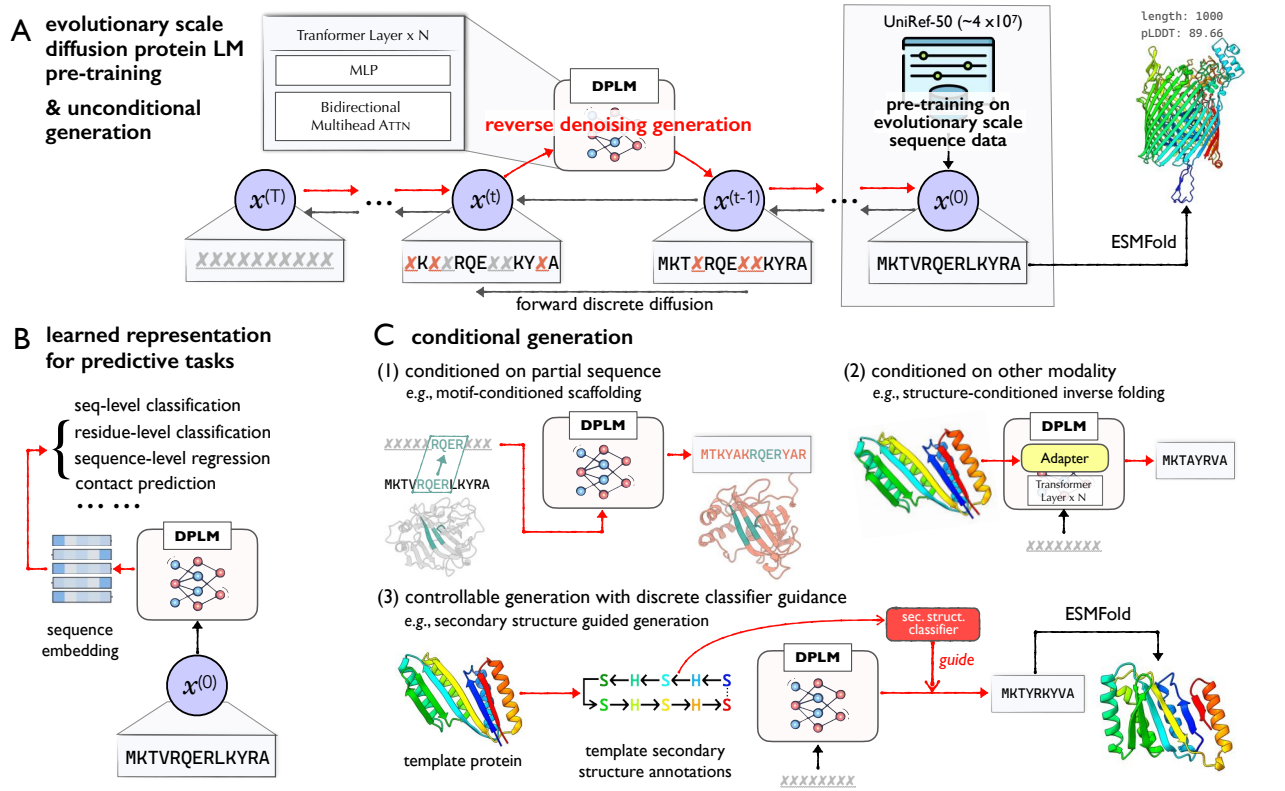

Figure 1. Overall illustration of DPLM. **(A)**: modeling, pre-training and unconditional generation; **(B)**: protein sequence representation for predictive tasks; **(C)**: conditional generation, including **(1)** sequence conditioning (*e.g.*, motif-scaffolding), **(2)** cross-modal conditioning (*e.g.*, inverse folding), and **(3)** plug-and-play controllable generation with discrete classifier guidance (*e.g.*, secondary structure).

capturing the complex global interactions of amino acids, thereby hindering both generative and predictive capabilities of protein LMs.

This highlights the demand for a general-purpose and versatile protein LM that combines predictive and generative capabilities. Provided the aforementioned analysis, we reason that, *the key ingredients* for such a versatile protein LM lie in (1) *strong & scalable generative modeling framework* to best digest the universe of massive protein sequences; and (2) *bi-directional receptive field* for better modeling residue-wise global interactions.

On the other hand, diffusion models (Ho et al., 2020; Song et al., 2020) have shown great success in generating *continuous* data, especially in rendering photorealistic images (Rombach et al., 2021, *inter alia*). They have further manifested incredible achievement in modeling protein structures (Yim et al., 2023; Watson et al., 2023; Ingraham et al., 2023). This can be attributed to their favorable properties of non-autoregressive denoising generation with iterative refinement and global receptive field. Besides, denoising autoencoding has a long history for representation learning (Vincent et al., 2010, *inter alia*), while recent studies have verified that diffusion-based generative models can be effective self-supervised learners (Chen et al., 2024a). These make diffusion models an appealing generative foundation for protein language modeling. However, directly applying conventional Gaussian diffusion to protein sequences necessitates additional continuous relaxations (Lisanza et al., 2023), which does not fit the *discrete* nature of protein sequence and has not yet proven successful in practice.

In this paper, we present **d**iffusion **p**rotein **l**anguage **m**odel (DPLM), a novel approach aimed at achieving a unified and versatile protein LM through diffusion generative pre-training on evolutionary-scale protein sequences. DPLM is grounded in a discrete diffusion probabilistic framework, serving as a principled generative generalization of language modeling. During pre-training, DPLM is tasked with denoising the input protein sequence at different noise levels, ranging from completely noisy to clean ones, enforcing DPLM to best the model complex intrinsic dependencies of amino acid sequences. After pre-training, DPLM can be used for protein sequence generation and providing effective representations for downstream predictive tasks. We highlight our contributions as follows:

- We propose DPLM, a versatile protein LM under discrete diffusion framework, with model size up to 3B, pre-trained on evolutionary-scale protein sequences. We further develop multiple conditioning strategies covering various use needs, especially discrete classifier guidance for controllable generation. As a result,

DPLM combines the best of both worlds, *i.e.*, the scalable expressiveness of language models and the strong generative power of diffusion models, serving as a versatile biological foundation model (Fig. 1, §3).

- We show that DPLM is capable of generating highly structurally plausible (*i.e.*, averaged pLDDT > 80), novel and diverse for unconditional protein sequence generation, suggesting that DPLM well captures the universe of protein sequence data (Fig. 1A, §4.1).
- We demonstrate that DPLM *understands* protein better, serving as a superior representation learner, which can be fine-tuned for various downstream tasks, comparing favorably with widely-used protein sequence encoder models, *e.g.*, ESM-2 (Lin et al., 2022) (Fig. 1B, §4.2).
- DPLM can be further exploited for conditional generation for a variety of needs: DPLM can (1) condition on pre-specified partial sequence, *e.g.*, scaffolding for functional motifs with high success rate; (2) incorporate other modalities as conditions, *e.g.*, structure-conditioned generation for inverse folding; (3) generate protein sequences towards desired properties with plug-and-play classifier-guidance, *e.g.*, steering DPLM to synthesize proteins that satisfy arbitrary user-defined secondary structure annotations (Fig. 1C, §4.3).

2 Preliminaries

2.1 Language Modeling for Protein

Language modeling aims to estimate the underlying distribution $\mathbf{x} \sim q(\mathbf{x})$ of the sequence data of our interest, *e.g.*, text or protein sequence, by learning a probabilistic model $p_\theta(\mathbf{x})$. Here the *language model* (LM) θ is parameterized by a neural network, in particular Transformers (Vaswani et al., 2017), which have become the *de facto* choice dominating different domains with scalable and performing expressiveness. In this work, we are interested in language modeling for protein sequences, for which $\mathbf{x} = (x_1, x_2, \ldots, x_L) \in \{0,1\}^{L\times|\mathcal{V}|}$ is a sequence composing L elements, $\mathcal{V}$ is the vocabulary within a discrete data support of 20 amino acids $\mathcal{V} = \{1, ..., 20\}$. One thing we most care about is the generative and representational capabilities of protein LMs. Here we review the typical probabilistic paradigms for language modeling, *i.e.*, *masked prediction* and *autoregression*, and their pros and cons as the foundation for protein LMs, as follows.

Masked Prediction. Masked language models (Masked-LMs or MLMs), *e.g.*, BERT (Devlin et al., 2019) and its variants for protein sequence (ESM family, Rives et al., 2019; Lin et al., 2022), employ a bidirectional transformer to take into account both the left and right context to predict the masked (amino acid) symbols in a *mask-predict* autoencoding manner,

$$\mathbb{E}_{q(\mathbf{x})} \log p_\theta(\mathbf{x}) = \mathbb{E}_{q(\mathbf{x})} \textstyle\sum_{1\le i\le L} b_i \cdot \log p_\theta(x_i|\bar{\mathbf{x}}_{\text{m}}), \quad (1)$$

where $b_i = \mathbf{1}_{\bar{\mathbf{x}}_i = \texttt{[X]}}$ derived from a fixed chance (*e.g.*, widely-adopted 15%) of masking $\mathbf{x}$ with a special mask symbol [X], resulting in the masked observation $\bar{\mathbf{x}}_{\text{m}}$. A per-token conditional independence assumption is made as well. Masked-LMs significantly excel the performance of a wide range of sequence understanding tasks for both natural language and protein. However, its bidirectionality nature makes it difficult to apply to sequence generation.

Autoregression. AR-LMs are prevailing in the realm sequence generation (OpenAI, 2023; Nijkamp et al., 2022), which adopts a sequential factorization over the sequence using the probability chain rule. In this case, the log-likelihood of such models is maximized over the dataset given by:

$$\mathbb{E}_{q(\mathbf{x})} \log p_\theta(\mathbf{x}) = \mathbb{E}_{q(\mathbf{x})} \textstyle\sum_{1\le i\le L} \log p_\theta(x_i|\mathbf{x}_{<i}), \quad (2)$$

where causal masking is used to ensure sequential dependency structure. To sample from AR-LMs, it requires ancestral sampling for L iterative steps from $x_1 \sim p_\theta(x_1), x_2 \sim p_\theta(x_2|x_1)$ towards $x_L \sim p(x_L|x_1, ..., x_{L-1})$ in a strict left-to-right unidirectional manner.

2.2 Diffusion Probabilistic Models

Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020) are a class of generative models characterized by a pair of Markov processes, *i.e.*, a forward diffusion process and a backward denoising process. The *forward* process $q(\mathbf{x}^{(1:T)}|\mathbf{x}^{(0)}) = \prod_{t=1}^{T} q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})$ gradually perturb the data $\mathbf{x}^{(0)} \sim q(\mathbf{x}^{(0)})$ into a stationary distribution $\mathbf{x}^{(T)} \sim q_{\text{noise}}$ with T increasingly noisy steps $\mathbf{x}^{(1:T)} = \mathbf{x}_1, \ldots, \mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}, \ldots, \mathbf{x}^{(T)}$. The learned *backward* process $p_\theta(\mathbf{x}^{(0:T)}) = p(\mathbf{x}^{(t)}) \prod_{t=1}^{T} p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})$, reversely, gradually denoises the samples towards the data distribution. To fit the model $p_\theta(\mathbf{x}^{(0)})$ to the data distribution $q(\mathbf{x}^{(0)})$, the denoiser model is typically optimized by the variational bound of the log-likelihood (Ho et al., 2020):

$$\begin{aligned}
&\mathbb{E}_{q(\mathbf{x}^{(0)})}\big[\log p_\theta(\mathbf{x}^{(0)})\big] \ge \mathbb{E}_{q(\mathbf{x}^{(0:T)})}\left[\log \frac{p_\theta(\mathbf{x}^{(0:T)})}{q(\mathbf{x}^{(1:T)}|\mathbf{x}^{(0)})}\right] \\
&= \mathbb{E}_{q(\mathbf{x}^{(0)})}\Big[\log p_\theta(\mathbf{x}^{(0)}|\mathbf{x}^{(1)}) + \text{const.} \\
&\quad + \textstyle\sum_{t=2}^{T} \underbrace{-\text{KL}\big[q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)}) \| p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})\big]}_{\mathcal{J}_t}\Big].
\end{aligned}$$

Afterwards, it generates by first sampling from $q_{\text{noise}}(\mathbf{x}^{(T)})$, followed by iterative denoising with $p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})$.

3 DPLM: A *Versatile* Protein LM

Motivation. Continuous diffusion with Gaussian perturbation kernel has demonstrated impressive performance in generating continuous data in Euclidean space (Rombach et al., 2021; Ho et al., 2022), and the more general Riemannian manifolds (De Bortoli et al., 2022). Recently, continuous diffusion has shown to rival in modeling protein structures (Watson et al., 2023; Ingraham et al., 2023, *inter alia*), wherein its *bidirectional receptive field* is ideally suited for modeling residue-wise global interactions.

This motivates us to blend diffusion models, which are well-suited for protein as discussed above, and language models, which are well known as *scalable and expressive sequence learners*. This leads to our pursuit of a diffusion protein LM, taking the best of both worlds.

A direct use of continuous diffusion, however, is not necessarily the best choice for modeling discrete sequence data (Li et al., 2022; Dieleman et al., 2022; Lisanza et al., 2023), due to the *pitfall of discreteness* that makes Gaussian diffusion hardly model the discrete nature of sequence data in embedding space (Ye et al., 2023b). To this end, discrete diffusion (Hoogeboom et al., 2021b; Austin et al., 2021) that directly operates over the discrete state space, becomes a more well-suited probabilistic model for protein sequences.

3.1 Protein Language Modeling *w/* Discrete Diffusion

Modeling. Let $\texttt{Cat}(\mathbf{x};\mathbf{p})$ be a categorical distribution on protein sequence $\mathbf{x}$ parameterized by a vector $\mathbf{p}$ on $(|\mathcal{V}|-1)$-dimensional probability simplex. The forward process of discrete diffusion defines a Markov process governed by the transition kernel:

$$q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = \texttt{Cat}\big(\mathbf{x}^{(t)}; \beta_t\mathbf{x}^{(t-1)} + (1-\beta_t)\mathbf{q}_{\text{noise}}\big),$$

where $\mathbf{q}_{\text{noise}}$ is the probability vector of stationary distribution $q_{\text{noise}}(\mathbf{x}^{(t)})$, *i.e.*, $q(\mathbf{x}^{(t)}) = \texttt{Cat}(\mathbf{x}^{(t)};\mathbf{p}=\mathbf{q}_{\text{noise}})$, and $0 \ll \beta_t < 1$ is the noise schedule controlling the degree of corruption at timestep t. In this case, the distribution of corrupted sample $\mathbf{x}^{(t)}$ given its original data $\mathbf{x}^{(0)}$ has a closed-form expression:

$$q(\mathbf{x}^{(t)}|\mathbf{x}^{(0)}) = \texttt{Cat}\big(\mathbf{x}^{(t)}; \alpha_t\mathbf{x}^{(0)} + (1-\alpha_t)\mathbf{q}_{\text{noise}}\big), \quad (3)$$

where $\alpha_t = \prod_{i=1}^{t}\beta_i$ such that $\lim_{t\to T}\alpha_t \to 0$, which preserves no information from the data and converges to the stationary distribution $\mathbf{q}_{\text{noise}}$ at timestep T. This shows that the diffusion process is intuitively a convex combination between data and the stationary noise prior distribution. Different stationary distributions $\mathbf{q}_{\text{noise}}$ lead to different formulations of discrete diffusion models. Here we primarily consider the *absorbing* diffusion with $q(\mathbf{x}^{(t)}) = \{1 \text{ if } \mathbf{x}^{(t)} = \texttt{[X]};\ 0 \text{ if } \mathbf{x}^{(t)} \neq \texttt{[X]}\}$, where $\texttt{[X]}$ is an absorbing state, akin to Masked-LMs. The formulation of Eq. (3) results in $\mathbf{x}^{(t)}$ either being masked or the same as $\mathbf{x}^{(0)}$, with a masking ratio $(1-\alpha_t)$.

Learning. As stated in Austin et al. (2021), discrete diffusion inherently connects to AR-LM and Masked-LM, whilist Zheng et al. (2023a) further simplifies the learning objective of discrete diffusion, with their proposed reparameterized backward transition, from KL divergences between two categoricals into reweighted cross-entropies:

$$\begin{aligned}\mathcal{J}_t &= \mathbb{E}_{q(\mathbf{x}^{(0)})} - \text{KL}\big[q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)},\mathbf{x}^{(0)})\|p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})\big] \\ &= \mathbb{E}_{q(\mathbf{x}^{(0)})}\Big[\lambda^{(t)}\textstyle\sum_{1\le i\le L} b_i(t)\cdot\log p_\theta(\mathbf{x}_i^{(0)}|\mathbf{x}^{(t)})\Big], \quad (4)\end{aligned}$$

where $\lambda^{(t)}$ is a weighting coefficient induced from the specific noising schedule (see Appendix A for proof). Eq. (4) reveals that Masked-LMs (*i.e.*, $\mathbf{x}^{(t)} \triangleq \bar{\mathbf{x}}_{\text{m}}$ in Eq. (1)) and AR-LMs (*i.e.*, $\mathbf{x}^{(t)} \triangleq \mathbf{x}_{<t}$ and $b_i \triangleq 1$ in Eq. (2)) can be considered as special cases in this generalized form of discrete diffusion LMs, contingent on their respective specifications of the noise-induced configurations. As a result, the process of learning according to Eq. (4) inherently encapsulates both Masked-LMs and AR-LMs within the ambit of the proposed DPLM.

Evolutionary-scale Pre-training. The pre-training procedure for DPLM utilizes the UniRef50 database (Suzek et al., 2015), which comprises around 45 million protein sequences, totaling about 14 billion amino acid tokens. In the case of exceedingly lengthy protein sequences, we emulate ESM2 (Lin et al., 2022) by truncating these proteins to a random sequence of 1024 tokens. Besides, we adhere to the setting for model architecture and scales as ESM2, which correspond to DPLM with sizes of 150M, 650M and 3B. We train all models for 100K updates, with batch size of 320K for 150M model and 1M for 650M/3B models.

Generation. Given a trained DPLM, it can synthesize new amino acid sequences by the reverse iterative denoising process of discrete diffusion (Hoogeboom et al., 2021b; Austin et al., 2021). Formally, discrete diffusion samples from the following distribution,

$$p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) = \textstyle\sum_{\hat{\mathbf{x}}_0} q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)},\hat{\mathbf{x}}_0)p_\theta(\hat{\mathbf{x}}_0|\mathbf{x}^{(t)}).$$

In particular, at time t, we first generate $\hat{\mathbf{x}}_0$ from $p_\theta(\cdot|\mathbf{x}^{(t)})$, then a less noisy $\mathbf{x}^{(t-1)}$ is sampled by $q(\cdot|\mathbf{x}^{(t)},\mathbf{x}^{(0)}=\hat{\mathbf{x}}_0)$ given $\mathbf{x}^{(t)}$ and $\hat{\mathbf{x}}_0$. This process is repeated from T to 1. The generative denoising process of DPLM can be viewed as an iterative *mask-predict* approach. Specifically, the starting sequence is initialized as 100%-noisy state (*i.e.*, all $\texttt{[X]}$'s). At each iteration, a subset of masked tokens is updated based on the model's prediction $\hat{\mathbf{x}}_0$, while the remaining tokens are re-masked, according to ranked $\log p_\theta(\hat{\mathbf{x}}_0|\mathbf{x}^{(t)})$ (Ghazvininejad et al., 2019; Zheng et al., 2023a).

Representation. DPLM is tasked with denoising the input protein sequence at all noise levels, including the original noise-free data (*e.g.*, noise level at 0%). As a result, DPLM can simultaneously serve as a protein sequence representation learner over massive protein sequence data, providing useful sequence embedding for various protein predictive downstream tasks, *e.g.*, sequence/residue-level classification/regression. The sequence embedding can be attained by simply letting DPLM take as input the given amino acid sequence $\mathbf{x}$: $\mathbf{h}(\mathbf{x}) \leftarrow \text{DPLM}_\theta(\mathbf{x}, t=0) \in \mathbb{R}^{L\times d}$, where d is the dimension of embedding.

3.2 Conditioning

Being able to efficiently sample realistic proteins is necessary but not sufficient for downstream applications such as therapeutic development, since unconditional samples

are unlikely to possess desired functional properties. Here we elaborate on how to make DPLM practically useful by conditioning for various needs, which covers most common scenarios, *i.e.*, sequence conditioning, cross-modal conditioning, and plug-and-play preference-guided conditioning.

Case I: Conditioning on partial sequence (Fig. 1C-1). Protein generation containing pre-specified polypeptides corresponds to various use cases such as generating scaffolds for given functional motifs, infilling antibody CDR loops, or imposing expert knowledge a-priori. This implies our desire for DPLM to sample from this conditional distribution $\mathbf{x} \sim p_\theta(\mathbf{x}|\bar{\mathbf{x}}) = \prod_{i=1}^{L} b_i \cdot p_\theta(x_i|\bar{\mathbf{x}})$, which has already been learned through Eq. (4). The observed partial sequence $\bar{\mathbf{x}} = \{\bar{x}_i \in \mathcal{V} \text{ if } b_i = 0;\ \texttt{[X]} \text{ if } b_i = 1 | i \in [1, L]\}$. Namely, $b_i \in \{0, 1\}$ indicates whether the predicted sequence must preserve the observation for the i-th residue such that $x_i = \bar{x}_i$.

Case II: Adapting DPLM to conditioned on other modalities (Fig. 1C-2). Generating protein sequence subject to cross-modal constraints $\mathbf{c}$, *i.e.*, $\mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{c})$, has profound value in practice, such as inverse protein folding where sequences are generated for given backbone structure (Dauparas et al., 2022; Zheng et al., 2023b), or conditioning on small molecule ligands for binder design (Dauparas et al., 2023). Given that DPLM primarily operates over amino acid tokens, in these cases, we can equip DPLM with cross-modal conditioning by adapter-tuning with a pre-trained modality expert encoder $\mathcal{E}_\phi(\mathbf{c})$ and a newly-added cross-attention-based adapter following Zheng et al. (2023b). During training, we freeze the parameters of the modality encoder and DPLM, and only update the parameters of the adapter via supervised fine-tuning on the given paired data $(\mathbf{x}, \mathbf{c})$. We then obtain a conditional DPLM for $p_\theta(\mathbf{x}|\mathcal{E}_\phi(\mathbf{c}))$ making the full potentials of both DPLM and the modality expert $\mathcal{E}_\phi(\mathbf{c})$. In §D.5, we also develop classifier-free guidance for such adapter-tuned DPLM as an immediately available booster for cross-modal conditional generation without intricate condition dropout during training.

Case III: Plug-and-play controllable generation with discrete classifier guidance (Fig. 1C-3). Directly building a conditional model is prohibitive in most cases due to data scarcity. Thus, incorporating classifier guidance into continuous diffusion models (Dhariwal & Nichol, 2021a) proves particularly useful. This integration with pre-trained classifiers enables steering generation towards desired preferences. However, continuous classifier guidance requires valid definition of $\nabla_{\mathbf{x}} \log p_\theta(\mathbf{x})$, or "score" (Song & Ermon, 2019), which does not exist for discrete diffusion. Inspired by continuous diffusion classifier guidance and DiGress on guided graph diffusion (Vignac et al., 2022), here we introduce classifier-guided conditional generation for discrete diffusion LMs. Concretely, we want to sample from the conditional distribution of $q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{y}) \propto q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) q(\mathbf{y}|\mathbf{x}^{(t-1)})$, which is approximated by $p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) p_\phi(\mathbf{y}|\mathbf{x}^{(t-1)})$ where $p_\phi(\mathbf{y}|\mathbf{x}^{(t-1)})$ is a discriminative guidance model (classifier or regressor *w.r.t.* user's desired properties). However, $p_\phi(\mathbf{y}|\mathbf{x}^{(t-1)})$ cannot be factorized as a product over all positions, prohibiting evaluation of all possible values of $\mathbf{x}^{(t-1)}$. To this end, we resort to an approximation with first-order Taylor expansion around $\mathbf{x}^{(t)}$ (Dhariwal & Nichol, 2021a), where we treat $\mathbf{x}$ as a continuous one-hot variable on probability simplex to make $\nabla_{\mathbf{x}}$ a valid operator, thereby,

$$\begin{aligned}
&\log q(\mathbf{y}|\mathbf{x}^{(t-1)}) \\
&\approx \log q(\mathbf{y}|\mathbf{x}^{(t)}) + \langle \nabla_{\mathbf{x}} \log q(\mathbf{y}|\mathbf{x}^{(t)}), \mathbf{x}^{(t-1)} - \mathbf{x}^{(t)} \rangle \\
&\approx \textstyle\sum_{1 \le i \le L} \langle \nabla_{\mathbf{x}_i} \log q(\mathbf{y}|\mathbf{x}^{(t)}), \mathbf{x}_i^{(t-1)} \rangle + C(\mathbf{x}^{(t)}),
\end{aligned}$$

where $C(\mathbf{x}^{(t)})$ is a constant that does not depend on $\mathbf{x}^{(t-1)}$. We use $p_\phi(\mathbf{y}|\mathbf{x}^{(t)})$ to estimate $q(\mathbf{y}|\mathbf{x}^{(t)})$ and plug it into the above expression. We can now sample from the resulting conditional distribution instead at each timestep t,

$$\begin{aligned}
\mathbf{x}^{(t-1)} &\sim p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) p_\phi(\mathbf{y}|\mathbf{x}^{(t-1)})^\eta \\
&\propto p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) e^{\left(\eta \cdot \sum_i \langle \nabla_{\mathbf{x}_i} \log p_\phi(\mathbf{y}|\mathbf{x}^{(t)}), \mathbf{x}_{(i)}^{t-1} \rangle\right)},
\end{aligned} \tag{5}$$

where a tunable η controls the strength of guidance.

3.3 Comparisons with The Most Related Work

Comprehensive representations for protein sequence understanding are achieved by pre-training on protein sequence data via masked language modeling (Devlin et al., 2019), akin to language understanding. Among those, the family of ESM-1b/ESM2 (Rives et al., 2019; Lin et al., 2022) serves as the pioneer & cornerstone sequence embedding models for extensive protein predictive tasks. Therefore, DPLM follows the best practice of ESM2 in network architecture and pre-training strategies. DPLM takes a significant leap from ESM2 with immediate strong generative capabilities, without expensive needs for Monte Carlo methods (Verkuil et al., 2022) or Gibbs sampler (Johnson et al., 2021), which treat Masked-LM as Markov random fields (Wang & Cho, 2019). Besides, as verified from predictive experiments (§4.2), the generative ability of DPLM further enables its enhanced representation learning, echoing Richard Feynman's famous quote "*What I cannot create, I do not understand*".

Regarding protein sequence generation, EvoDiff (Alamdari et al., 2023) is the most relevant approach, which uses order-agnostic autoregressive diffusion models (OADM, Hoogeboom et al., 2021a) for unconditional generation, with conditional applications on intrinsic disordered sequence infilling and motif-scaffolding, whereas attaining better performance necessitates multiple sequence alignments (MSAs) based on a MSA-Transformer (Rao et al., 2021) parameterization. DPLM differs from EvoDiff in several aspects: (1) DPLM manifests superior representation learning, which, to the best of our knowledge, is the first time for protein diffusion

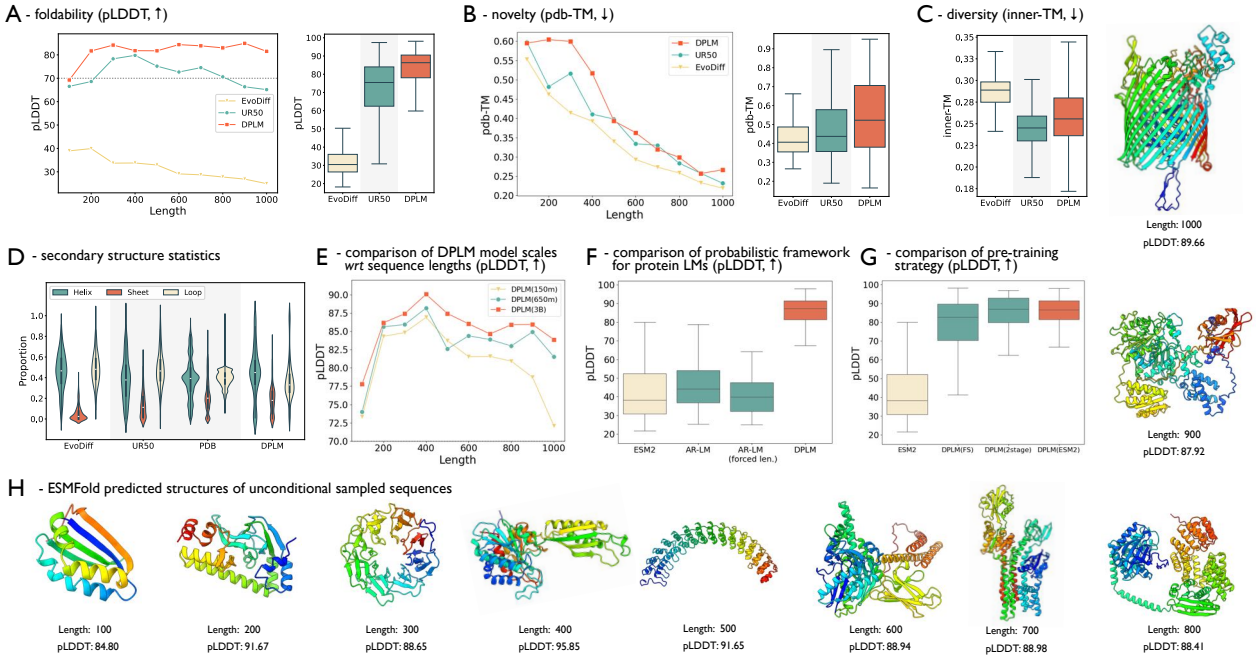

Figure 2. Evaluation of unconditional generation. Here we use ESMFold as the folding model to predict structures and calculate `pLDDT` for all the sampled sequences. We measure the (structural) novelty of the generated sequences against all known structures in PDB by `TM-score` (*i.e.*, `pdb-TM`, and measure the (structural) diversity within the sampled candidates for each model (*i.e.*, `inner-TM`).

models, even in general language learning regime, showing DPLM's appealing versatility, as shown in Tab. 1; (2) DPLM is based on a more principled discrete diffusion framework beyond the special (order-agnostic) autoregressive diffusion, which is not compatible with refining intermediate predictions and requires expensive $O(L)$ decoding overhead; (3) we investigate the ability of DPLM to accommodate extensive conditioning, especially conditioning on other modality and programmable generation steered by discrete classifier guidance, pushing steps forward beyond simple sequence conditioning investigated in EvoDiff paper. Please refer to Appendix §E for a more detailed discussion of the related work.

4 Experiments

We evaluate DPLM on extensive generative and understanding tasks, spanning unconditional generation (§4.1), a variety of protein predictive downstream tasks (§4.2), and conditional tasks, including motif-scaffolding (§4.3.1), inverse-folding task (§4.3.2), and secondary structure guided controllable generation (§4.3.3). We find that, in general, DPLM with larger model scales can attain better results than smaller ones, demonstrating the scaling law can also hold for protein language modeling. Please refer to the Appendix for more detailed experimental settings.

4.1 Evaluation of Unconditional Generation

Fig. 2 shows the results of DPLM for unconditional generation, where we evaluate the performance regarding a set of lengths $[100, 200, ..., 900, 1000]$ in intervals of 100. The reverse process of DPLM for sampling iterates for 500 steps. Meanwhile, we also randomly pick the natural sequences of the same length from UniRef50 as reference (denoted as UR50) We highlight our primary findings as follows:

(1) On Foldability: DPLM is capable of generating protein sequences with reasonable predicted structures. We examine the structural plausibility or foldability of protein sequences using the state-of-the-art single-sequence structure prediction model, *i.e.*, ESMFold (Lin et al., 2022), and measured by the predicted local distance difference test (`pLDDT`) score, which is considered high confidence if `pLDDT` > 70. We can find that protein sequences generated by DPLM achieve the highest `pLDDT` score across all lengths (Fig. 2A). Plus, secondary structure analysis of the sequences generated by DPLM reveals a higher proportion of beta-strands (Fig. 2D), and overall similar to the statistics of known protein structures in Protein Data Bank (PDB; Berman et al., 2000). Moreover, we can see that scaling DPLM leads to better foldability performance, especially for very long proteins (Fig. 2E).

(2) On Novelty. We investigate whether DPLM can sample sequences possessing novel structures, where we compare the structural similarity against known structures in PDB with `TMScore`. The highest TMscore is used to measure the novelty of each sequence, which we refer to as `pdb-TM` score. Overall, DPLM has relatively higher `pdbTM` than EvoDiff and natural sequences, as shown in Fig. 2B. Interestingly, the `pdbTM` score of DPLM will decrease as protein gets longer than 300 while maintaining the `pLDDT` > 75. This indicates that DPLM possesses the ability to sample

Table 1. Performance on various protein predictive downstream tasks. †: benchmarked results are quoted from Su et al. (2023).

Models	Thermostability	HumanPPI	Metal Ion Binding	EC	GO			DeepLoc		SSP
					MF	BP	CC	Subcellular	Binary	CASP12
	Spearman's ρ	Acc (%)	Acc (%)	Fmax	Fmax	Fmax	Fmax	Acc (%)	Acc (%)	Acc (%)
†SaProt (*structure provided)	0.724	86.41	75.75	0.884	0.678	0.356	0.414	85.57	93.55	-
†ESM-1b (Rives et al., 2019)	0.708	82.22	73.57	0.859	0.661	0.320	0.392	80.33	92.83	-
†MIF-ST (Yang et al., 2022b)	0.694	75.54	75.08	0.803	0.627	0.239	0.248	78.96	91.76	-
Masked-LM (ESM2-650M)	0.691	84.78	71.88	0.866	0.676	0.344	0.402	83.68	92.28	0.80
AR-LM (650M)	0.638	68.48	61.16	0.691	0.566	0.258	0.287	68.53	88.31	-
DPLM (150M)	0.687	80.98	72.17	0.822	0.662	0.328	0.379	82.41	92.63	-
DPLM (650M)	0.695	86.41	75.15	0.875	0.680	0.357	0.409	84.56	93.09	0.82
DPLM (3B)	**0.704**	**90.00**	**75.94**	**0.883**	**0.687**	**0.369**	**0.463**	**85.32**	**93.93**	-

sequences with structures not similar to PDB across various lengths, with the discrepancy becoming increasingly apparent as the sequence length extends.

(3) On Diversity. We quantify the diversity of sequences sampled by DPLM by `inner-TM` score. Specifically, for each sampled candidate, we use ESMFold to predict its structure and compute `TMscore` against the rest. The average `TMscore` is considered as the diversity. As shown in Fig. 2C, DPLM has a considerably low average `inner-TM`, demonstrating that the DPLM can synthesize structurally diverse sequences.

(4) On Learning: Discrete diffusion is the best-suited probabilistic framework for protein sequence generation, compared to Masked-LM and AR-LM. As shown in Fig. 2F, DPLM outperforms Masked-LM and AR-LM in terms of foldability, verifying our motivation to pursue a diffusion protein LM that diffusion is a more proper probabilistic framework for protein modeling. Moreover, AR-LM also falls short of precisely controlling the length of sampled sequences, making it less flexible in practice. As revealed in Fig. 2G, we find that despite attaining improved generation quality over ESM2 with directly pre-training DPLM from scratch (DPLM-FS), it can bring additional learning challenges and training overheads. As such, we leverage a 2-stage training strategy, which consists of masked language modeling as the first stage objective, followed by diffusion objective, solving this problem and obtaining high-quality generation with `pLDDT` closely approaching 90.

(5) Case Study. In Fig. 2H, we showcase proteins sampled by DPLM across various lengths, ranging from 100 to 1000, while more cases are presented in the Appendix. As the protein gets longer, the complexity of its structure will increase, containing rich helices and sheets. We also find that DPLM can sample proteins composed of tandem repeats such as beta-barrel or Kelch repeat domain.

4.2 Evaluation of Protein Representation Learning on Downstream Predictive Tasks

We evaluate DPLM across a variety of protein predictive tasks (Su et al., 2023; Dallago et al., 2021; Xu et al., 2022), including protein function prediction (Thermostability and Metal Ion Binding), protein localization prediction (DeepLoc), protein annotation prediction (EC and GO), protein-protein interaction prediction (HumanPPI), where we perform full-parameters supervised fine-tuning on each dataset. We also include linear probing for secondary structure from TAPE (Rao et al., 2019).

DPLM is a superior protein sequence representation learner. As demonstrated in Tab. 1, DPLM outperforms ESM2 across all tasks. This improved performance is due to the proposed diffusion pre-training, which requires DPLM to adeptly learn to reconstruct the native sequence from a varied proportion of masking, including very high noise level, in contrast to ESM2 of a fixed 15% masking ratio. Under this circumstance, it becomes a much more challenging missing amino acid reconstruction task encouraging the model to capture the deep dependencies from the very context. Besides, we surprisingly find that DPLM also closely approaches the performance of SaProt (Su et al., 2023), which is a structure-aware LM that incorporates explicitly protein structures based on Foldseek (van Kempen et al., 2023) and folding models like AlphaFold (Jumper et al., 2021). This implies that DPLM may implicitly learn the protein structures from massive sequence data. Integrating explicit structural information into DPLM like Su et al. (2023) may bring further benefits, which deserve further exploration. Our results substantiate our initial premise that DPLM gains a deeper understanding of protein through the generative learning process, *i.e.*, it learns to better understand proteins by learning to generate them, leading to improved predictive performance.

4.3 Evaluation of Conditional Generation

4.3.1 SEQUENCE-COND.: MOTIF-SCAFFOLDING

The goal of motif-scaffolding requires a valid scaffold to maintain the structure of the given motif such that the original function can be preserved. Here, we follow the experimental setting in Alamdari et al. (2023), where we (1) initially determine the length of a scaffold and fill the scaffold positions with the mask token; then (2) keep the motif fragment fixed during inference, and sample scaffold conditioned on the motif; and finally use OmegaFold (Wu et al., 2022b) to predict the structure of the sampled sequences. A scaffold is considered successful when it meets two condi-

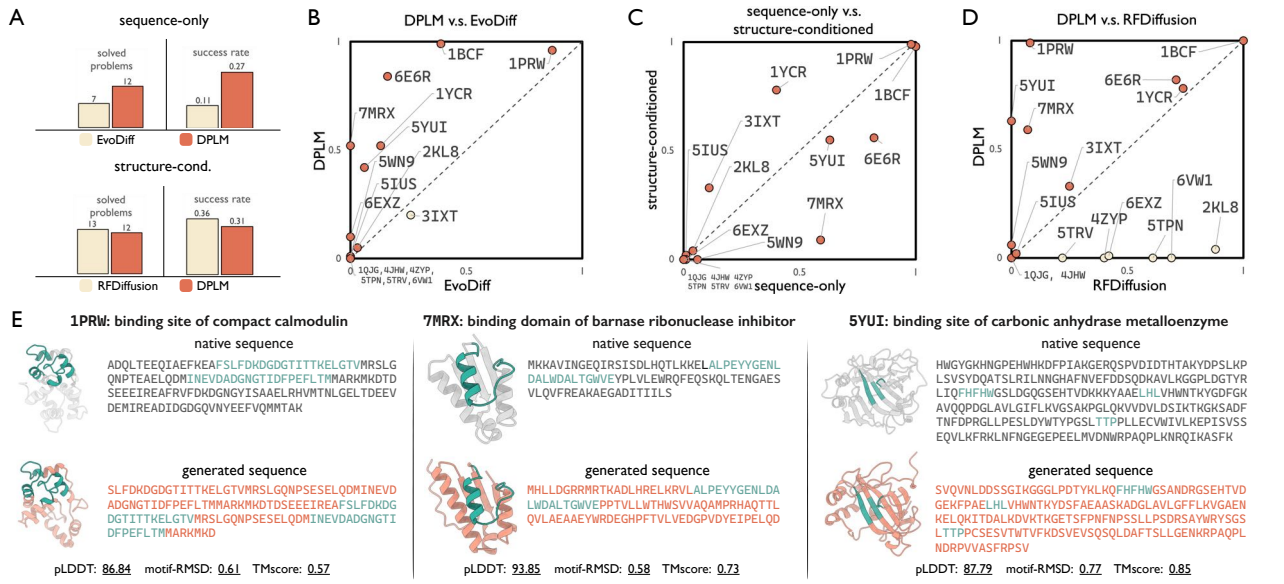

Figure 3. Evaluation of motif-scaffolding. **(A)** comparison regarding overall success rate and number of solved problems; **(B)** comparison between sequence-only approaches (DPLM *vs.* EvoDiff); **(C)** comparison between sequence-only *vs.* structure-conditioned DPLM; and **(D)** comparison between DPLM (structure-conditioned and sequence-only DPLM) *vs.* RFDffusion; **(E)** case study for three problems.

tions: (1) the RMSD between the predicted motif structure and the ground truth, referred to as motif-RMSD $< 1\AA$; and (2) the structure should have an overall pLDDT > 70. Overall, we examine 17 motif-scaffolding problems, and for each problem, we sample 100 sequences and then calculate the success rate according to the above criterion.

DPLM can generate reasonable scaffolds for the given functional motifs. As shown in Fig. 3, we find that DPLM outperforms EvoDiff in terms of the number of solved problems and the average success rate. Moreover, on the problems that both DPLM and EvoDiff can solve, the success rate of DPLM is higher than EvoDiff, except 3ixt. This indicates that DPLM excels in motif-scaffolding, preserving the motif structure during scaffold generation. To gain more insights, we compare DPLM with structure conditioning (see §4.3.2) with state-of-the-art structure designer RFDiffusion (Watson et al., 2023). We find that DPLM shows better results in 6 problems, especially for 1PRW and 5YUI). We find that utilizing motif structure helps DPLM make a further improvement on 4 problems compared to the original sequence-only DPLM, while decreasing performance on the other 6 problems. This implies that for some specific motifs, scaffolding in sequence space may be better. The detailed analysis unveiled a common biological property among the motifs observed in these two cases. Specifically, the motif sequence displayed a remarkable level of evolutionary conservation, playing pivotal roles in binding critical signal passengers (1PRW: calmodulin EF hand for calcium binding and 5YUI: carbonic anhydrase II for CO2 binding). Notably, the motif structures predominantly comprised flexible loops. Conversely, 5TPN, 6VW1, and 2KL8, which exhibited a distinct advantage in motif scaffolding as indicated by the RFdiffusion, featured rigid helical structures that lacked functional evolutionary conservations. This intriguing phenomenon suggests that DPLM holds great promise as a superior method for constructing structurally flexible yet evolutionarily conserved functional motif scaffolding.

Table 2. Performance comparison between DPLM *and different baseline approaches on CATH 4.2 and CATH 4.3 datasets.* DPLM's results are obtained by argmax decoding (*i.e.*, no sampling). †: benchmarked results are quoted from Gao et al. (2022b).

	Models	Trainable Params.	AAR	struct. eval. scTM	struct. eval. pLDDT
CATH 4.2	†StructTrans (Ingraham et al., 2019)	1.6M/1.6M	35.82	-	-
	†GVP (Jing et al., 2020)	1.0M/1.0M	39.47	-	-
	†ProteinMPNN (Dauparas et al., 2022)	1.9M/1.9M	45.96	-	-
	PiFold (Gao et al., 2022b)	6.6M/6.6M	51.66	-	-
	ProteinMPNN + CMLM	1.9M/1.9M	48.62	0.87	74.07
	LM-DESIGN (*w/* ProtMPNN encoder)	5.0M/650M	54.41	0.88	77.07
	DPLM (*w/* ProtMPNN encoder)	5.0M/650M	**54.54**	0.88	**77.12**
CATH 4.3	PiFold (Gao et al., 2022b)	6.6M/6.6M	51.66	-	-
	GVP-Transformer (Hsu et al., 2022)	142M/142M	51.60	-	-
	LM-DESIGN (*w/* GVP-Trans encoder)	6.3M/650M	56.49	0.85	74.89
	DPLM-150M (*w/* GVPTrans encoder)	3.1M/150M	53.27	0.85	**75.31**
	DPLM-650M (*w/* GVPTrans encoder)	6.3M/650M	56.61	0.86	76.78
	DPLM-3B (*w/* GVPTrans encoder)	68.2M/3.0B	**59.44**	**0.86**	**77.12**

4.3.2 STRUCTURE-CONDITIONED: INVERSE FOLDING

The goal of inverse folding is to find an amino acid sequence that can fold to a given protein backbone structure. We follow LM-DESIGN (Zheng et al., 2023b) to implant a structural adapter into the last network layer of DPLM, and use GVP-Transformer Encoder (Hsu et al., 2022) as the expert protein backbone structure encoder. We assess DPLM on CATH 4.2 and 4.3 (Orengo et al., 1997). We use amino acid recovery (AAR) for sequence evaluation, whilst for structure evaluation, we first predict the structure

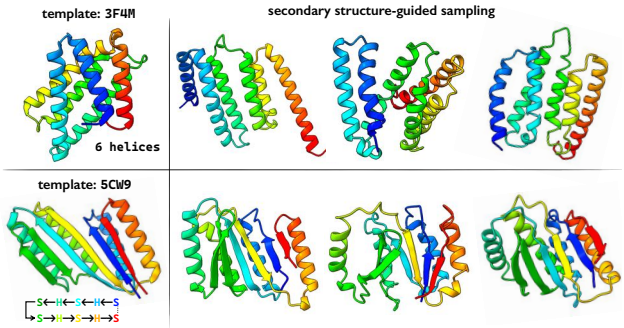

Figure 4. Secondary structure guided conditional sampling. The first case contains 6 alpha-helices, The second case is much more complicated as a globally twisted structure with interleaved alpha-helices and beta-strands, where the N-terminus and C-terminus are structurally contiguous.

of the generated sequence using ESMFold, then calculate the pLDDT score and self-consistency TM-score (scTM) between predicted structure and the input one.

DPLM yields sequences that can confidently fold into the given backbone structure. As shown in Tab. 2, DPLM can outperform or be on par with our strong baselines, including the state-of-the-art approach LM-DESIGN (Zheng et al., 2023b), manifesting in AAR, and most importantly, decent performance regarding structure evaluation (scTM $= 0.85$ and pLDDT > 76). We suggest this derives from the well-learned protein sequence knowledge of DPLM. When given structure backbone information, DPLM can leverage this advantage and generate the sequence whose structure is both plausible and similar to the reference.

4.3.3 CONTROLLABLE GENERATION: SECONDARY STRUCTURE GUIDED PROTEIN SAMPLING

Classifier guidance is preferred for its flexible control over the generation process without retraining for each new condition, especially beneficial in scenarios with too limited labeled data to directly attain conditional models. Here we showcase how to guide DPLM to generate proteins satisfying desired secondary structures. We train a secondary structure prediction (SSP) model as a sequence labeling task on TAPE dataset. We then integrate this SSP discriminative model into DPLM to provide guiding signals.

DPLM enjoys plug-and-play programmability. Fig. 4 showcases that the proposed discrete classifier guidance helps steer a pre-trained DPLM to generate samples satisfying provided secondary structure annotations extracted from template natural proteins. These findings suggest that DPLM is highly programmable, and its full potential of generative capabilities can be realized in a plug-and-play fashion, indicating that DPLM preserves the appealing characteristic of controllable generation inherent in diffusion models, but for discrete data. This flexibility to swiftly adapt to the evolving needs of users across a broad spectrum of preferences is also significant in practical applications with time and computational paramount.

5 Discussions

In this paper, we introduce diffusion protein LM (DPLM), a versatile protein LM that is capable of both protein sequence generation and representation learning. We further develop several conditioning strategies for various needs of conditional generation, including sequence conditioning, cross-modal conditioning, and programmable generation with plug-and-play discrete classifier guidance.

Despite these promising results, there remain several limitations and future work directions deserving to be explored.

(i) *Exploring* DPLM*'s conditional generation for wider applications.* We can further extend the cross-modal conditioning strategy of DPLM to more diverse modalities as conditioners, including MSA-conditioned homologous sequence generation, small molecule-conditioned binder design for ligands, antigen-conditioned antibody CDR design, among others. Also, the inclusion of demonstrations featuring plug-and-play classifier-guided controllable generation is essential for more scenarios toward diverse user preferences, *e.g.*, structural symmetry, superfamily, binding affinity, thermostability, fluorescence, and beyond.

(ii) DPLM *can further benefit from best practices of cutting-edge technical advancement in the vastness of large language models (LLMs).* For example, **(1)** long context extension (Chen et al., 2023b) can rapidly adapt DPLM to handle very long proteins beyond its training length limit, and offering potential for modeling exceptionally long biological sequences such as DNAs and RNAs, unifying and deciphering the languages associated with the central dogma of life; **(2)** fine-tuning DPLM with human feedback or even wet-lab experimental feedback, leveraging reinforcement learning (RL; Ouyang et al., 2022), direct preference optimization (DPO; Rafailov et al., 2024), and self-play fine-tuning (Chen et al., 2024b); **(3)** eliciting instruction-following and in-context learning (Wei et al., 2022a) analogs for protein LMs can also be a promising direction would fully harness DPLM's learned knowledge.

(iii) *It is imperative to integrate protein structure modeling into* DPLM. The advance of protein structure modeling manifest tremendous success, including AlphaFold (Jumper et al., 2021), ESMFold (Lin et al., 2022) for structure prediction, RFDIffusion (Watson et al., 2023), Chroma (Ingraham et al., 2023) for structure design, and even full-atom molecular modeling, *e.g.*, the latest generation of AlphaFold (DeepMind, 2023) and RF-AA (Krishna et al., 2023). Developing a universal protein language model with the next-generation DPLM, which accounts for both sequence and structure, is a particularly promising avenue.

We leave these exciting directions as future work.

Impact Statement

Our work on protein generation and representation learning can be used in developing potent therapeutic macromolecues such as antibodies and accelerate the research process of drug discovery. Our method may be adapted to other scenarios of computer-aided design, such as small molecule design, material design, and chip design. It is also needed to ensure the responsible use of our method and refrain from using it for harmful purposes.

Acknowledgements

We thank anonymous reviewers for their insightful feedback. We would like to especially thank Dr. Hang Li for insightful discussions on the project and feedback on the manuscript that help shape this study. We thank Yi Zhou, Jing Yuan, Dr. Yilai Li and Jiasheng Ye for their valuable comments.

References

Alamdari, S., Thakkar, N., van den Berg, R., Lu, A. X., Fusi, N., Amini, A. P., and Yang, K. K. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pp. 2023–09, 2023.

Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993, 2021.

Bengio, Y., Ducharme, R., and Vincent, P. A neural probabilistic language model. *Advances in neural information processing systems*, 13, 2000.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., and Bourne, P. E. The protein data bank. *Nucleic acids research*, 28(1): 235–242, 2000.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110, 2022.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. volume 33, pp. 1877–1901, 2020.

Chen, J., Zhang, A., Li, M., Smola, A., and Yang, D. A cheaper and better diffusion language model with soft-masked noise. *arXiv preprint arXiv:2304.04746*, 2023a.

Chen, S., Wong, S., Chen, L., and Tian, Y. Extending context window of large language models via positional interpolation. *arXiv preprint arXiv:2306.15595*, 2023b.

Chen, X., Liu, Z., Xie, S., and He, K. Deconstructing denoising diffusion models for self-supervised learning. *arXiv preprint arXiv:2401.14404*, 2024a.

Chen, Z., Deng, Y., Yuan, H., Ji, K., and Gu, Q. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024b.

Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

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*, pp. 2021–11, 2021.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N., et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378 (6615):49–56, 2022.

Dauparas, J., Lee, G. R., Pecoraro, R., An, L., Anishchenko, I., Glasscock, C., and Baker, D. Atomic context-conditioned protein sequence design using ligandmpnn. *Biorxiv*, pp. 2023–12, 2023.

De Bortoli, V., Mathieu, E., Hutchinson, M., Thornton, J., Teh, Y. W., and Doucet, A. Riemannian score-based generative modelling. *Advances in Neural Information Processing Systems*, 35:2406–2422, 2022.

DeepMind, G. Performance and structural coverage of the latest, in-development alphafold model. 2023.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021a.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021b.

Dieleman, S., Sartran, L., Roshannai, A., Savinov, N., Ganin, Y., Richemond, P. H., Doucet, A., Strudel, R., Dyer, C., Durkan, C., et al. Continuous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

Ferruz, N. and Höcker, B. Controllable protein design with language models. *Nature Machine Intelligence*, 4(6): 521–532, 2022.

Ferruz, N., Schmidt, S., and Höcker, B. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

Fu, Y., Peng, H., Ou, L., Sabharwal, A., and Khot, T. Specializing smaller language models towards multi-step reasoning. *arXiv preprint arXiv:2301.12726*, 2023.

Gao, Z., Guo, J., Tan, X., Zhu, Y., Zhang, F., Bian, J., and Xu, L. Difformer: Empowering diffusion model on embedding space for text generation. *arXiv preprint arXiv:2212.09412*, 2022a.

Gao, Z., Tan, C., and Li, S. Z. Pifold: Toward effective and efficient protein inverse folding. *arXiv preprint arXiv:2209.12643*, 2022b.

Ghazvininejad, M., Levy, O., Liu, Y., and Zettlemoyer, L. Mask-predict: Parallel decoding of conditional masked language models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6112–6121, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1633. URL https://www.aclweb.org/anthology/D19-1633.

Gong, S., Li, M., Feng, J., Wu, Z., and Kong, L. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*, 2022.

Gu, J. and Kong, X. Fully non-autoregressive neural machine translation: Tricks of the trade. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 120–133, 2021.

Gu, J., Bradbury, J., Xiong, C., Li, V. O., and Socher, R. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*, 2018.

Guo, J., Tan, X., Xu, L., Qin, T., Chen, E., and Liu, T.-Y. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7839–7846, 2020a.

Guo, J., Zhang, Z., Xu, L., Wei, H.-R., Chen, B., and Chen, E. Incorporating bert into parallel sequence decoding with adapters. *Advances in Neural Information Processing Systems*, 33:10843–10854, 2020b.

Han, X., Kumar, S., and Tsvetkov, Y. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. *arXiv preprint arXiv:2210.17432*, 2022.

He, L., Zhang, S., Wu, L., Xia, H., Ju, F., Zhang, H., Liu, S., Xia, Y., Zhu, J., Deng, P., et al. Pre-training co-evolutionary protein representation via a pairwise masked language model. *arXiv preprint arXiv:2110.15527*, 2021.

He, Z., Sun, T., Wang, K., Huang, X., and Qiu, X. Diffusionbert: Improving generative masked language models with diffusion models. 2023.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D. P., Poole, B., Norouzi, M., Fleet, D. J., et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Hoogeboom, E., Gritsenko, A. A., Bastings, J., Poole, B., van den Berg, R., and Salimans, T. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2021a.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., and Welling, M. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021b.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International Conference on Machine Learning*, pp. 8867–8887. PMLR, 2022.

Hsu, C., Verkuil, R., Liu, J., Lin, Z., Hie, B., Sercu, T., Lerer, A., and Rives, A. Learning inverse folding from millions of predicted structures. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8946–8970. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/hsu22a.html.

Hu, M., Yuan, F., Yang, K. K., Ju, F., Su, J., Wang, H., Yang, F., and Ding, Q. Exploring evolution-aware &-free protein language models as protein function predictors. In *Advances in Neural Information Processing Systems*, 2022.

Huang, F., Ke, P., and Huang, M. Directed acyclic transformer pre-training for high-quality non-autoregressive text generation. *Transactions of the Association for Computational Linguistics*, 2023.

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Generative models for graph-based protein design. In *Advances in neural information processing systems*, 2019.

Ingraham, J. B., Baranov, M., Costello, Z., Barber, K. W., Wang, W., Ismail, A., Frappier, V., Lord, D. M., Ng-Thow-Hing, C., Van Vlack, E. R., et al. Illuminating protein space with a programmable generative model. *Nature*, pp. 1–9, 2023.

Jing, B., Eismann, S., Suriana, P., Townshend, R. J. L., and Dror, R. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2020.

Johnson, S. R., Monaco, S., Massie, K., and Syed, Z. Generating novel protein sequences using gibbs sampling of masked language models. *bioRxiv*, pp. 2021–01, 2021.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Kim, J., Kong, J., and Son, J. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In *International Conference on Machine Learning*, pp. 5530–5540. PMLR, 2021.

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., and Iwasawa, Y. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Krishna, R., Wang, J., Ahern, W., Sturmfels, P., Venkatesh, P., Kalvet, I., Lee, G. R., Morey-Burrows, F. S., Anishchenko, I., Humphreys, I. R., et al. Generalized biomolecular modeling and design with rosettafold all-atom. *bioRxiv*, pp. 2023–10, 2023.

Lee, J. S., Kim, J., and Kim, P. M. Proteinsgm: Score-based generative modeling for de novo protein design. *bioRxiv*, pp. 2022–07, 2022.

Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. Diffusion-lm improves controllable text generation. In *Advances in Neural Information Processing Systems*, volume abs/2205.14217, 2022.

Lin, Y. and AlQuraishi, M. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. *arXiv preprint arXiv:2301.12485*, 2023.

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.

Lisanza, S. L., Gershon, J. M., Tipps, S. W. K., Arnoldt, L., Hendel, S., Sims, J. N., Li, X., and Baker, D. Joint generation of protein sequence and structure with rosettafold sequence space diffusion. *bioRxiv*, pp. 2023–05, 2023.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Lu, A. X., Zhang, H., Ghassemi, M., and Moses, A. Self-supervised contrastive learning of protein representations by mutual information maximization. *BioRxiv*, pp. 2020–09, 2020.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos Jr, J. L., Xiong, C., Sun, Z. Z., Socher, R., et al. Deep neural language modeling enables functional protein generation across families. *bioRxiv*, pp. 2021–07, 2021.

McDermott, M., Yap, B., Hsu, H., Jin, D., and Szolovits, P. Adversarial contrastive pre-training for protein sequences. *arXiv preprint arXiv:2102.00466*, 2021.

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. In *Advances in Neural Information Processing Systems*, pp. 29287–29303, 2021.

Melnyk, I., Chenthamarakshan, V., Chen, P.-Y., Das, P., Dhurandhar, A., Padhi, I., and Das, D. Reprogramming large pretrained language models for antibody sequence infilling. *arXiv preprint arXiv:2210.07144*, 2022.

Meshchaninov, V., Strashnov, P., Shevtsov, A., Nikolaev, F., Ivanisenko, N., Kardymon, O., and Vetrov, D. Diffusion on language model embeddings for protein sequence generation. *arXiv preprint arXiv:2403.03726*, 2024.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y. (eds.), *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

Min, S., Park, S., Kim, S., Choi, H.-S., Lee, B., and Yoon, S. Pre-training of deep bidirectional protein sequence representations with structural information. *IEEE Access*, 9:123912–123926, 2021.

Muennighoff, N., Rush, A. M., Barak, B., Scao, T. L., Piktus, A., Tazi, N., Pyysalo, S., Wolf, T., and Raffel, C. Scaling data-constrained language models. *arXiv preprint arXiv:2305.16264*, 2023.

Nambiar, A., Heflin, M., Liu, S., Maslov, S., Hopkins, M., and Ritz, A. Transforming the language of life: transformer neural networks for protein prediction tasks. In *Proceedings of the 11th ACM international conference on bioinformatics, computational biology and health informatics*, pp. 1–8, 2020.

Nijkamp, E., Ruffolo, J., Weinstein, E. N., Naik, N., and Madani, A. Progen2: exploring the boundaries of protein language models. *arXiv preprint arXiv:2206.13517*, 2022.

Nourani, E., Asgari, E., McHardy, A. C., and Mofrad, M. R. Tripletprot: deep representation learning of proteins based on siamese networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 19 (6):3744–3753, 2021.

OpenAI. Gpt-4 technical report, 2023.

Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B., and Thornton, J. M. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Qian, L., Zhou, H., Bao, Y., Wang, M., Qiu, L., Zhang, W., Yu, Y., and Li, L. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1993–2003, Online, August 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.155. URL https://aclanthology.org/2021.acl-long.155.

Qian, L., Zhou, Y., Zheng, Z., Zhu, Y., Lin, Z., Feng, J., Cheng, S., Li, L., Wang, M., and Zhou, H. The volctrans glat system: Non-autoregressive translation meets wmt21. *WMT 2021*, pp. 187, 2021b.

Qian, L., Wang, M., Liu, Y., and Zhou, H. Diff-glat: Diffusion glancing transformer for parallel sequence to sequence learning. *arXiv preprint arXiv:2212.10240*, 2022.

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32, 2019.

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., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019. doi: 10.1101/622803. URL https://www.biorxiv.org/content/10.1101/622803v4.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2021.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Le Scao, T., Raja, A., et al. Multitask prompted training enables zero-shot task generalization. In *ICLR 2022-Tenth International Conference on Learning Representations*, 2022.

Savinov, N., Chung, J., Binkowski, M., Elsen, E., and van den Oord, A. Step-unrolled denoising autoencoders for text generation. In *International Conference on Learning Representations*, 2021.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Bach, F. and Blei, D. (eds.), *International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR, PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

Strodthoff, N., Wagner, P., Wenzel, M., and Samek, W. Udsmprot: universal deep sequence models for protein classification. *Bioinformatics*, 36(8):2401–2409, 2020.

Sturmfels, P., Vig, J., Madani, A., and Rajani, N. F. Profile prediction: An alignment-based pre-training task for protein sequence models. *arXiv preprint arXiv:2012.00195*, 2020.

Su, J., Lu, Y., Pan, S., Murtadha, A., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Su, J., Han, C., Zhou, Y., Shan, J., Zhou, X., and Yuan, F. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pp. 2023–10, 2023.

Sun, T. and Qiu, X. Moss. https://github.com/OpenLMLab/MOSS, 2023.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, volume 27, pp. 3104–3112, 2014.

Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and Consortium, U. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. Llama: Open and efficient foundation language models, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R., and Jaakkola, T. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

Unsal, S., Atas, H., Albayrak, M., Turhan, K., Acar, A. C., and Doğan, T. Learning functional properties of proteins with language models. *Nature Machine Intelligence*, 4 (3):227–245, 2022.

van Kempen, M., Kim, S. S., Tumescheit, C., Mirdita, M., Lee, J., Gilchrist, C. L., Söding, J., and Steinegger, M. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, pp. 1–4, 2023.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information*

Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, volume 30, pp. 5998–6008, 2017.

Verkuil, R., Kabeli, O., Du, Y., Wicky, B. I., Milles, L. F., Dauparas, J., Baker, D., Ovchinnikov, S., Sercu, T., and Rives, A. Language models generalize beyond natural proteins. *bioRxiv*, pp. 2022–12, 2022.

Vignac, C., Krawczuk, I., Siraudin, A., Wang, B., Cevher, V., and Frossard, P. Digress: Discrete denoising diffusion for graph generation. In *The Eleventh International Conference on Learning Representations*, 2022.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A., and Bottou, L. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

Wang, A. and Cho, K. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pp. 30–36, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-2304. URL https://www.aclweb.org/anthology/W19-2304.

Wang, Y., He, S., Chen, G., Chen, Y., and Jiang, D. Xlm-d: Decorate cross-lingual pre-training model as non-autoregressive neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 6934–6946, 2022.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F., et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976): 1089–1100, 2023.

Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2021.

Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022a.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E. H., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022b.

Wettig, A., Gao, T., Zhong, Z., and Chen, D. Should you mask 15% in masked language modeling? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 2985–3000, 2023.

Wu, K. E., Yang, K. K., Berg, R. v. d., Zou, J. Y., Lu, A. X., and Amini, A. P. Protein structure generation via folding diffusion. *arXiv preprint arXiv:2209.15611*, 2022a.

Wu, R., Ding, F., Wang, R., Shen, R., Zhang, X., Luo, S., Su, C., Wu, Z., Xie, Q., Berger, B., et al. High-resolution de novo structure prediction from primary sequence. *BioRxiv*, pp. 2022–07, 2022b.

Wu, T., Fan, Z., Liu, X., Gong, Y., Shen, Y., Jiao, J., Zheng, H.-T., Li, J., Wei, Z., Guo, J., et al. Ar-diffusion: Auto-regressive diffusion model for text generation. *arXiv preprint arXiv:2305.09515*, 2023.

Xiao, Y., Qiu, J., Li, Z., Hsieh, C.-Y., and Tang, J. Modeling protein using large-scale pretrain language model. *arXiv preprint arXiv:2108.07435*, 2021.

Xu, M., Zhang, Z., Lu, J., Zhu, Z., Zhang, Y., Chang, M., Liu, R., and Tang, J. Peer: a comprehensive and multi-task benchmark for protein sequence understanding. *Advances in Neural Information Processing Systems*, 35: 35156–35173, 2022.

Yang, K. K., Wu, Z., and Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nature methods*, 16(8):687–694, 2019.

Yang, K. K., Lu, A. X., and Fusi, N. Convolutions are competitive with transformers for protein sequence pre-training. *bioRxiv*, pp. 2022–05, 2022a.

Yang, K. K., Zanichelli, N., and Yeh, H. Masked inverse folding with sequence transfer for protein representation learning. *bioRxiv*, pp. 2022–05, 2022b.

Ye, J., Zheng, Z., Bao, Y., Qian, L., and Gu, Q. Diffusion language models can perform many tasks with scaling and instruction-finetuning. *arXiv preprint arXiv:2308.12219*, 2023a.

Ye, J., Zheng, Z., Bao, Y., Qian, L., and Wang, M. Dinoiser: Diffused conditional sequence learning by manipulating noises. *arXiv preprint arXiv:2302.10025*, 2023b.

Yi, K., Zhou, B., Shen, Y., Lio, P., and Wang, Y. G. Graph denoising diffusion for inverse protein folding. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=u4YXKKG5dX.

Yim, J., Trippe, B. L., De Bortoli, V., Mathieu, E., Doucet, A., Barzilay, R., and Jaakkola, T. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023.

Yuan, H., Yuan, Z., Tan, C., Huang, F., and Huang, S. Seqdiffuseq: Text diffusion with encoder-decoder transformers. *arXiv preprint arXiv:2212.10325*, 2022.

Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

Zheng, L., Yuan, J., Yu, L., and Kong, L. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023a.

Zheng, Z., Deng, Y., Xue, D., Zhou, Y., YE, F., and Gu, Q. Structure-informed language models are protein designers. In *International Conference on Machine Learning*, 2023b.

A Reparameterizaed Discrete Diffusion Models (RDM)

DPLM uses reparameterized discrete diffusion model (RDM) as its discrete diffusion framework (Zheng et al., 2023a). Here we briefly summarize its basic training and sampling. Please refer to Zheng et al. (2023a) for more details.

Zheng et al. (2023a) shows that the backward transition of discrete diffusion models $q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)})$ can be rewritten as

$$q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)}) = \begin{cases} \lambda_{t-1}^{(1)}\mathbf{x}^{(t)} + (1-\lambda_{t-1}^{(1)})\mathbf{q}_{\text{noise}}, & \text{if } \mathbf{x}^{(t)} = \mathbf{x}^{(0)} \\ \lambda_{t-1}^{(2)}\mathbf{x}^{(0)} + (1-\lambda_{t-1}^{(2)})\mathbf{q}_{\text{noise}}(\mathbf{x}^{(t)}), & \text{if } \mathbf{x}^{(t)} \neq \mathbf{x}^{(0)} \end{cases}$$

where $\mathbf{q}_{\text{noise}}(\mathbf{x}^{(t)}) = \beta_t \mathbf{x}^{(t)} + (1-\beta_t)\mathbf{q}_{\text{noise}}$, and both $\lambda_{t-1}^{(1)}$ and $\lambda_{t-1}^{(2)}$ are constants relating to β_t and β_{t-1}. This reformulation interprets the backward transition as a mixture distribution. Sampling from it is equivalent to first sampling from a Bernoulli distribution and then the corresponding component distribution, *i.e.*,

$$v_{t-1}^{(1)} \sim \text{Bernoulli}\left(\lambda_{t-1}^{(1)}\right), \mathbf{u}_t^{(1)} \sim \texttt{Cat}\left(\mathbf{u}; \mathbf{p} = \mathbf{q}_{\text{noise}}\right),$$
$$v_{t-1}^{(2)} \sim \text{Bernoulli}\left(\lambda_{t-1}^{(2)}\right), \mathbf{u}_t^{(2)} \sim \texttt{Cat}\left(\mathbf{u}; \mathbf{p} = \mathbf{q}_{\text{noise}}(\mathbf{x}_t)\right),$$

$$\mathbf{x}^{(t-1)} = \begin{cases} v_{t-1}^{(1)}\mathbf{x}^{(t)} + \left(1 - v_{t-1}^{(1)}\right)\mathbf{u}_t^{(1)}, & \text{if } \mathbf{x}^{(t)} = \mathbf{x}^{(0)} \\ v_{t-1}^{(2)}\mathbf{x}^{(0)} + \left(1 - v_{t-1}^{(2)}\right)\mathbf{u}_t^{(2)}, & \text{if } \mathbf{x}^{(t)} \neq \mathbf{x}^{(0)} \end{cases}.$$

This reparameterizes the transitions $q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)})$ and $p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})$ into $q(\mathbf{x}^{(t-1)}, \mathbf{v}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)})$ and $p_\theta(\mathbf{x}^{(t-1)}, \mathbf{v}^{(t-1)}|\mathbf{x}^{(t)})$. With this reparameterization, the training objective of diffusion models (*i.e.*, the variational bound of negative log-likelihood) becomes

$$-\mathbb{E}_q(\mathbf{x}_{1:T}, \mathbf{v}_{1:T}|\mathbf{x}_0)\left[\log \frac{p_\theta(\mathbf{x}_0, \mathbf{x}_{1:T}, \mathbf{v}_{1:T})}{q(\mathbf{x}_{1:T}, \mathbf{v}_{1:T}|\mathbf{x}_0)}\right]$$
$$= \mathcal{J}_1 + \sum_{t=2}^{T} \mathcal{J}_t + \text{const.},$$

where $\mathcal{J}_1 = -\mathbb{E}_{q(\mathbf{x}_1|\mathbf{x}_0)}\left[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)\right]$ and Zheng et al. (2023a) shows that $\mathcal{J}_t$ can be simplified into a weighted cross-entropy loss. Since each token is modeled conditionally independently, so we can consider the backward transition for each token, and sum the losses for them. For i-th position, the backward transition is $q(\mathbf{x}_i^{(t-1)}, \mathbf{v}_i^{(t-1)}|\mathbf{x}_i^{(t)}, \mathbf{x}_i^{(0)})$. As shown in Zheng et al. (2023a) appendix C, the loss at i-th token can be written as

$$\mathcal{J}_{t,i} = \mathbb{E}_{q(\mathbf{v}_i^{(t-1)})}\left[\text{KL}[q(\mathbf{x}_i^{(t-1)}|\mathbf{v}_i^{(t-1)}, \mathbf{x}_i^{(t)}, \mathbf{x}_i^{(0)})||p_\theta(\mathbf{x}_i^{(t-1)}|\mathbf{v}_i^{(t-1)}, \mathbf{x}_i^{(t)})]\right]$$

Let $b_i(t) = \mathbf{1}_{x_i^{(t)} \neq x_i^{(0)}}$, $q(\mathbf{x}_i^{(t-1)}|\mathbf{v}_i^{(t-1)}, \mathbf{x}_i^{(t)}, \mathbf{x}_i^{(0)})$ can be written as:

$$q(\mathbf{x}_i^{(t-1)}|\mathbf{v}_i^{(t-1)}, \mathbf{x}_i^{(t)}, \mathbf{x}_i^{(0)}) = \begin{cases} v_{t-1,i}^{(1)}\mathbf{x}_i^{(t)} + (1 - v_{t-1,i}^{(1)})\mathbf{q}_{\text{noise}} & \text{if } b_i(t) = 0, \\ v_{t-1,i}^{(2)}\mathbf{x}_i^{(0)} + (1 - v_{t-1,i}^{(2)})\mathbf{q}_{\text{noise}} & \text{if } b_i(t) = 1, \end{cases}$$

And $p_\theta(\mathbf{x}_i^{(t-1)}|\mathbf{v}_i^{(t-1)}, \mathbf{x}_i^{(t)})$ can be written as:

$$p_\theta(\mathbf{x}_i^{(t-1)}|\mathbf{v}_i^{(t-1)}, \mathbf{x}_i^{(t)}) = \begin{cases} v_{t-1,i}^{(1)}\mathbf{x}_i^{(t)} + (1 - v_{t-1,i}^{(1)})\mathbf{q}_{\text{noise}} & \text{if } b_i(t) = 0, \\ v_{t-1,i}^{(2)}p_\theta(\mathbf{x}_i^{(0)}|\mathbf{x}^{(t)}) + (1 - v_{t-1,i}^{(2)})\mathbf{q}_{\text{noise}} & \text{if } b_i(t) = 1, \end{cases}$$

Therefore, the loss at i-th token can be computed by enumerating all cases with respect to $\mathbf{v}_i^{(t-1)}$ and $b_i(t)$. As noted in Zheng et al. (2023a), the KL divergence is equal to $-\log p_\theta(x_i^{(0)}|x^{(t)})$ when $v_{t-1,i}^{(2)} = 1$ and $b_i(t) = 1$, while in other cases the KL divergence is 0. So we have:

$$\begin{aligned} \mathcal{J}_t &= \sum_{1 \le i \le L} \mathcal{J}_{t,i} \\ &= \sum_{1 \le i \le L} \mathbb{E}_{q(\mathbf{v}_i^{(t-1)})}\Big[\\ &\text{KL}[q(\mathbf{x}_i^{(t-1)}|\mathbf{v}_i^{(t-1)}, \mathbf{x}_i^{(t)}, \mathbf{x}_i^{(0)})||p_\theta(\mathbf{x}_i^{(t-1)}|\mathbf{v}_i^{(t-1)}, \mathbf{x}_i^{(t)})]\Big] \\ &= \sum_{1 \le i \le L} q(\mathbf{v}_i^{(t-1)} = 1) \cdot b_i(t) \cdot (-\log p_\theta(x_i^{(0)}|x^{(t)})) \\ &= -\lambda^{(t-1)} \sum_{1 \le i \le L} b_i(t) \cdot \log p_\theta(\mathbf{x}_i^{(0)}|\mathbf{x}^{(t)}) \end{aligned}$$

Notably, training with different noise schedules only differs in the weighting of the objective.

During sampling, RDM leverages this observation and proposes to employ a discriminative approach. Specifically, it denoises a token only when it receives a top-k score (log-probability) from the network where k in each step is determined by a denoising schedule. The overall sampling process is shown in algorithm 1.

B Training Stratety

B.1 Pre-training of DPLM

During the training phase, we investigate two different approaches: (1) training with the diffusion objective from scratch, and (2) what we refer to as two-stage training, which consists of initial training with the masked language modeling (MLM) objective followed by continuous training with the diffusion objective.

Empirical Observation. In our preliminary experiments, we observed that discrete diffusion pre-training "from scratch (FS)" often yielded instability in the form of frequent loss spiking, therefore hurting our model performance.

Algorithm 1 Sampling from RDM

Input: trained network $f_\theta(\cdot)$ and temperature τ.
Output: generated sample $\mathbf{x}^{(0)}$.
for $n = 1, 2, \ldots, N$ **do**
  Initialize $\mathbf{x}_{T,n} \sim q_{\text{noise}}$;
  Initialize $b_{T,n} = 0$;
end for
for $t = T, \ldots, 1$ **do**
  for $n = 1, 2, \ldots, N$ **do**
    Draw $\widetilde{\mathbf{x}}_{0,n} \sim \text{Categorical}\left(f_\theta\left(\mathbf{x}_{t,n}\right)/\tau\right)$;
    Generate $\mathbf{v}_{t-1,n}$ according to $\log p(\widetilde{\mathbf{x}}_{0,n})$
    if $b_{t,n} = 1$ **then**
      Draw $\mathbf{u}^{(1)}_{t,n} \sim q_{\text{noise}}$;
      $\mathbf{x}_{t-1,n} = v^{(1)}_{t-1,n}\mathbf{x}_{t,n} + \left(1 - v^{(1)}_{t-1,n}\right)\mathbf{u}^{(1)}_{t,n}$;
    else
      Draw $\mathbf{u}^{(2)}_{t,n} \sim q_{\text{noise}}(\mathbf{x}_{t,n})$;
      $\mathbf{x}_{t-1,n} = v^{(2)}_{t-1,n}\widetilde{\mathbf{x}}_{0,n} + \left(1 - v^{(2)}_{t-1,n}\right)\mathbf{x}^{(2)}_{t,n}$;
    end if
    Let $b_{t-1,n} = b_{t,n} \wedge v^{(1)}_{t-1,n} \vee v^{(2)}_{t-1,n}$;
  end for
end for
Return $\mathbf{x}_{0,1:N}$.

Our Hypothesis. We noticed that the absorbing diffusion objective leads to a variable masking ratio ranging from 0% to 100%, while conventional MLM objective's masking ratio keep fixed at 15% such that a masked LM is always exposed to rich condition of 85% observation/context. In other words, in contrast to MLM, for absorbing discrete diffusion models like DPLM, in some of the extreme cases there are nearly all tokens getting masked, which means that the model is required to recover all tokens of the ground-truth sequence from nothing). This could impose severe learning challenges, especially at the early phase of pre-training, where the model has yet not acquired sufficient capability to extract informative features and correlations from limited observation.

Solution in Principle. Inspired by the success of curriculum learning in non-autoregressive text generation (Qian et al., 2021a; Gu & Kong, 2021; Guo et al., 2020a;b; Wang et al., 2022), we suggested that a masking warmup strategy could mitigate this issue, where we can start with a small upper bound of masking ratio (e.g., 15% as conventional MLM, to preserve a high proportion of observation) in the early phase of pre-training, and then gradually increase the masking ratio towards the authentic discrete diffusion objective during pre-training.

Solution in Practice. In the current form of our manuscript, adhering to this principle, we proposed a two-stage training method: initialized DPLM from an established masked LM, either from our in-house pre-trained one or from the official ESM-2 checkpoint, and then trained DPLM by the discrete diffusion language modeling objective afterwards. Though it may or may not lead to the best model performance, it offers the possibilities of standing on the shoulders of any pre-trained masked LM such as ESM2 or the advanced LM architecture such as a Llama/Mistral-style masked LMs, in the broad open-source AI community. This also enables us to bypass the time-consuming process of gradual masking warmup during pre-training. As a result, this can be the most efficient and effective approach in practice, which also shares a similar principle as finetuning from RosettaFold in RFDiffusion (Watson et al., 2023).

Although discrete diffusion pre-training from scratch is challenging, we have also explored several ways to improve it. As shown in the Fig. 2G, DPLM can achieve comparable performance through pre-training from scratch. More specifically, we found that (1) gradient norm clipping can effectively help stabilize training process of discrete diffusion language modeling, greatly reducing the chance of loss spiking and gradient nan. In addition, we also found that (2) training longer (i.e. scaling compute) is another key to attain a good ultimate model performance (trained for 300k steps).

We are also curious about the performance of the vanilla masking warmup strategy and would like to see if it can lead to a better pre-trained DPLM. The training dynamics of discrete diffusion based DPLM is an interesting and exciting direction deserving further exploration, and we leave these as our future work.

B.2 Pre-training of AR-LM baseline

We pretrain a AR-LM using autoregressive training objective. In order to be comparable with DPLM, the autoregressive language model we trained adopts the same architecture as DPLM. To be capable of adapting the autoregressive training, we modify the mask matrix of the attention module to causal mask, which guarantees each token can only attend the previous position and keep unseen for future. The training objective is next word prediction, and we process the input sequence with teacher forcing for efficient parallel training. During decoding, we start with <bos> token, sample one token each timestep from left to right, and the sampled token in the current timestep will be concatenated to the end of the sequence, becoming input for next timestep. The decoding process terminates until <eos> token is sampled. Because we can not know when to obtain the <eos> token in advance, we can not decide the length of sampled sequence. We attempt to force the sampling length by modifying the sampling probability: the probability of <eos> is 1 when and only when the sequence length is up to the predefined length, while 0 in all the previous timesteps. However, we observe this will decline the quality of sampled sequence significantly.

C Reasons for choosing absorbing discrete diffusion

We employed absorbing discrete diffusion as the pre-training method, instead of other forms of discrete diffusion or latent diffusion, for the following considerations.

C.1 Regarding discrete diffusion (DD) definitions (multinomial vs absorbing as proposed in D3PM)

Intuition. We favor absorbing DD since the learning objective of absorbing DD generalizes existing language modeling objectives, as highlighted in section 4 of Austin et al. (2021), In particular, absorbing DD is a natural extension of the masked language modeling (MLM) objective, which has been thoroughly studied (Wettig et al., 2023) in the field of NLP and widely proven to be a robust and effective sequence learning protocol. In contrast, multinomial/uniform-DD resembles (tranfitional) denoising autoencoders (Savinov et al., 2021). There is little solid evidence and remains highly unclear about (mutlinomial) denoising autoencoder as a sequence learning objective can scale up w.r.t data, model size and application scenarios.

Empirical verification. In our preliminary exploration, we have studied the performance of multinomial/uniform-DD and absorbing-DD. Here we provide the result regarding unconditional sampling, as shown in Fig. 5. We can find that DPLM-absorbing generally manifests better performance than DPLM-multinomial across different lengths.

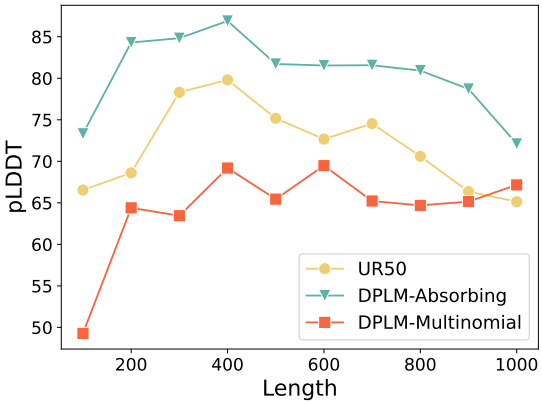

Figure 5. The unconditional sampling performance of multinomial/uniform-DD and absorbing-DD.

C.2 Regarding latent diffusion

We reason that latent diffusion for discrete sequence data, which performs Gaussian diffusion in continuous embedding space, requires additional lossy continuous relaxation of discrete sequence data like protein sequence, which does not fit the discrete nature of protein sequence and is not necessarily the best choice for modeling discrete sequence data. Recent study (Meshchaninov et al., 2024) presents a latent diffusion model on protein LM embedding, where the pLDDT score unconditional sampling attains reasonable pLDDT in their paper, while our discrete diffusion based approach still excels.

D Additional Experimental Details

D.1 A modified unconditional sampling strategy

The sampling algorithm proposed in the Zheng et al. (2023a) is to unmask positions with top-k prediction score (log-probability) predicted by $p_\theta(\mathbf{x}^{(0)}|\mathbf{x}^{(t)})$, and mask all the rest position in each denoising step. However, we find that if we use this sampling algorithm to sample sequence unconditionally, the sampled sequence will collapse to trivial pattern, such as repeating with a single amino acid. We suggest this is because, without any additional conditions, the model initially tends to give a higher prediction score to the amino acids that appear frequently in the training set. Subsequently, based on these high-frequency amino acid tokens, the model will continue to sample the same tokens beside these tokens with high confidence. The other amino acid can also be sampled, but possibly with a lower prediction score, thereby leading to be dropped according to the top-k sampling algorithm. Then, this amino acid will spread throughout the entire sequence like a virus, forming a sequence composed entirely of the same amino acids.

In response, we impose a slight disturbance during sampling, utilizing the Gumbel-Max trick. The Gumbel-Max trick is a procedure for drawing a sample from a categorical distribution using Gumbel-distributed random variables. Let's assume we have a discrete random variable X with distribution $p_\theta(\mathbf{x} = i) = p_i$ for $i = 1, \ldots, K$. Now, consider the variables $g_i = -\log(-\log U_i)$ where U_i is a variable uniformly distributed on $(0, 1]$. The g_i are random variables following a Gumbel distribution. The key to the Gumbel-Max trick is this relationship:

$$i^* = \arg\max_i \{\tilde{p}_i\}, \text{ where } \tilde{p} \propto \exp g_i + \log p_i \quad (6)$$

This operation provides a sample from the discrete distribution $p_\theta(\mathbf{x} = i)$. In other words, the category corresponding to the maximum value is the results of sampling. But in the other hand, the maximum value, *i.e.* $g_i + \log p_i$, is not equal to the original log-probability, which is actually the prediction score in our sampling algorithm. Therefore, the Gumbel-Max trick helps us sample an amino acid with a slightly modified prediction score while maintaining the original distribution. As a result, the previously dominant amino acid with the highest prediction score may be discarded, and a variety of other amino acids may be retained, thereby avoiding falling into a trivial pattern such as repeating with a single amino acid. We find that this technique can significantly reduce the number of trivial cases and further improve the diversity.

D.2 Delve deeply into the pLDDT score of unconditional sampling

According to the Fig. 2A, we surprisingly find that the pLDDT score of DPLM unconditional sampling is even higher than the UniRef50 dataset. We investigate this phenomenon as follows.

Regarding the lower pLDDT in UniRef50. The lower pLDDT here is because UniRef50 contains some data with lower structural plausibility, such as sequences with a large number of repetitive patterns. These cases decrease the average pLDDT of UniRef50, for instance sequence `ADADAD...ADADAD` with pLDDT 35.14.

We also investigate the average pLDDT of the PDB data where the pLDDT score of PDB is similar to DPLM, suggesting that DPLM learns to generate protein sequences with overall similar structural characteristics as PDB, as shown in Tab. 3.

Table 3. pLDDT score of UniRef50, PDB and DPLM unconditional sampling.

Length	UniRef50	PDB	DPLM
100	66.54	84.62	70.66
200	68.61	79.32	83.55
300	78.30	84.51	82.39
400	79.80	80.49	86.75
500	75.17	77.79	82.56
avg. pLDDT	73.68	**81.34**	81.18

Regarding mode collapse. We also want to investigate whether DPLM collapses into the modes with high pLDDT sequences. To verify this, we evaluate the pseudo-perplexity of DPLM against subsets of UR50 sequences of high pLDDT and low pLDDT. The results are shown in Tab. 4. We can find that the ppl of less-structural proteins (pLDDT < 50) is similar to the structural proteins (pLDDT > 70), suggesting that DPLM equally learns protein sequences with diverse structural patterns.

Table 4. pseudo-ppl of less structural and more structural sequences.

	less structural sequences	more structural sequences
pseudo-ppl	2.36	2.55

In conclusion, we would like to provide a possible explanation for this phenomenon. We suggest that from a perspective of probabilistic graphical model (PGM), more structured data is generally more easy to learn due to stronger correlation between its elements. As such, we hypothesize the learning dynamics of protein LMs from through evolutionary sequences is first to digest those more structural proteins as co-evolutionary effects between amino acids play a prevailing and vital role in folding patterns, and then start to learn those less structural folding patterns, which could be long-tailed.

This could somehow relate to the so-called emergence phenomenon in the realm of LLMs, where scaling up LLMs leads to "grokking" those long-tailed abilities. We would leave a study of the learning dynamics of protein LM as an exciting future investigation and hopefully can bring some interesting insights to the community.

Table 5. Results of the success rate of each problem, the number of the solved problems and the average success rate across 17 motif-scaffolding problems. Here we follow previous work to use OmegaFold as the folding model.

	seq-only		struct-cond.	
	EvoDiff	DPLM	RFDiffusion	DPLM
1bcf	0.39	**0.99**	1.00	1.00
1prw	0.87	**0.96**	0.08	**0.99**
1qjg	0.00	0.00	0.00	0.00
1ycr	0.13	**0.52**	0.74	**0.78**
2kl8	0.03	**0.05**	0.88	0.05
3ixt	0.26	0.20	0.25	**0.33**
4jhw	0.00	0.00	0.00	0.00
4zyp	0.00	**0.01**	0.40	0.01
5ius	0.00	**0.10**	0.02	**0.10**
5tpn	0.00	0.00	0.61	0.00
5trv	0.00	0.00	0.22	0.00
5wn9	0.00	**0.01**	0.00	**0.01**
5yui	0.06	**0.42**	0.00	**0.63**
6e6r	0.16	**0.84**	0.71	**0.84**
6exz	0.00	**0.01**	0.42	0.01
6vw1	0.00	0.00	0.69	0.00
7mrx	0.00	**0.59**	0.07	**0.59**
pass rate	7/17	**12/27**	13/17	12/17
avg. success rate	0.09	**0.27**	0.36	0.31

D.3 Sequence-conditional generation: motif-scaffolding

The overall motif-scaffolding results are shown in Tab. 5. We sample 100 scaffold sequences for each motif scaffolding case, and compute the success rate according to the standard mentioned in section 4.3.1. Furthermore, we also show the pass rate (e.g. the number of solved problems) and the average success rate for all problems. We use sequence-only and structure-conditioned sampling paradigms. For sequence-only sampling, DPLM generates scaffold according to the motif sequence fragment. For structure-conditioned sampling, DPLM makes generation by leveraging both sequence and structure information of motif. Specifically, as noted in section 4.3.2, we utilize the pre-trained GVPTransformerEncoder and structural adapter to process the motif structure. DPLM is able to solve 12 of 17 motif scaffolding problems. The overall success rate is 0.27 for sequence-only sampling, while 0.31 for structure-conditioned sampling. It should be noted that not all problems are suitable for using structure information. We rec-

Table 6. Motif-scaffolding results evaluated by ESMFold.

	seq-only		struct-cond.
	EvoDiff	DPLM	DPLM
1bcf	0.38	**1.00**	**1.00**
1prw	0.36	**0.75**	**0.81**
1qjg	0.00	0.00	0.00
1ycr	0.03	**0.27**	**0.48**
2kl8	0.00	**0.01**	**0.01**
3ixt	0.09	**0.15**	**0.37**
4jhw	0.00	0.00	0.00
4zyp	0.00	0.00	**0.01**
5ius	0.00	0.00	0.00
5tpn	0.00	0.00	0.00
5trv	0.00	0.00	0.00
5wn9	0.00	0.00	0.00
5yui	0.05	**0.94**	**0.94**
6e6r	0.03	**0.79**	**0.79**
6exz	0.00	**0.01**	**0.01**
6vw1	0.00	0.00	0.00
7mrx	0.00	**0.54**	**0.54**
pass rate	6/17	**9/27**	**10/17**
avg. success rate	0.06	**0.26**	**0.29**

Table 7. Ablation study on the CATH4.3 benchmark, which w/ draft means that the reverse process is based on the $\mathbf{x}_{draft}^{(t)}$.

Models	Trainable Params.	AAR	struct. eval.	
			scTM	pLDDT
LM-DESIGN (*w/* draft)	6.3M/650M	56.49	0.85	74.89
DPLM	6.3M/650M	55.75	0.83	73.72
DPLM (*w/* draft)	6.3M/650M	**56.61**	**0.86**	**76.78**

ommend using structure-conditioned sampling for 1YCR, 1PRW, 3IXT and 5YUI, while sequence-only sampling for others.

Evaluation with more advanced folding model. Moreover, we also investigate evaluation with other structure prediction models, such as ESMFold (Lin et al., 2022). Results are shown in Tab. 6, we consider the Alpha Carbon (CA) pLDDT score predicted by ESMFold as the overall pLDDT score of the amino acid. We observe that ESMFold judges more strictly than OmegaFold. When we evaluate scaffold by ESMFold, there is a slight decline in the overall pass rate and average success rate, compared with the evaluation of OmegaFold.

D.4 Structure-conditional generation: inverse folding

Model architecture. DPLM only takes amino acid tokens as input, instead of structure formats such as 3D coordinates. Therefore, in order to endow DPLM with structural awareness, we follow LM-DESIGN (Zheng et al., 2023b) and place a structural adapter after the last layer of DPLM, which can attach the structure information to the original output probability. The overall architecture of the structural adapter is constituted by three components, *i.e.*, a structure encoder, a *p*LM as sequence decoder, and a structural adapter that bridges both. We can utilize an arbitrary pretrained structure encoder to process the 3D coordinates and provide structure information for DPLM. For *p*LMs as the sequence decoder side, we primarily used the DPLM, with its pretrained model weights. The structural adapter composes a multi-head attention that queries structure information from the structure encoder, followed by a *bottleneck* feedforward network (FFN) to impose non-linearity and abstract features/representations. ROPE (Su et al., 2021) was used the supplement multi-head attention for better modeling of positional information. In all our experiments, only one structural adapter was placed after the last layer of DPLM, following Zheng et al. (2023b).

Training and inference details. During training, we freeze the parameters of the structure encoder and DPLM, only optimizing the structural adapter with the simplified discrete diffusion objective (Zheng et al., 2023a). However, we find that there is an exposure bias problem here: DPLM learns the reverse denoising process based on the ground truth context, *i.e.* $\mathbf{x}^{(t)}$, which is obtained by adding noise on the ground truth sequence, *i.e.* $\mathbf{x}^{(0)}$. During inference, DPLM has to denoise given the context predicted, which is not always right, leading to training-inference inconsistency. Therefore, we slightly modify the training objective in Eq. (4). Specifically, we obtain $\mathbf{x}^{(t)}$ by adding noise on the draft sequence generated by the pretrained structure encoder, rather than the ground truth $\mathbf{x}^{(0)}$, which we refer to as $\mathbf{x}_{draft}^{(t)}$. Then DPLM will learn the reverse process that reconstructs the $\mathbf{x}^{(0)}$ given the $\mathbf{x}_{draft}^{(t)}$, as shown in Eq. (7). Since the draft sequence is available both in training and inference time, the issue of exposure bias is mitigated. We find this technique can further boost the performance of DPLM in the inverse folding task, as illustrated in the Tab. 7

$$\mathcal{J}_t = \mathbb{E}_{q(\mathbf{x}^{(0)})}\left[\lambda^{(t)} \sum_{1\le i\le L} b_i(t)\cdot \log p_\theta(\mathbf{x}_i^{(0)}|\mathbf{x}_{\text{draft}}^{(t)})\right], \quad (7)$$

At inference time, we follow the DPLM generative process, except that we obtain protein sequence via greedy deterministic decoding, instead of random sampling from the distribution. Besides, considering that we have had an unconditional model, i.e. the DPLM itself, and a conditional model, *i.e.*, the DPLM with structural adapter, we can also seamlessly utilize the classifier-free guidance paradigm during inference.

D.5 Classifier-free guidance

Classifier-free guidance (Ho & Salimans, 2021) has been shown as an effective way to enhance conditional diffusion models. Likewise, for DPLM, we can derive an implicit

classifier using the Bayes rule

$$q(\mathbf{y}|\mathbf{x}^{(t-1)}) = q(\mathbf{y}|\mathbf{x}^{(t-1)}, \mathbf{x}^{(t)}) = \frac{q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{y})}{q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})} q(\mathbf{y}|\mathbf{x}^{(t)}).$$

If we already have an unconditional model $p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})$ and a conditional model $p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{y})$ as the estimates, then by substituting this implicit classifier into Eq. 5, we can obtain

$$\begin{aligned}\mathbf{x}^{(t-1)} &\sim p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) p_\phi(\mathbf{y}|\mathbf{x}^{(t-1)})^\eta \\ &\propto p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) \left(\frac{p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{y})}{p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})}\right)^\eta \\ &= p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{y})^\eta \cdot p_\theta(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})^{(1-\eta)},\end{aligned}$$

wherein when $\eta = 1$, it is equivalent to sampling from the original conditional DPLM without guidance, whereas $\eta > 1$, we not only prioritize the conditional model to contribute more but also discourage the samples from moving away from the unconditional distribution. In other words, it reduces the chance of generating samples that do not use conditioning information, in favor of the samples that explicitly do.

Note that when we use adapter tuning to adapt DPLM for conditional generation, we only finetune the newly-added parameters, which means that we can already access both the unconditional model (original DPLM) and conditional model (the adapter-tuned model) simultaneously for free. As demonstrated in Fig. 6 on structure-conditioned sequence generation, we can find that DPLM as a diffusion model can benefit from classifier-free guidance, improving its conditional generation immediately.

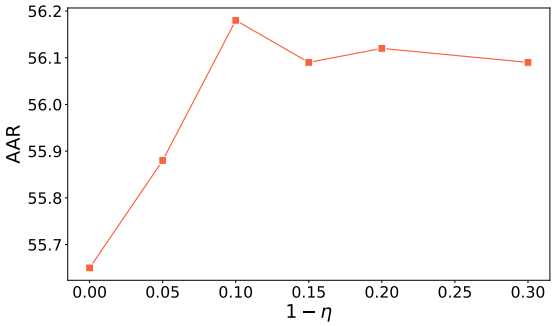

Figure 6. Classifier-free guidance enhances structure-conditioned sequence generation (inverse folding).

E Related Work

E.1 Language Models

The dominant paradigm of language models is autoregressive language models, which breaks down the mutual distribution over the tokens of a sequence into conditional probabilities via the chain rule $p(\mathbf{x}^{[1:N]}) = \prod_{i=1}^{N} p(x^{[i]}|\mathbf{x}^{[1:i-1]})$ and generates tokens by ancestral sampling from left to right (Bengio et al., 2000; Sutskever et al., 2014; Vaswani et al., 2017). Recently, researchers propose the non-autoregressive language models as an alternative (Gu et al., 2018). These models do not need to obey the left to right generation order (Qian et al., 2022; Huang et al., 2023) and demonstrate competitive or superior performance compared to their autoregressive counterpart across a wide range of domains including languages (Qian et al., 2021b; Huang et al., 2023; Qian et al., 2022; Huang et al., 2023; Zheng et al., 2023a), speeches (Kim et al., 2021), proteins (Zheng et al., 2023b), and molecules (Hoogeboom et al., 2022). Among the numerous non-autoregressive language models, diffusion language models (Li et al., 2022; Gong et al., 2022; Zheng et al., 2023a) have emerged as a solid and promising framework. Pretraining language models on a massive scale of unlabeled data markedly improves their downstream task performance (Mikolov et al., 2013; Peters et al., 2018; Radford et al., 2018; Devlin et al., 2019). As data volume and model sizes scale up, the training loss of language models predictably declines (Kaplan et al., 2020; Hoffmann et al., 2022; Muennighoff et al., 2023), and enhancing downstream task performance even without specific tuning (Radford et al., 2019). GPT3 (Brown et al., 2020) is a significant point in the journey, taking model sizes to 175B parameters, proposing in-context learning to bolster language models' competence in solving certain tasks with only a handful of demonstrations. Furthermore, Wei et al. (2021); Sanh et al. (2022); Ouyang et al. (2022) introduce instruction tuning, finetuning pretrained language models on series of tasks described via instructions, which elicits the instruction following ability of models and significantly enhances their zero-shot performance on unseen tasks. More impressively, sufficiently large language models exhibit the emergent abilities such as multi-step reasoning (Kojima et al., 2022; Wei et al., 2022a;b), which small models do not possess (Fu et al., 2023). Empowered by large language models, helpful applications such as conversational AI systems[1] and autonomous agents[2] have garnered much interest. Although the most capable models at the moment are restricted in access, open-sourced efforts (Zeng et al., 2022; Touvron et al., 2023a;b; Taori et al., 2023; Chiang et al., 2023; Sun & Qiu, 2023) have largely enhanced the public accessibility of powerful large language models.

E.2 Protein Language Models

Thanks for the abundance of 1D amino acid sequences, there is growing interest in developing protein LMs at the scale of evolution, such as the series of ESM (Rives et al., 2019; Lin et al., 2022), TAPE (Rao et al., 2019), ProtTrans (Elnaggar et al., 2021), PRoBERTa (Nambiar et al., 2020),

[1] https://chat.openai.com/

[2] https://github.com/Significant-Gravitas/Auto-GPT

PMLM (He et al., 2021), ProteinLM (Xiao et al., 2021), PLUS (Min et al., 2021), Adversarial Masked LMs (McDermott et al., 2021), ProteinBERT (Brandes et al., 2022), CARP (Yang et al., 2022a) in masked language modeling (MLM) paradigm, ProtGPT2 (Ferruz et al., 2022) in causal language modeling paradigm, and several others (Melnyk et al., 2022; Madani et al., 2021; Unsal et al., 2022; Nourani et al., 2021; Lu et al., 2020; Sturmfels et al., 2020; Strodthoff et al., 2020). These protein language models exhibit remarkable generalization ability on various downstream tasks and be able to capture evolutionary information about secondary and tertiary structures from sequences alone. Meanwhile, recent study shows these models' potency in revealing protein structures (Lin et al., 2022), predicting the effect of sequence variation on function (Meier et al., 2021), antibody infilling (Melnyk et al., 2022) and many other general purposes (Rives et al., 2019). Simultaneously, Verkuil et al. (2022) demonstrate that the large scale protein LMs can generate *de novo* proteins by generalizing beyond natural proteins, both theoretically and experimentally validating their hypothesis in exhaustive detail, in which *p*LMs demonstrate competency in designing protein structure despite being exclusively trained on sequences.

E.3 Diffusion Language Models

Derived from diffusion models (Sohl-Dickstein et al., 2015), diffusion language models is a variety of generative model that samples data via an iterative denoising process from noise. They can be divided into continuous (Ho et al., 2020; Song et al., 2020) and discrete (Hoogeboom et al., 2021b; Austin et al., 2021) categories according to the distribution they model. Continuous diffusion models make great success in vision (Dhariwal & Nichol, 2021b; Rombach et al., 2021; Ho et al., 2022), but they struggle in languages for operating on continuous surrogates of discrete tokens (Li et al., 2022; Gong et al., 2022; Han et al., 2022; Dieleman et al., 2022; Yuan et al., 2022; Gao et al., 2022a; Ye et al., 2023b; Chen et al., 2023a; Wu et al., 2023), which has difficulty bypassing the pitfall of discreteness (Ye et al., 2023b) and still lags behind autoregressive language models. In contrast, discrete diffusion models, albeit having limited progress in large-scale applications, are innately suited to the data type inherent to languages (*i.e.*, sequences of discrete tokens). Zheng et al. (2023a) makes commendable strides in discrete diffusion models and enhancing these models to yield comparable performance with autoregressive models on typical language generation benchmarks like machine translation. Furthermore, as shown by He et al. (2023); Zheng et al. (2023b), there are close relationship between discrete diffusion models and masked language models (MLM), a widely adopted pretraining paradigm in NLP (Devlin et al., 2019; Liu et al., 2019). Following this line, Ye et al. (2023a) propose scaling discrete diffusion LMs with diffusive adaptation, showing strong performance on several conditional text generation tasks, and accessing zero-shot instruction following, few-shot in-context learning and the promise of structured reasoning with instruction tuning.

E.4 Protein Structure Diffusion Models

Diffusion models have become popular tools in structural biology for protein generation, and their utility has been demonstrated across a range of generative tasks in recent years. Trippe et al. (2022), along with others, have introduced several diffusion model variants, each with its unique approach. For instance, while some models focus on generating the protein backbone by diffusing over protein coordinates, others, such as those proposed by Wu et al. (2022b), target inter-residue angles. Lin & AlQuraishi (2023) and Yim et al. (2023) have developed models that handle both the position and orientation of residue frames. RFDiffusion (Watson et al., 2023) is a model that assists in designing protein structures for specific functions, such as enzymes. It is versatile in protein design and has been used to create therapeutic proteins, with some designs being confirmed in the laboratory. ProteinSGM (Lee et al., 2022) is a model that uses 2D matrices, which represent the distances and angles between protein parts, to create 3D protein structures for novel protein designs. FoldingDiff (Wu et al., 2022a) is a model that generates protein sequences expected to fold into a specific structure. These sequences are verified with prediction tools, although they have not been experimentally confirmed yet. Chroma (Ingraham et al., 2023) is a model designed for creating large proteins and protein complexes, considering various constraints like distances and symmetry. It transforms a collapsed polymer into protein backbone and sequence more quickly than older methods, thereby allowing for the efficient generation of large structures.

E.5 Protein Inverse Folding

The structure-based protein sequence design is typically formulated as a conditional sequence generation problem by deep generative modeling, wherein protein 3D structures are usually depicted as a k-NN graph (Ingraham et al., 2019). The protein graph establishes edge features between adjacent residues and encodes residue information as node features, modeled by graph neural networks (GNNs). GraphTrans (Ingraham et al., 2019) and GVP (Jing et al., 2020) utilizes the graph attention encoder and autoregressive decoder for protein design. Recently, ProteinMPNN (Dauparas et al., 2022) and PiFold (Gao et al., 2022b) introduce more complex protein features and expressive GNNs, resulting in significant improvements. Furthermore, in addition to the primary generative purpose, this task can also be used as a proxy for protein (structure-aware) representation learning (Yang et al., 2022b). A critical and significant challenge herein is the lack of sufficient protein structure data. To this end, ESM-IF (Hsu et al., 2022) alleviate this issue with effective data augmentation by back-translation with AlphaFold

2 (Jumper et al., 2021). , resulting in dramatic improvements. On the other hand, Zheng et al. (2023b) demonstrate how to efficiently steering large pretrained protein LMs into a structure-informed sequence generative models in a mask-predict generative manner, attaining state-of-the-art results on single-chain and complex protein benchmark. Most recently, graph diffusion models have also been studied for inverse folding problem (Yi et al., 2023).

F Visualization of Unconditional Samples

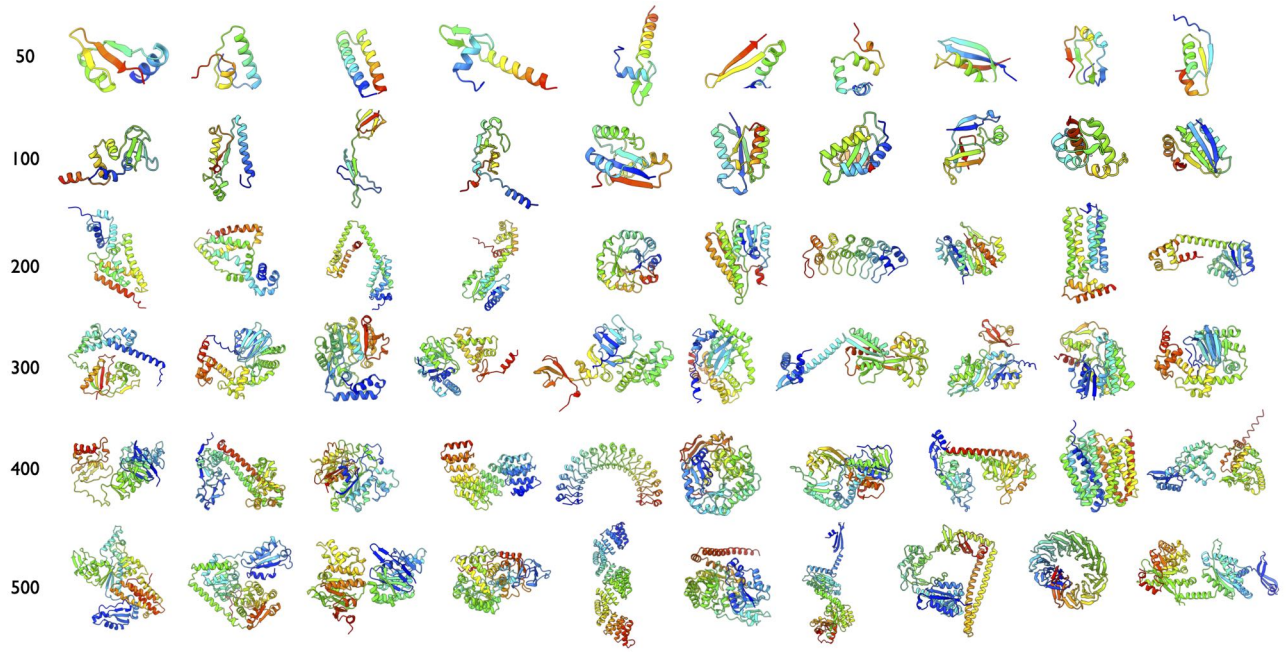

Figure 7. Visualized examples from 50 to 500 in length.

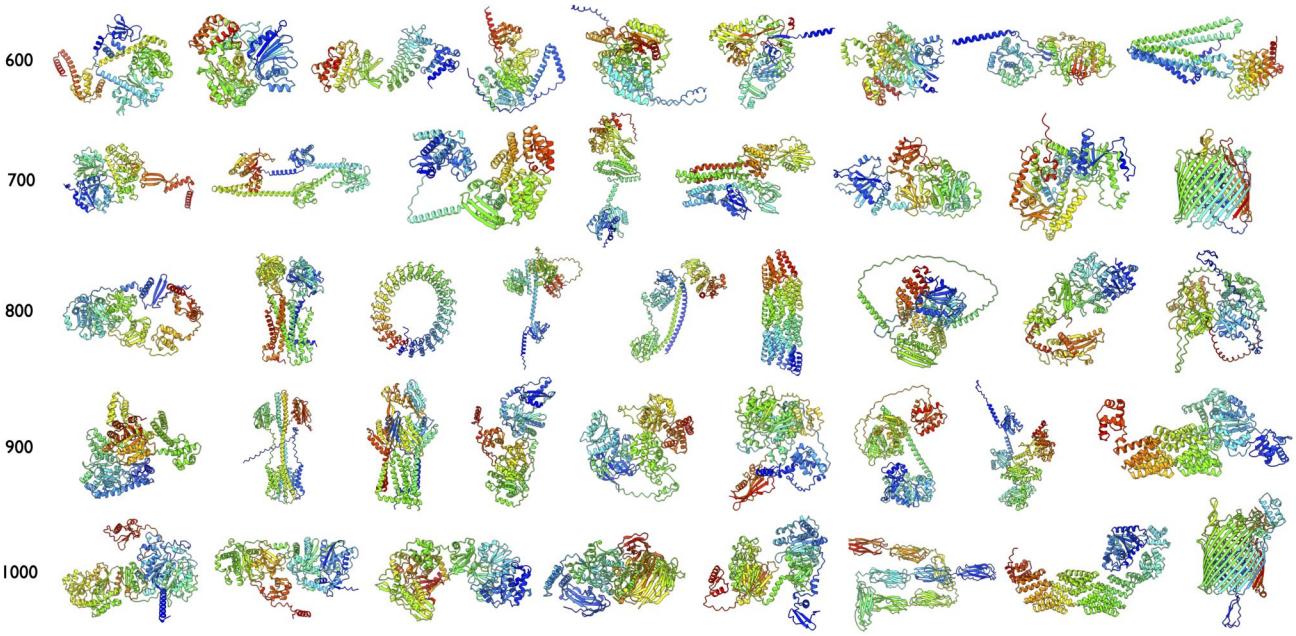

Figure 8. Visualized examples from 600 to 1000 in length.