SHORTLISTING MODEL: A STREAMLINED SIMPLEX DIFFUSION FOR BIOLOGICAL SEQUENCE GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative modeling of discrete variables is challenging yet crucial for applications in natural language processing and biological sequence design. We introduce the Shortlisting Model (SLM), a novel simplex-based diffusion model inspired by progressive candidate pruning. SLM operates on simplex centroids, reducing complexity and enhancing scalability. Additionally, SLM incorporates a flexible implementation of classifier-free guidance, enhancing unconditional generation performance. Extensive experiments in DNA promoter and enhancer design, and protein design demonstrates SLM's competitive performance and scalability.

1 INTRODUCTION

Although autoregressive models, such as large language models (LLMs), have achieved remarkable success in text generation (Achiam et al., 2023; Brown et al., 2020), they struggle with applications lacking an intrinsic sequential-ordering inductive bias, including DNA design (Avdeyev et al., 2023; Stark et al., 2024), protein sequence design (Wang et al., 2024; Lin et al., 2023), and molecular graph generation (Vignac et al., 2022). Consequently, there is growing interest in developing new discrete generative paradigms, such as diffusion-based (Lou et al.; Sahoo et al., 2024; Shi et al., 2024) and flow-matching-based (Gat et al., 2024; Davis et al., 2024; Cheng et al., 2024) methods. This trend underscores the motivation to explore novel paradigms and models for discrete variable generation.

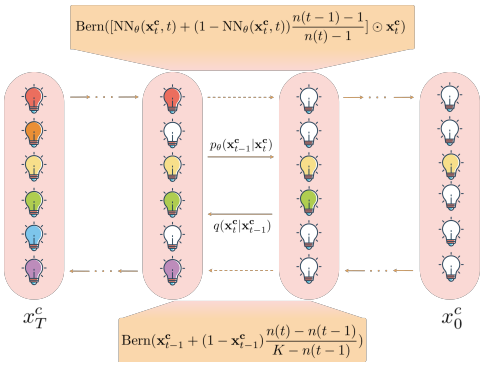

Figure 1: SLM's forward and reverse process. A comparison between MDLM and DP3M-Uniform is located in Appendix.B.1

Recent discrete generative models are generally classified into two main categories based on their operational spaces: discrete-space models and continuous-space models. The discrete-space models, specifically discrete diffusion models (Lou et al.; Xu et al., 2024), which mimic continuous diffusion processes using substitution or masking operations to decompose information, have shown impressive performance for discrete generative modeling. However, these discrete counterparts differ fundamentally from original continuous diffusion models (Ho et al., 2020), where the smooth and gradual information transitions intrinsic to the generation process are key to their success; On the other hand, continuous space models map discrete data into continuous representations allowing for the navigation of the success over continuous generative models. While these models benefit from continuous properties, capturing the geometric structure and adhering to constraints introduces new challenges. In this context of continuous-space models, simplex-based approaches (Cheng et al., 2024; Davis et al., 2024; Graves et al., 2023) offer a balanced solution by representing discrete data on the probability simplex, which naturally adheres to the fundamental properties of categorical distributions and have demonstrated impressive performance.

However, existing simplex-based approaches often rely on intricate operations to define trajectories over the entire continuous space. For instance, Statistical Flow Matching(SFM) (Cheng et al., 2024)

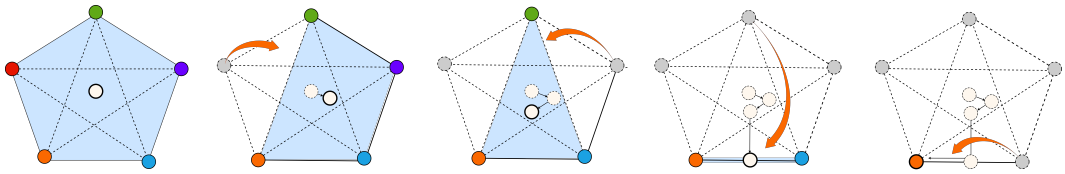

Figure 2: Pathological behavior of SLM on one simplex with K = 5(Δ^4). Each vertex represents one of the categorical targets while the trajectory of the white point serves as a probability path in sampling. Note that the trajectory of shortlisting model could be seen as jumping among the centroids of subspaces in simplex space.

as well as Fisher Flow Matching (Davis et al., 2024) define geodesics based on sphere map and the Fisher-Rao metric. The training also incorporates Riemannian optimal transport; Similarly, Bayesian Flow Networks (BFNs) involve heavy mathematical derivations and change-of-variable techniques to define simplex trajectories through Gaussian-formed count variables. While these methods are mathematically rigorous, their complexity can pose challenges to scalability and practical implementation in large-scale generative tasks.

In this paper, we aim to preserve the key principle of simplex-based approaches, *i.e.*, information should grow gradually and smoothly throughout the generation process, while exploring simpler yet effective alternatives. To this end, we proposed interpreting the generation of discrete variables as a progressive candidate pruning process, where the candidate set starts with all possible categories and is gradually refined to a single category. We therefore refer to such model as **shortlisting models**. Formally, the shortlist models lie in the scope of diffusion models which enables the training with the variational lower bounds(vlb VLB) of likelihood. Unlike other simplex-based approaches, we demonstrate that the shortlisting model can be trained using a simplified Cross Entropy loss which also effectively mitigates the vanishing gradient issue of the original objective, as opposed to the vocabulary-level MSE loss (Graves et al., 2023; Cheng et al., 2024; Davis et al., 2024), offering greater potential in large vocabulary settings. Besides, the proposed model could be seen as transforming in the **centroids** of the simplex instead of the full space as shown in Figure.1 and Figure.2 which minimizes the degrees of freedom. Furthermore, we demonstrate that the shortlisting models take a flexible formulation for further adaptation such as classifier-free guidance.

We comprehensively evaluate the proposed approach over various biological sequence design tasks and benchmarks, such as DNA promoter and enhancer design, as well as protein sequence design. In DNA design tasks, our non-guided variants achieve results comparable to previous guided methods. Furthermore, with classifier-free guidance, our model attains SOTA performance while being less sensitive to hyperparameters; Additionally, our 38M-parameter shortlisting model could design proteins with enhanced foldability, fitness, self-consistency, and diversity, surpassing the performance of the larger, well-known ESM2-150M model (Lin et al., 2022).

2 Preliminary

2.1 Definition and Notations

We encode a discrete variable with K distinct categories using one-hot vectors $\mathbf{e} = [e_1, e_2, \ldots, e_K]^T$. In each vector, only the i-th element $\mathbf{e}(i) = 1$ signifies the inclusion of the i-th category, while all other elements are zero. We define:

Definition 2.1. A **candidate set** for K categories is defined as a binary-valued vector $\mathbf{c} = [c_1, c_2, \ldots, c_K]^T$, where each $c_i \in \{0, 1\}$, and the vector has at least one non-zero entry, *i.e.,* $\mathbf{1}^T\mathbf{c} > 0$.

The candidate set variable $\mathbf{c}$ encodes the selection status of each category: $\mathbf{c}(i) = 1$ indicates that the i-th category is included, while $\mathbf{c}(i) = 0$ denotes its exclusion. Specifically, there are two distinct instances of the candidate variable: $\mathbf{c}$ is an all one's vector ($[1, \cdots, 1]$) which represents maximum candidates setting, *i.e.* all K categories are selected; one-hot vector is another special case which represents minimum candidates setting, here there is only one category included.

2.2 DIFFUSION MODELS

As the shortlisting model is a variant of diffusion models, we briefly introduce the necessary components of the corresponding latent variable models here. Diffusion models can be viewed as latent variable models in which the latent variables form a Markov chain. Specifically, for a diffusion model with the sequence latent variable $x_{1:T} = x_1, \cdots x_T$, the implied density function p_θ holds the following Markovness by definition: $p_\theta(x_0, x_{1:T}) = p_\theta(x_0|x_1)p_\theta(x_1|x_2)\cdots p_\theta(x_{T-1}|x_T)$ To learn this latent variable model, a carefully designed constant variational distribution $q\left(\boldsymbol{x}_{1:T} \mid \boldsymbol{x}_0\right) = \prod_{t=1}^{T} q\left(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}\right)$, also referred to as the forward process, is involved. Based on the variational distribution, the diffusion model is generally trained with the following variational lower bound (Austin et al., 2021; Ho et al., 2020):

$$L_{\text{vlb}} = \mathbb{E}_{q(\boldsymbol{x}_0)}[\underbrace{D_{\text{KL}}\left[q\left(\boldsymbol{x}_T \mid \boldsymbol{x}_0\right) \| p\left(\boldsymbol{x}_T\right)\right]}_{L_T} + \sum_{t=2}^{T} \underbrace{\mathbb{E}_{q(\boldsymbol{x}_t|\boldsymbol{x}_0)}\left[D_{\text{KL}}\left[q\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0\right) \| p_\theta\left(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t\right)\right]\right]}_{L_{t-1}}$$
$$\underbrace{-\mathbb{E}_{q(\boldsymbol{x}_1|\boldsymbol{x}_0)}\left[\log p_\theta\left(\boldsymbol{x}_0 \mid \boldsymbol{x}_1\right)\right]}_{L_0}]. \tag{1}$$

Here $q(x_0)$ refers to the data distribution.

3 SHORTLISTING MODELS

The shortlisting model is inspired by the idea of treating the generation process of discrete variables as a category selection process. It begins by considering all categories in the vocabulary as potential candidates and progressively narrows down the options until reaching a final one-hot representation, indicating a single category. We introduce the detailed component of the shortlisting model in the following.

3.1 FORWARD CANDIDATE APPENDING PROCESS

To transfer the above insight into modeling, we design a forward candidate appending process over the space of **candidate set** as introduced in Definition. 2.1: For any discrete variable $\mathbf{x}$, its one-hot representation (as a special case of candidate set) is considered as the initial step and denoted as $\mathbf{x}_0^{\mathbf{c}}$; For the last step, $\mathbf{x}_T^{\mathbf{c}}$, we make it as the all 1s vector ($\mathbf{1}$). Then we seek the following Markov chain interpolation $\mathbf{x}_0^{\mathbf{c}} - \mathbf{x}_1^{\mathbf{c}} - \cdots - \mathbf{x}_T^{\mathbf{c}}$ between the $\mathbf{x}_0^{\mathbf{c}}$ and $\mathbf{x}_T^{\mathbf{c}}$, which satisfies that:

$$\forall_{0\le i<j\le T}\ \ \mathbf{1}^T\mathbf{x}_i^{\mathbf{c}} \le \mathbf{1}^T\mathbf{x}_j^{\mathbf{c}}, [\mathbf{x}_j^{\mathbf{c}}]^T\mathbf{x}_i^{\mathbf{c}} = \mathbf{1}^T\mathbf{x}_i^{\mathbf{c}} \tag{2}$$

Recall the $\mathbf{x}^{\mathbf{c}}$ is binary valued vector, hence the above condition essentially indicates the possible categories implied by the candidate set of early steps are **strictly scooped** by the later steps. We propose using a multivariate Bernoulli distribution to model the forward process over the candidate set variable, denoted as $\mathbf{x}^{\mathbf{c}} \sim \text{Bern}(\boldsymbol{\phi})$, where ϕ is a K dimensional vector representing the parameters of the distribution.To control the noise level, we introduce $n(t)$ as a scheduling function over the candidate numbers, where $1 \le n(t) \le K$. By our definition, $n(t)$ is a monotonically increasing function from time step 0 to time step T, designed to gradually perturb the signal. Intuitively, $n(t)$ can be seen as controlling the number of ones in the vector $\mathbf{x}_t^{\mathbf{c}}$, representing the number of possible categories at time t. To satisfy the condition in Eq. 2, we set $n(0) = 1$ and $n(T) = K$, and define the transition probabilities from $t-1$ to t as:

$$q(\mathbf{x}_t^{\mathbf{c}}|\mathbf{x}_{t-1}^{\mathbf{c}}) = \text{Bern}(\mathbf{x}_{t-1}^{\mathbf{c}} + (1 - \mathbf{x}_{t-1}^{\mathbf{c}})\frac{n(t) - n(t-1)}{K - n(t-1)}) \tag{3}$$

Proposition 3.1. *With Eq. 3 as the transition probability, the marginal distribution is as:*

$$q(\mathbf{x}_t^{\mathbf{c}}|\mathbf{x}_0^{\mathbf{c}}) = \text{Bern}(\mathbf{x}_{t-1}^{\mathbf{c}} + (1 - \mathbf{x}_{t-1}^{\mathbf{c}})\frac{n(t) - 1}{K - 1}) \tag{4}$$

and corresponding posterior distribution $q(\mathbf{x}_{t-1}^{\mathbf{c}}|\mathbf{x}_t^{\mathbf{c}}, \mathbf{x}_0^{\mathbf{c}})$ also lies in the form of Bernoulli distribution and the analytical form is ($t \ge 2$):

$$q(\mathbf{x}_{t-1}^{\mathbf{c}}|\mathbf{x}_t^{\mathbf{c}}, \mathbf{x}_0^{\mathbf{c}}) = \text{Bern}(\mathbf{x}_0^{\mathbf{c}} + [(1 - \mathbf{x}_0^{\mathbf{c}}) \odot \mathbf{x}_t^{\mathbf{c}}]\frac{n(t-1) - 1}{n(t) - 1}) \tag{5}$$

Here $\odot$ stands for the Hadamard products between two vectors.

We leave the detailed proof in the Appendix.A

3.2 Reverse Candidate Pruning Process

The reverse process implied by $p_\theta(\mathbf{x}^\mathbf{c}_{t-1}|\mathbf{x}^\mathbf{c}_t)$ corresponds to the progressive candidate pruning process. We follow previous literature (Austin et al., 2021; Sahoo et al., 2024) to parameterize $p_\theta(\mathbf{x}^\mathbf{c}_{t-1}|\mathbf{x}^\mathbf{c}_t)$ by combining a neural network(θ) predicted $\mathbf{x}^\mathbf{c}_0$ based on $\mathbf{x}^\mathbf{c}_t$ and the formulation of the posterior in Eq. 5:

$$\begin{aligned} p_\theta(\mathbf{x}^\mathbf{c}_{t-1}|\mathbf{x}^\mathbf{c}_t) &= q(\mathbf{x}^\mathbf{c}_{t-1}|\mathbf{x}^\mathbf{c}_t, \mathrm{NN}_\theta(\mathbf{x}^\mathbf{c}_t, t)) \\ &= \mathrm{Bern}([\mathrm{NN}_\theta(\mathbf{x}^\mathbf{c}_t, t) + (1 - \mathrm{NN}_\theta(\mathbf{x}^\mathbf{c}_t, t))\frac{n(t-1)-1}{n(t)-1}] \odot \mathbf{x}^\mathbf{c}_t) \end{aligned} \tag{6}$$

Here $\mathrm{NN}_\theta(\mathbf{x}^\mathbf{c}_t, t)$ refers to a probability distribution over the K-dim, *e.g.*, outputs after the $\mathrm{softmax}$. Each parameter in $p_\theta\left(\mathbf{x}^\mathbf{c}_{t-1} \mid \mathbf{x}^\mathbf{c}_t\right)$ can be viewed as an interpolation between the constant value $\frac{n(t-1)-1}{n(t)-1}$ and 1, weighted by NN_θ.

Moreover, we propose incorporating the property of the forward process where $\mathbf{x}^\mathbf{c}_{t-1}$ is strictly contained within $\mathbf{x}^\mathbf{c}_t$, expressed as $[\mathbf{x}^\mathbf{c}_t]^T \mathbf{x}^\mathbf{c}_{t-1} = \mathbf{1}^T \mathbf{x}^\mathbf{c}_{t-1}$. This property is integrated into the parameterization by ensuring that $\mathrm{NN}_\theta\left(\mathbf{x}^\mathbf{c}_t, t\right)$ has non-zero values only for categories within $\mathbf{x}^\mathbf{c}_t$, satisfying $[\mathrm{NN}_\theta\left(\mathbf{x}^\mathbf{c}_t, t\right)]^T\left(\mathbf{1} - \mathbf{x}^\mathbf{c}_t\right) = 0$. Practically, such condition could be satisfied by add $-\infty$ to the logits before the $\mathrm{softmax}$ operation. The prior distribution is set as the all ones vector, *i.e.*, $p_\theta(\mathbf{x}^\mathbf{c}_T) = \mathrm{Bern}(\mathbf{1})$.

3.3 Training of Shortlisting Models

Here we present the training of the shortlisting models. We put the formulation in Eq. 4, Eq. 5 and Eq. 6 into the Variational lower bound in Eq. 1 to derive the final objective for shortlisting models. We start with the first term L_T. As mentioned in above section, $p_\theta(\mathbf{x}^\mathbf{c}_T) = \mathrm{Bern}(\mathbf{1})$; And we put the time step T into Eq. 4,

$$q(\mathbf{x}^\mathbf{c}_T|\mathbf{x}^\mathbf{c}_0) = \mathrm{Bern}(\mathbf{x}^\mathbf{c}_0 + (1 - \mathbf{x}^\mathbf{c}_0)\frac{n(T)-1}{K-1}) = \mathrm{Bern}(\mathbf{x}^\mathbf{c}_0 + (1 - \mathbf{x}^\mathbf{c}_0)\frac{K-1}{K-1}) = \mathrm{Bern}(\mathbf{1})$$

The first term L_T in Eq. 1 is as: $L_T = \mathbb{E}_{q\left(\boldsymbol{x}^\mathbf{c}_0\right)} D_{\mathrm{KL}}\left[\mathrm{Bern}(\mathbf{1}) \| \mathrm{Bern}(\mathbf{1})\right] = 0$. For the last term L_0, with $n(0) = 1$ the $p_\theta(\mathbf{x}^\mathbf{c}_0|\mathbf{x}^\mathbf{c}_1)$ in Eq. 6 could be expressed as $p_\theta(\mathbf{x}^\mathbf{c}_0|\mathbf{x}^\mathbf{c}_1) = \mathrm{Bern}(\mathrm{NN}_\theta(\mathbf{x}^\mathbf{c}_t, t))$. Then L_0 could be expressed as:

$$\begin{aligned} L_0 &= -\mathbb{E}_{q\left(\boldsymbol{x}^\mathbf{c}_1|\boldsymbol{x}^\mathbf{c}_0\right)}\left[\log p_\theta\left(\boldsymbol{x}^\mathbf{c}_0 \mid \boldsymbol{x}^\mathbf{c}_1\right)\right] \\ &= -\mathbb{E}_{q\left(\boldsymbol{x}^\mathbf{c}_1|\boldsymbol{x}^\mathbf{c}_0\right)}[\log \langle \mathrm{NN}_\theta(\mathbf{x}^\mathbf{c}_1, t), \boldsymbol{x}^\mathbf{c}_0\rangle + \begin{cases} \log \langle 1 - \mathrm{NN}_\theta(\mathbf{x}^\mathbf{c}_1, t), \boldsymbol{x}^\mathbf{c}_1 - \boldsymbol{x}^\mathbf{c}_0\rangle, & \|\mathbf{x}^\mathbf{c}_1 - \mathbf{x}^\mathbf{c}_0\| > 0 \\ 0, & \|\mathbf{x}^\mathbf{c}_1 - \mathbf{x}^\mathbf{c}_0\| = 0 \end{cases} \end{aligned}$$

Here $\langle\cdot\rangle$ denotes the inner product. Then we focus over the term of L_{t-1}, for simplicity we use $\mathrm{pred}_\theta(\mathbf{x}^\mathbf{c}_t)$ as shorted notation for $[\mathrm{NN}_\theta(\mathbf{x}^\mathbf{c}_t, t) + (1 - \mathrm{NN}_\theta(\mathbf{x}^\mathbf{c}_t, t))\frac{n(t-1)-1}{n(t)-1}] \odot \mathbf{x}^\mathbf{c}_t)$, correspondingly, $\mathrm{gd}(\mathbf{x}^\mathbf{c}_t)$ for $\mathbf{x}^\mathbf{c}_0 + [(1 - \mathbf{x}^\mathbf{c}_0) \odot \mathbf{x}^\mathbf{c}_t]\frac{n(t-1)-1}{n(t)-1}$.

$$L_{t-1} = \mathbb{E}_{q\left(\boldsymbol{x}^\mathbf{c}_t|\boldsymbol{x}^\mathbf{c}_0\right)}\left[D_{\mathrm{KL}}\left[\mathrm{Bern}(\mathrm{gd}(\mathbf{x}^\mathbf{c}_t)) \| \mathrm{Bern}(\mathrm{pred}_\theta(\mathbf{x}^\mathbf{c}_t))\right]\right] \tag{7}$$

The KL divergence between the Multivariate Bernoulli distribution could be extended as:

$$\begin{aligned} &D_{\mathrm{KL}}[\mathrm{Bern}(\mathrm{gd}(\mathbf{x}^\mathbf{c}_t)) \| \mathrm{Bern}(\mathrm{pred}_\theta(\mathbf{x}^\mathbf{c}_t))] \\ &= \sum_{i, \mathbf{x}^\mathbf{c}_t(i) > 0} (\mathrm{gd}(\mathbf{x}^\mathbf{c}_t)(i) \log \frac{\mathrm{gd}(\mathbf{x}^\mathbf{c}_t)(i)}{\mathrm{pred}_\theta(\mathbf{x}^\mathbf{c}_t)(i)} + (1 - \mathrm{gd}(\mathbf{x}^\mathbf{c}_t)(i)) \log \frac{1 - \mathrm{gd}(\mathbf{x}^\mathbf{c}_t)(i)}{1 - \mathrm{pred}_\theta(\mathbf{x}^\mathbf{c}_t)(i)}) \end{aligned} \tag{8}$$

3.3.1 Mitigating the Gradient Vanishing

However, we observe that directly optimizing the KL divergence of a multi-dimensional Bernoulli distribution, as in Eq. 7, can lead to optimization failure, with the process getting stuck from the

beginning. This issue is likely due to gradient vanishing, where the gradients become too small to drive effective parameter updates. We formally investigate the issue in the following. Taking dimension i in the K dimensions and $\mathbf{x}_t^{\mathbf{c}}(i) > 0$ as an example, the gradient towards the parameter θ:

$$\nabla_\theta D_{\mathrm{KL}}[\mathrm{Bern}(\mathrm{gd}(\mathbf{x}_t^{\mathbf{c}})(i)) \| \mathrm{Bern}(\mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i))] = -(\frac{\mathrm{gd}(\mathbf{x}_t^{\mathbf{c}})(i)}{\mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i)} - \frac{1-\mathrm{gd}(\mathbf{x}_t^{\mathbf{c}})(i)}{1-\mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i)})\nabla_\theta \mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i) \tag{9}$$

Recall that $\mathrm{gd}(\mathbf{x}_t^{\mathbf{c}})(i)$ and $\mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i)$ are both interpolations between 1 and $\frac{n(t-1)-1}{n(t)-1}$ as discussed in Section 3.2, we have that $\frac{n(t-1)-1}{n(t)-1} \leq \mathrm{gd}(\mathbf{x}_t^{\mathbf{c}})(i), \mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i) \leq 1$. Consider the situation when $\mathbf{x}_0^{\mathbf{c}}(i) = 1$, while the network work prediction $\mathrm{NN}_\theta(\mathbf{x}_1^{\mathbf{c}}, t)(i)$ hold a very small value. Here the norm of the weight: $\| \frac{\mathrm{gd}(\mathbf{x}_t^{\mathbf{c}})(i)}{\mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i)} - \frac{1-\mathrm{gd}(\mathbf{x}_t^{\mathbf{c}})(i)}{1-\mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i)} \|_2 \leq \frac{n(t)-1}{n(t-1)-1}$. Then we consider the term of $\nabla_\theta \mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i)$: $\nabla_\theta \mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i) = \frac{n(t)-n(t-1)}{n(t)-1} \nabla_\theta \mathrm{NN}_\theta(\mathbf{x}_t^{\mathbf{c}}, t)(i)$ Then we have followed observation towards the gradient norm of Eq. 9 as:

$$\begin{aligned}
&\|\nabla_\theta D_{\mathrm{KL}}[\mathrm{Bern}(\mathrm{gd}(\mathbf{x}_t^{\mathbf{c}})(i)) \| \mathrm{Bern}(\mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i))]\|_2 \\
&= \| \frac{\mathrm{gd}(\mathbf{x}_t^{\mathbf{c}})(i)}{\mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i)} - \frac{1-\mathrm{gd}(\mathbf{x}_t^{\mathbf{c}})(i)}{1-\mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i)} \|_2 \|\mathrm{pred}_\theta(\mathbf{x}_t^{\mathbf{c}})(i)\|_2 \\
&\leq \frac{n(t)-1}{n(t-1)-1} \frac{n(t)-n(t-1)}{n(t)-1} \|\nabla_\theta \mathrm{NN}_\theta(\mathbf{x}_t^{\mathbf{c}}, t)(i)\|_2
\end{aligned} \tag{10}$$

Even when NN_θ entirely mispredicts the ground truth category, the penalized weight remains bounded by $\frac{n(t)-n(t-1)}{n(t-1)-1}$. This bound can be relatively small when $n(t-1)$ is large and the increment $n(t) - n(t-1)$ is modest. Such conditions are common in practical tasks involving large vocabulary sizes (K) and numerous timesteps (T). To mitigate such issues, we propose to address the weighted term in the gradient(Eq. 9) which results in a simplified objective as:

$$L_{t-1}^{\text{simple}} = -\mathbb{E}_{q(\boldsymbol{x}_t^{\mathbf{c}} | \boldsymbol{x}_0^{\mathbf{c}})} [\langle \log \mathrm{NN}_\theta(\boldsymbol{x}_t^{\mathbf{c}}, t), \boldsymbol{x}_0^{\mathbf{c}} \rangle] \tag{11}$$

The above objective is essentially the Cross-Entropy loss between the network prediction and the original data sample. Compared with the original objective in Eq. 7, the simplified objective has gradient over i-dim as $-\frac{\nabla_\theta \mathrm{NN}_\theta(\boldsymbol{x}_t^{\mathbf{c}}, t)(i)}{\mathrm{NN}_\theta(\boldsymbol{x}_t^{\mathbf{c}}, t)(i)}$. This ensures that the aforementioned misprediction is adequately penalized. By default, the shortlisting model is trained using Eq. 11. However, a slight discrepancy exists with the variation bounds in Eq. 7. To bridge this gap, we introduce a weighted function aimed at enhancing density estimation performance for specific tasks:

$$L_{t-1}^{\text{weight}} = -\mathbb{E}_{q(\boldsymbol{x}_t^{\mathbf{c}} | \boldsymbol{x}_0^{\mathbf{c}})} \left[\frac{n(t)-n(t-1)}{n(t)-1} \langle \log \mathrm{NN}_\theta(\boldsymbol{x}_t^{\mathbf{c}}, t), \boldsymbol{x}_0^{\mathbf{c}} \rangle \right] \tag{12}$$

3.3.2 Candidate Set Size Scheduling

Another important component of the framework is the scheduling function over the candidate set size, *i.e.* $n(t)$. It is noteworthy that $n(t)$ is not restricted to integer values; rather, it can take any real value in the interval $[1, K]$. We take a similar intuition from (Graves et al., 2023), by considering the normalized vector $\frac{\mathbf{x}_t^{\mathbf{c}}}{\sum_{i=1}^K \mathbf{x}_t^{\mathbf{c}}(i)}$ as the probability of distribution over vocabulary, then we expected the entropy of the distribution increase linearly from $t = 1$ to $t = T$. Note the expected ones of $\mathbf{x}_t^{\mathbf{c}}$ is exactly $n(t)$, and hence the corresponding entropy of the aforementioned distribution is $\log n(t)$. Then we could design scheduling function as:

$$n(t) = e^{(\log K) \frac{t}{T}} \tag{13}$$

4 Sampling of Shortlisting Models

The sampling process of shortlisting models could be directly conducted with ancestral sampling based on the learned $p_\theta(\mathbf{x}_{t-1}^{\mathbf{c}} | \mathbf{x}_t^{\mathbf{c}})$ with $\mathbf{x}_T^{\mathbf{c}} \sim \mathrm{Bern}(\mathbf{1})$ as the starting point. We provide the full pseudocode for training and sampling in Appendix.B.2.

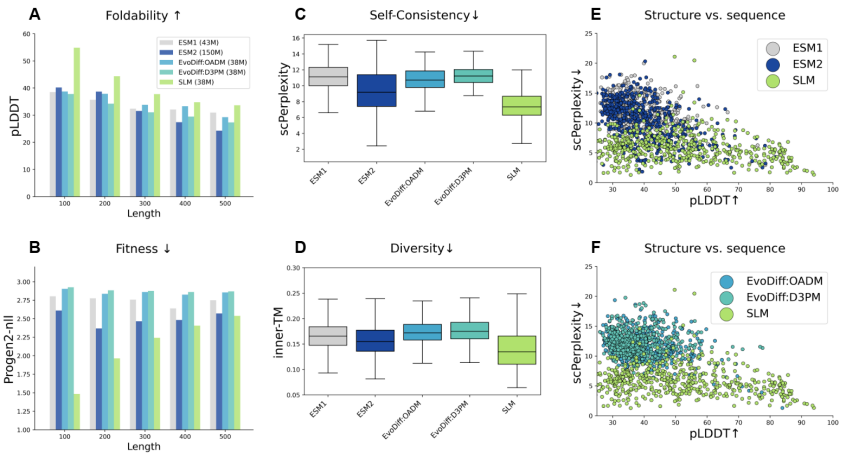

Figure 3: **Quantitative Performance on Protein Sequence Design SLM compared to baselines**: (A-D) pLDDT(A), Progen2-nll(B), scPerplexity(C) and inner-TM(D) scores for sequence sampled from ESM1-43M, ESM2-150M, EvoDiff-OADM-38M, EvoDiff-D3PM-38M and SLM-38M models. (E-F) The joint distribution of pLDDT and scPerplexity from SLM model and Masked Language Models(E) and Diffusion Models(F).

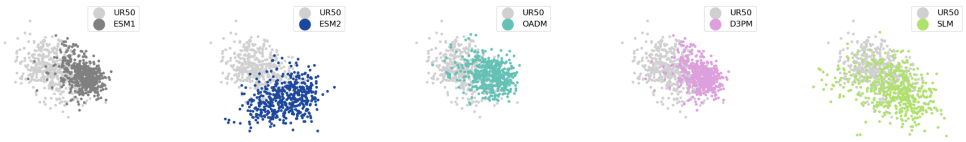

Figure 4: SLM not only fits the reference distribution well but also explores a broader outer area under ProstT5 embedding.

4.1 Classifier-free Guidance

We show that the simple formulation offers the advantage of flexibility to implement the classifier-free guidance with an extra class-conditioned shortlist model. Denoting the output of unconditional model at timestep t as $\mathrm{NN}_\theta(\mathbf{x}_t^{\mathbf{c}}, t, K)$ and the conditional model is as $\mathrm{NN}_\theta(\mathbf{x}_t^{\mathbf{c}}, t, \mathrm{cls})$. Here $\mathrm{cls} \in [0, K-1] \cap \mathbb{Z}$ denotes the class label. The reverse process based on classfier-free guidance could be obtained as:

$$p_\theta^{\mathrm{CFG}}(\mathbf{x}_{t-1}^{\mathbf{c}} | \mathbf{x}_t^{\mathbf{c}}) = \mathrm{Bern}([\hat{\mathrm{NN}}_\theta(\mathbf{x}_t^{\mathbf{c}}, t) + (1 - \hat{\mathrm{NN}}_\theta(\mathbf{x}_t^{\mathbf{c}}, t))\frac{n(t-1)-1}{n(t)-1}] \odot \mathbf{x}_t^{\mathbf{c}})$$

Here the $\hat{\mathrm{NN}}_\theta$ is as: $\hat{\mathrm{NN}}_\theta(\mathbf{x}_t^{\mathbf{c}}, t) = \gamma \mathrm{NN}_\theta(\mathbf{x}_t^{\mathbf{c}}, t, \mathrm{cls}) + (1-\gamma)\mathrm{NN}_\theta(\mathbf{x}_t^{\mathbf{c}}, t, K)$. When $\gamma > 1$, there could be negative number in $\hat{\mathrm{NN}}_\theta$. Following (Stark et al., 2024), we project the value of $\hat{\mathrm{NN}}_\theta$ back to the simplex based on (Wang & Carreira-Perpinán, 2013).

5 Experiments

5.1 De novo Design of Protein Sequence

In this section, we focus primarily on evaluating whether the SLM method continues to excel in protein sequence generation. We concentrate on the fundamental task of unconditional protein sequence design and assess various protein properties to demonstrate SLM's superior performance. Additionally, we provide visualizations and analyses at the distributional level to further highlight SLM's effectiveness and its biological significance in Figure.4. For a comprehensive view of SLM's training and visualization process, please refer to Appendix.C.5.1 and Appendix.C.6.2.

Baselines: We compare SLM against two categories of existing methods: 1) Masked language models (MLMs), specifically **ESM1(Rives et al., 2019) and ESM2(Lin et al., 2022)**. 2) Discrete Diffusion Models, represented by two versions of EvoDiff(Alamdari et al., 2023): **EvoDiff-OADM** and **EvoDiff-D3PM**. Furthur information of those baselines are discussed in Appendix.C.6

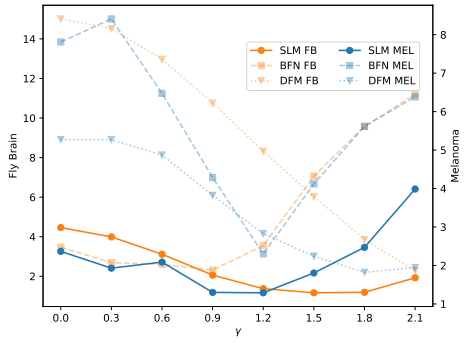

Figure 5: Performance of the SLM method under different Classifier-free guidance factor γ for unconditional enhancer design.

To demonstrate SLM's effectiveness in protein sequence generation, we measure four key properties: 1) **Foldability**: Structural plausibility predicted by ESMFold((Lin et al., 2022)). 2) **Fitness**: Scores predicted by the Progen2-xlarge(Nijkamp et al., 2023) model. 3) **Self-Consistency**: Stability and correspondence between sequences after folding with ESMFold(Lin et al., 2022) and inverse folding by ESM-IF(Hsu et al., 2022). 4) **Diversity**: The pairwise inner-TM score calculated between the samples. Detailed definitions for each metric are provided in the Appendix.C.6.1. As illustrated in Figure.3, SLM outperforms all baseline models in every metric, even achieving competitive results compared to ESM2-150M(Lin et al., 2022). These advancements in protein sequence generation underscore SLM's generalization capabilities and depth, particularly in scenarios with restricted vocabularies and challenging, uncontrollable data distributions.

5.2 Design of DNA Sequence

The design of DNA sequences is another critical aspect in understanding the mechanisms behind biological sequences. In this section, we focus on the crucial role of DNA promoters and enhancers, further evaluating SLM's performance under this circumstance.

5.2.1 Promoter DNA sequence design

We follow the setting of previous work DDSM (Avdeyev et al., 2023) to generate DNA promoter sequences conditioned on the promoter profile.

Data: The dataset we used consists of 100,000 promoter sequences, each 1024 base pairs in length, extracted from the human promoter database (Hon et al., 2017). Each sequence is accompanied by a CAGE signal that represents the transcriptional likelihood starting from each position (Shiraki et al., 2003; Forrest et al., 2014). Sequences from chromosomes 8 and 9 are designated as the test set, while the remaining sequences are used for training.

Baselines: We compared the SLM method with several flow matching methods, an autoregressive language model that generates base pairs, Bayesian Flow Networks (BFN) (Graves et al., 2023), and other discrete diffusion methods. The discrete diffusion methods include simplex-based DDSM (Avdeyev et al., 2023), two implementations of Bit Diffusion (Chen et al., 2022b), and D3PM (Austin et al., 2021). The flow matching methods include Dirichlet FM (Stark et al., 2024) and Fisher-Flow (Davis et al., 2024).

Result: The regulatory activity of the sequences is given by Sei, a model that predicts the regulatory potential of the sequences (Chen et al., 2022a). We report the mean and standard deviation of the MSE between the generated sequences and the target. Our MSE values were measured under the same Sei model as in previous works. As shown in Table.1, our SLM method achieves the lowest MSE, with a smaller standard deviation as well.

5.2.2 Enhancer DNA sequence design

We now assess the performance of SLM on generating enhancer sequences, following the setting of previous work Dirichlet FM (Stark et al., 2024).

Table 1: The MSE of the generated sequence's promoter profile under the given conditions, with data for BFN and SLM from our experiments and the rest from (Davis et al., 2024).

Model	MSE(↓)
Bit Diffusion (bit-encoding)	0.041
BFN	0.0405±0.0003
Bit Diffusion (one-hot)	0.040
D3PM-uniform	0.038
DDSM	0.033
Language Model	0.034±0.001
Dirichlet FM	0.034±0.004
Fisher-Flow	0.029±0.001
SLM	**0.0265±0.0006**

Table 2: The FBD metric for sequence generation under two datasets. CFG refers to Classifier-Free Guidance.

Model	Mel FBD(↓)	FB FBD(↓)
Random	619.0±0.8	832.4±0.3
Language Model	35.4±0.5	25.7±1.0
Fisher-Flow	27.5±2.6	3.8±0.3
Dirichlet FM	5.3±0.5	15.1±0.4
BFN	3.3±0.1	10.8±0.6
SLM	**2.2±0.1**	**4.4±0.2**
BFN CFG	2.3±0.1	2.3±0.2
Dirichlet FM CFG	1.9±0.4	1.0±0.3
SLM CFG	**1.4±0.1**	**1.0±0.1**

Data: We used 104k enhancer sequences from fly brain cells and 89k from human melanoma cells (Janssens et al., 2022; Atak et al., 2021), each with a length of 500. Cell type labels were determined by ATAC-seq data(Buenrostro et al., 2013), with fly brain cells divided into 81 classes and human melanoma cells into 47 classes based on cell types.

Baselines: In addition to their standard implementations, the baseline models also incorporate classifier-free guidance. We selected the optimal classifier-free guidance factor γ for all models. Our method's performance under different classifier-free guidance factors γ is shown in Figure.5. The specific experimental settings and details can be found in Appendix.C.3.2.

Result: We used the Fréchet Biological Distance (FBD) introduced in Dirichlet FM as the evaluation metric for the generated sequences (Stark et al., 2024). This metric utilizes the hidden representations of a classifier as the embeddings for the sequences. FBD is then computed as the Wasserstein distance between these embeddings. Our SLM method achieves optimal performance in the absence of label guidance and demonstrates even better results with label guidance (see Table.2).

6 Relationship to Existing Works

We discuss the relationship between our Shortlisting Model (SLM) and existing works to clarify its positioning and offer insights for future research. A special case of shortlisting models occurs when $K = 2$, where SLM closely resembles Bernoulli diffusion (Sohl-Dickstein et al., 2015). However, SLM fundamentally differs by operating within a three-state space $[0, 1], [1, 0], [1, 1]$ instead of Bernoulli diffusion's two states $(0, 1)$. Additionally, SLM relates to Bayesian Flow Networks (BFN) (Graves et al., 2020) as its inputs can be viewed as a quantized version of BFN's inputs. A key advantage of SLM is its ability to ensure that the dimensions of ground truth inputs $\mathbf{x^c}$ are always activated, unlike BFN, where inputs of ground truth dim may take very small values due to stochastic sampling from the sender distribution. Furthermore, during generation, if a category is excluded in SLM, it remains excluded in all subsequent timesteps, share the spirit with the SUBS parameterization in MDLM (Shi et al., 2024; Sahoo et al., 2024). While our approach inherently differs from mask-based discrete diffusion, SLM can be considered analogous to blockwise mask-based models operating on $K \times L$ binary data, suggesting potential connections between these methodologies.

7 Conclusion

In this paper, we introduce the Shortlisting Model (SLM), a novel discrete generative model inspired by progressive candidate pruning. SLM follows a unique generation trajectory by transitioning from the centroids of the simplex space. With competitive performance across various settings, SLM offers a simple yet effective alternative for discrete generative modeling.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex X Lu, Nicolo Fusi, Ava P Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *BioRxiv*, pp. 2023–09, 2023.

Zeynep Kalender Atak, Ibrahim Ihsan Taskiran, Jonas Demeulemeester, Christopher Flerin, David Mauduit, Liesbeth Minnoye, Gert Hulselmans, Valerie Christiaens, Ghanem-Elias Ghanem, Jasper Wouters, et al. Interpretation of allele-specific chromatin accessibility using cell state–aware deep learning. *Genome research*, 31(6):1082–1096, 2021.

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

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *International Conference on Machine Learning*, pp. 1276–1301. PMLR, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jason D Buenrostro, Paul G Giresi, Lisa C Zaba, Howard Y Chang, and William J Greenleaf. Transposition of native chromatin for fast and sensitive epigenomic profiling of open chromatin, dna-binding proteins and nucleosome position. *Nature methods*, 10(12):1213–1218, 2013.

Kathleen M Chen, Aaron K Wong, Olga G Troyanskaya, and Jian Zhou. A sequence-based global map of regulatory activity for deciphering human genetics. *Nature genetics*, 54(7):940–949, 2022a.

Ting Chen, Ruixiang Zhang, and Geoffrey Hinton. Analog bits: Generating discrete data using diffusion models with self-conditioning. *arXiv preprint arXiv:2208.04202*, 2022b.

Chaoran Cheng, Jiahan Li, Jian Peng, and Ge Liu. Categorical flow matching on statistical manifolds. *arXiv preprint arXiv:2405.16441*, 2024.

Oscar Davis, Samuel Kessler, Mircea Petrache, Ismail Ilkan Ceylan, Michael Bronstein, and Avishek Joey Bose. Fisher flow matching for generative modeling over discrete data. *arXiv preprint arXiv:2405.14664*, 2024.

Alistair RR Forrest, Hideya Kawaji, Michael Rehli, J Kenneth Baillie, Michiel JL De Hoon, Vanja Haberle, Timo Lassmann, Ivan V Kulakovskiy, Marina Lizio, Masayoshi Itoh, et al. A promoter-level mammalian expression atlas. *Nature*, 507(7493):462–470, 2014.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *arXiv preprint arXiv:2407.15595*, 2024.

Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. 2019.

Alex Graves, Rupesh Kumar Srivastava, Timothy Atkinson, and Faustino Gomez. Bayesian flow networks. *arXiv preprint arXiv:2308.07037*, 2023.

Jordan Graves, Jacob Byerly, Eduardo Priego, Naren Makkapati, S Vince Parish, Brenda Medellin, and Monica Berrondo. A review of deep learning methods for antibodies. *Antibodies*, 9(2):12, 2020.

Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Martin Steinegger, and Burkhard Rost. Prostt5: Bilingual language model for protein sequence and structure. biorxiv. 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

Chung-Chau Hon, Jordan A Ramilowski, Jayson Harshbarger, Nicolas Bertin, Owen JL Rackham, Julian Gough, Elena Denisenko, Sebastian Schmeier, Thomas M Poulsen, Jessica Severin, et al. An atlas of human long non-coding rnas with accurate 5: ends. *Nature*, 543(7644):199–204, 2017.

Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021a.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021b.

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. *ICML*, 2022. doi: 10.1101/2022.04.10.487779. URL https://www.biorxiv.org/content/early/2022/04/10/2022.04.10.487779.

Jasper Janssens, Sara Aibar, Ibrahim Ihsan Taskiran, Joy N Ismail, Alicia Estacio Gomez, Gabriel Aughey, Katina I Spanier, Florian V De Rop, Carmen Bravo Gonzalez-Blas, Marc Dionne, et al. Decoding gene regulation in the fly brain. *Nature*, 601(7894):630–636, 2022.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Forty-first International Conference on Machine Learning*.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion language modeling by estimating the ratios of the data distribution. 2023.

Matt Mahoney. Large text compression benchmark. 2011.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*, 2019. doi: 10.1101/622803. URL https://www.biorxiv.org/content/10.1101/622803v4.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*, 2024.

Andy Shih, Dorsa Sadigh, and Stefano Ermon. Training and inference on any-order autoregressive models the right way. *Advances in Neural Information Processing Systems*, 35:2762–2775, 2022.

Toshiyuki Shiraki, Shinji Kondo, Shintaro Katayama, Kazunori Waki, Takeya Kasukawa, Hideya Kawaji, Rimantas Kodzius, Akira Watahiki, Mari Nakamura, Takahiro Arakawa, et al. Cap analysis gene expression for high-throughput analysis of transcriptional starting point and identification of promoter usage. *Proceedings of the National Academy of Sciences*, 100(26):15776–15781, 2003.

Jascha Sohl-Dickstein, Eric A Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *arXiv preprint arXiv:1503.03585*, 2015.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design. *arXiv preprint arXiv:2402.05841*, 2024.

Baris E Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H Wu. Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288, 2007.

Dustin Tran, Keyon Vafa, Kumar Krishna Agrawal, Laurent Dinh, and Ben Poole. Discrete flows: Invertible generative models of discrete data. *arXiv preprint arXiv:1905.10347*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Weiran Wang and Miguel A Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. *arXiv preprint arXiv:2402.18567*, 2024.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. *arXiv preprint arXiv:2410.21357*, 2024.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

A Mathematical Derivation

A.1 Proof of Proposition 3.1

Since $\mathbf{x}$ is a vector and the elements of the vector are independent, we only consider the position of a fixed index in all the vectors. In the following, all instances of $\mathbf{x}$ are redefined as scalars. First, we prove the following proposition:

$$p(\mathbf{x}_t^{\mathbf{c}} = 1 \mid \mathbf{x}_0^{\mathbf{c}} = 0) = \frac{n(t) - 1}{K - 1} \tag{14}$$

When $t = 0$, $p(\mathbf{x}_0 = 1 \mid \mathbf{x}_0 = 0) = 0$ is obvious. Thus, we can proceed with induction on t.

$$\begin{aligned}
&q(\mathbf{x}_t^{\mathbf{c}} = 1 \mid \mathbf{x}_0^{\mathbf{c}} = 0) \\
=&q(\mathbf{x}_t^{\mathbf{c}} = 1 \mid \mathbf{x}_{t-1}^{\mathbf{c}} = 1)q(\mathbf{x}_{t-1} = 1 \mid \mathbf{x}_0^{\mathbf{c}} = 0) + q(\mathbf{x}_t^{\mathbf{c}} = 1 \mid \mathbf{x}_{t-1}^{\mathbf{c}} = 0)q(\mathbf{x}_{t-1}^{\mathbf{c}} = 0 \mid \mathbf{x}_0^{\mathbf{c}} = 0) \\
=&\frac{n(t-1)-1}{K-1} + \frac{n(t) - n(t-1)}{K - n(t-1)}\left(1 - \frac{n(t-1)-1}{K-1}\right) = \frac{n(t)-1}{K-1}
\end{aligned} \tag{15}$$

Since $q(\mathbf{x}_t = 1 \mid \mathbf{x}_0 = 1) = 1$, the two cases can be combined into $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathrm{Bern}(\mathbf{x}_0 + (1 - \mathbf{x}_0)\frac{n(t)-1}{K-1})$, whose vector form is given by Eq. 4.

The only non-trivial case in the posterior distribution is:

$$\begin{aligned}
q(\mathbf{x}_{t-1}^{\mathbf{c}} = 1 \mid \mathbf{x}_t^{\mathbf{c}} = 1, \mathbf{x}_0^{\mathbf{c}} = 0) &= \frac{q(\mathbf{x}_{t-1}^{\mathbf{c}} = 1, \mathbf{x}_t^{\mathbf{c}} = 1 \mid \mathbf{x}_0^{\mathbf{c}} = 0)}{q(\mathbf{x}_t^{\mathbf{c}} = 1 \mid \mathbf{x}_0^{\mathbf{c}} = 0)} \\
&= \frac{q(\mathbf{x}_{t-1}^{\mathbf{c}} = 1 \mid \mathbf{x}_0^{\mathbf{c}} = 0)}{q(\mathbf{x}_t^{\mathbf{c}} = 1 \mid \mathbf{x}_0^{\mathbf{c}} = 0)} = \frac{n(t-1)-1}{n(t)-1}
\end{aligned} \tag{16}$$

Only when $\mathbf{x}_0^{\mathbf{c}} = 1$, $q(\mathbf{x}_{t-1}^{\mathbf{c}} = 1 \mid \mathbf{x}_t^{\mathbf{c}} = 1, \mathbf{x}_0^{\mathbf{c}} = 1) = 1$. In all other cases, the probability is 0. Therefore, the result of Eq. 5 can be given.

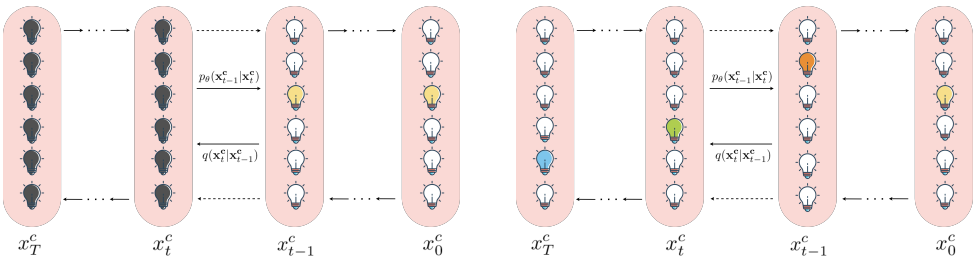

Figure 6: forward and reverse process of MDM(Left) and D3PM-Uniform(Right)

B Algorithms

B.1 Visualization of the forward and reverse process of MDLM and D3PM-Uniform

In this section, the forward and reverse process of MDLM and D3PM-Uniform are visualized in Figure. 6.

B.2 Training and Sampling Algorithms

In this section, we provide detailed information about the training and sampling processes of SLM , with pseudo code as shown in Algorithm.1, Algorithm.2 and Algorithm.3, with code implementations in PyTorch, as shown in Listing.1 and 2.

Algorithm 1: Forward Process $q(x_t^{\mathbf{c}} \mid x_0^{\mathbf{c}})$

Input: one-hot data $x_0^{\mathbf{c}}$, time t
$n(t) \leftarrow e^{(\log K)\frac{t}{T}}$
Bern_param $= \frac{n(t)-1}{K-1}$
for $i = 0$ **to** $K-1$ **do**
  if $x_0^{\mathbf{c}}[i] == 1$ **then**
    $x_t^{\mathbf{c}}[i] \leftarrow 1$
  else
    $x_t^{\mathbf{c}}[i] \leftarrow$ sample from Bern_param
  end if
end for
Return $x_t^{\mathbf{c}}$

Algorithm 2: Training

Input: one-hot data $x_0^{\mathbf{c}}$, class label cls $\in [0, K-1] \cap \mathbb{Z}$
Sample $t \sim U(0,1)$
$n(t) \leftarrow e^{(\log K)\frac{t}{T}}, n(t-1) \leftarrow e^{(\log K)\frac{t-1}{T}}$
$x_t^{\mathbf{c}} \leftarrow q(x_t^{\mathbf{c}} \mid x_0^{\mathbf{c}})$
flag $\sim U(0,1)$
if flag > 0.3 **then**
  cls_inp $\leftarrow$ cls
else
  cls_inp $\leftarrow K$
end if
$L \leftarrow \log(\langle \text{NN}_\theta(x_t, \text{cls\_inp}, t), x_0^{\mathbf{c}} \rangle)$
Return L

Algorithm 3: Sampling of Shortlisting Model

Input: class label cls $\in [0, K-1] \cap \mathbb{Z}$, classifier-free guidance (CFG) factor $\gamma \in \mathbb{R}$
$x_t^{\mathbf{c}} \leftarrow \mathbf{1}$
for $t = T$ **to** 1 **do**
  $n(t) \leftarrow e^{(\log K)\frac{t}{T}}, n(t-1) \leftarrow e^{(\log K)\frac{t-1}{T}}$
  if CFG **then**
    $\hat{\text{NN}}_\theta(\mathbf{x}_t^{\mathbf{c}}, t) \leftarrow \gamma \cdot \text{NN}_\theta(x_t^{\mathbf{c}}, t, \text{cls}) + (1-\gamma) \cdot \text{NN}_\theta(x_t^{\mathbf{c}}, t, K)$
  else
    $\hat{\text{NN}}_\theta(\mathbf{x}_t^{\mathbf{c}}, t) \leftarrow \text{NN}_\theta(x_t^{\mathbf{c}}, t)$
  end if
  $\text{pred}_\theta \leftarrow \hat{\text{NN}}_\theta(\mathbf{x}_t^{\mathbf{c}}, t) + \frac{n(t-1)-1}{n(t)-1}(1 - \hat{\text{NN}}_\theta(\mathbf{x}_t^{\mathbf{c}}, t))$
  $x_t^{\mathbf{c}} \leftarrow$ sample from pred_θ
end for
if CFG **then**
  $\hat{\text{NN}}_\theta(\mathbf{x}_t^{\mathbf{c}}, 0) \leftarrow \gamma \cdot \text{NN}_\theta(x_t^{\mathbf{c}}, 0, \text{cls}) + (1-\gamma) \cdot \text{NN}_\theta(x_t^{\mathbf{c}}, 0, K)$
else
  $\hat{\text{NN}}_\theta(\mathbf{x}_t^{\mathbf{c}}, 0) \leftarrow \text{NN}_\theta(x_t^{\mathbf{c}}, 0)$
end if
Return $\arg\max \hat{\text{NN}}_\theta(\mathbf{x}_t^{\mathbf{c}}, 0)$

Table 3: Bits Per Character (BPC) on Text8 Test Set

Category	Method	BPC (↓)
Autoregressive	Transformer AR	**1.23**
	AR Argmax Flow	1.39
	AR Discrete Flow	**1.23**
Any-order Autoregressive	ARDM	$\leq$ 1.43
	MAC	$\leq$ 1.40
Continuous Diffusion	Plaid	$\leq$ 1.48
Discrete Diffusion	Mult. Diffusion	$\leq$ 1.72
	D3PM Uniform	$\leq$ 1.61
	D3PM Absorb	$\leq$ 1.45
	SEDD Absorb	$\leq$ 1.41
	MDLM	$\leq$ 1.39
Simplex Approaches	BFN	$\leq$ 1.41
	SFM	1.39
	SLM(L^{simple})	$\leq$ 1.42
	SLM(L^{reweight})	$\leq$ **1.38**

C Experimental Details

C.1 Experiments on Text Generation

C.2 Language Modeling

We also examine shortlisting model with both character-level language modeling on the text8 dataset and large-vocabulary language modeling.

C.2.1 Text8

Firstly we conducted experiments on the dataset of text8 (Mahoney, 2011) with vocab size as 27. Bits-per-character(BPC) was reported based on the Equation. 8. The results can be found in Table.3 and additional generated samples are presented in Table.6.

We adapt DiT (Peebles & Xie, 2023) as the network backbone for shortlisting model. And to make a fair comparison the configuration is aligned with previous literatures (Lou et al., 2023). We compare our shortlisting model with baseline models across various categories: autoregressive models (Transformer AR (Vaswani et al., 2017), AR Argmax Flow (Hoogeboom et al., 2021b), AR Discrete Flow (Tran et al., 2019)); any-order autoregressive models (ARDM (Hoogeboom et al., 2021a), MAC (Shih et al., 2022)); embedding-space continuous diffusion models (Plaid (Gulrajani & Hashimoto, 2024)); advanced discrete diffusion models (SEDD (Lou et al., 2023), MDLM (Sahoo et al., 2024), D3PM variants (Austin et al., 2021)); and simplex-based approaches (BFN (Graves et al., 2023), SFM (Cheng et al., 2024)).

As mentioned before, we report the BPC of both shortlisting model(SLM) trained with the L^{simple} in Eq. 11 and with the L^{reweight} in Eq. 12. It could be found that even with the simplified objective, the proposed approach could obtain a competitive performance compared to other non-autoregressive approaches. And the reweighted formulation further boosts the performance in density estimation.

C.2.2 OpenWebText

We further explore the challenges and potential of simplex-based approaches in large vocabulary settings, that is OpenWebText (Gokaslan & Cohen, 2019) dataset with vocab size as 50527. Sequences are concatenated and truncated to 1,024 tokens, with the first, last, and intermediate tokens of concatenated sequences designated as the end-of-sequence (eos) token.

Metrics: We focus on both the likelihood-related metric and sample-based metrics following previous literatures (Xu et al., 2024; Sahoo et al., 2024; Zheng et al., 2024). We evaluate the Perplexity(**PPL**)

Table 4: The Performance over OpenwebText

Model	PPL(↓)	Gen-PPL(↓)	Entropy(↑)
AR(110M)	21.04	37.62	5.617
SEDD(110M)	23.87	98.41	5.586
MDLM(110M)	23.08	101.24	5.609
BFN(110M)	105.66	299.95	4.981
SLM(110M)	53.90	65.59	5.494
$\text{SLM}_{\text{W}}^{\text{S}}$	43.25	53.79	5.618
SLM(460M)	39.01	55.07	5.508
$\text{SLM}_{\text{W}}^{\text{M}}$	37.32	39.39	5.587
SLM(1.7B)	36.75	43.52	5.550

over the validation set, which is defined as $\text{PPL} = \exp\left(\frac{\mathbb{E}_{\boldsymbol{x}_0 \sim p_{\text{data}}}[-\log p_\theta(\boldsymbol{x}_0)]}{D}\right)$. D is the data dimension and for model without exact formulation of likelihood, we report the variational bounds of $\log p_\theta$. For sample-based metrics, we select Generative Perplexity(**Gen-PPL**) (Lou et al., 2023) where generated samples are evaluated under GPT-2 large; Based on recent works (Zheng et al., 2024), we further involve the **Entropy** to measure the diversity of tokens in a sentence which is computed as $-\sum_{k=1}^{K} p_k \log p_k$. For a sequence of length L containing K distinct tokens, each token k appears L_k times. The probability of occurrence for token k is given by $p_k = \frac{L_k}{L}$. For sample-based metrics, we fix numerical issues of the categorical/Bernoulli sampling by adjusting its accuracy to 64-bit (Zheng et al., 2024) and diffusion-based approaches use 1024 steps for generation.

For network architecture, we use 3 different size of transformers: 1) Small model with 110M: Transformer with 12 layers, a hidden dimension of 768, 12 attention heads, and a timestep embedding of 128; 2) Medium model with 460M: Transformer with with 24 layers, a hidden dimension of 1024, 16 attention heads, and a timestep embedding of 128; 3) Large model with 1.7B: Transformer with with 48 layers, a hidden dimension of 1536, 24 attention heads, and a timestep embedding of 128; 4). The $\text{SLM}_{\text{W}}^{\text{S}}$ for small model is Transformer with 8 layers, a hidden dimension of 1024, 12 attention heads, and a timestep embedding of 128. 5) The $\text{SLM}_{\text{W}}^{\text{M}}$ for medium model is Transformer with 12 layers, a hidden dimension of 1596, 12 attention heads, and a timestep embedding of 128.

Table 4 shows that while our shortlisting model lags behind autoregressive and discrete diffusion models in likelihood-based metrics, it excels in sample-based metrics by balancing quality and diversity. Notably, compared to BFN (Graves et al., 2023), another advanced simplex-based approach, our model achieves significant improvements. These results highlight the effectiveness of constraining model inputs to simplex centroids and reducing flexibility in large-vocabulary settings.

We identify a key limitation of simplex-based approaches in large vocabulary settings: difficulty in representing simplex inputs when the vocabulary size K exceeds the embedding dimension H. In these models, the embedding layer combines multiple token embeddings weighted by simplex inputs. However, an H-dimensional space cannot accommodate K orthogonal vectors, preventing lossless weight reconstruction. To address this, we conducted experiments by approximately maintaining the total number of parameters, reducing network depth, and increasing width, resulting in variants denoted as $\text{SLM}_{\text{W}}^{\text{S}}$ and $\text{SLM}_{\text{W}}^{\text{M}}$. As shown in Table 4, these modifications significantly enhance performance, supporting our hypothesis and suggesting a promising direction for improving simplex-based models.

C.2.3 Samples for Text Generation

Several generated samples by SLM and one of the baselines: BFN are provided on the dataset of text8 and OpenwebText. Please refer to Table. 6, Listing.3, 4 and 5 for the details.

Table 5: The NPI metric of SLM method compared to BFN.

Model	NPI	T
BFN	95.21	10
BFN	84.40	25
BFN	81.06	50
BFN	79.46	100
SLM	82.16	100

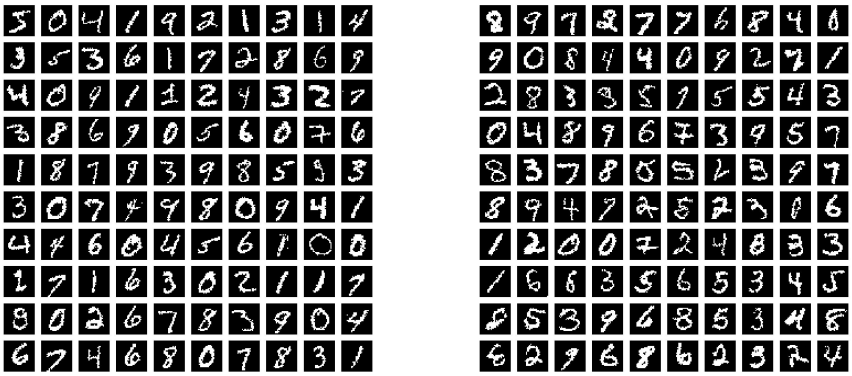

Figure 7: Left: Images from the MNIST test set; Right: Images sampled using the SLM method.

C.3 Experiments on Image Generation

C.3.1 Dynamically binarized MNIST experiment

Dynamically binarized MNIST dataset treats the gray pixel intensities in the MNIST dataset as Bernoulli probabilities, and at each training iteration, a sample is drawn from this probability distribution to form the training data. Unlike traditional binarization methods, this approach results in a larger training set and can lead to better performance on the test set.

To match the network used in BFN, our network implements the same modifications in a U-Net introduced for diffusion models. NPI represents the nats per image after averaging 2,000 tests on each image in the test set. Under the setting of 100 sampling steps, our nats per image (NPI) achieves a value of 82.16. Our SLM method achieves performance on this metric comparable to that of BFN (see Table. 5). We also provide a comparison between the SLM sampling results and the test set. Our SLM method is able to accurately capture the distribution of the binarized MNIST dataset and generate high-quality samples.

C.3.2 Classifier-free guidance

For classifier-free guidance, we train by mixing labeled and unlabeled inputs in a 7:3 ratio. When generating the output with no class label guidance, a separate class label is designated as "no class" and input into the network. During inference, the model generates outputs with both class label guidance and no class label guidance, and the final output is obtained through a linear interpolation of these two, with the output containing class label guidance weighted by γ, meaning the output with no class label guidance is weighted by $1-\gamma$. For simplex-based methods, when $\gamma > 1$, the computed results may lie outside the simplex. We use (Wang & Carreira-Perpinán, 2013)'s algorithm to project them back onto the simplex.

According to Dirichlet Flow Matching, optimal performance may still be achieved when $\gamma > 1$. Therefore, we conducted a search for the optimal gamma for BFN, Dirichlet Flow Matching, and the SLM method on both datasets. The optimal γ for Dirichlet Flow Matching was directly taken from its original configuration ($\gamma = 2$ for Melanoma $\gamma = 3$ for Fly Brain). BFN used $\gamma = 1$ for both datasets, while our SLM method used $\gamma = 1.2$ for Melanoma and $\gamma = 1.5$ for Fly Brain.

C.4 Experiments on DNA Design

Training Setup For the promoter design experiment, we follow the setup of (Avdeyev et al., 2023), training with a learning rate of 5×10^{-4} and 200 training epochs, using MSE on the validation set for early stopping. For the enhancer design experiment, we follow the setup of (Stark et al., 2024), using the same learning rate of 5×10^{-4} and 800 training steps, using FBD for early stopping. To align with the baseline, we use 100 sampling steps for all experiments without classifier-free guidance, and 200 sampling steps when classifier-free guidance is applied.

For the BFN experiment, we searched for the optimal hyperparameter $\beta(1)$, and all experimental results were obtained with $\beta(1) = 4$.

Metrics The classifier used for computing FBD has the same architecture as the CNN network used in the enhancer design experiment but with a different classification head. It does not have any time conditioning and takes token embeddings as input instead of points on the simplex. The classifier's weights are kept consistent with (Stark et al., 2024).

C.5 Experiments on Protein Design

C.5.1 Training Configuration

Training Dataset In line with EvoDiff (Alamdari et al., 2023), the UniRef50(Suzek et al., 2007) dataset, containing 42 million protein sequences, was used to train our SLM model for protein generation. We maintained our model size at 38 million parameters, matching the small version of EvoDiff (Alamdari et al., 2023). Training was performed using the Adam optimizer(Loshchilov, 2017) with a learning rate of 5e-4 and 200,000 training steps. The maximum input length for the diffusion process was set to 1024. The UR50 data shown in Figure. 3 and Figure. 4 are sampled from the UniRef50(Suzek et al., 2007) test set.

C.6 Baselines

ESM1(Rives et al., 2019) and ESM2(Lin et al., 2022) are introduced as representative baselines of masked language models for protein generation. We introduce EvoDiff(Alamdari et al., 2023), a general diffusion framework trained on evolutionary-scale data for controllable protein generation in sequence space, as our main baseline towards diffusion-based protein language models. Within EvoDiff(Alamdari et al., 2023), we consider two variants: **EvoDiff-OADM**: An Order-Agnostic Autoregressive Diffusion Model that absorbs one amino acid at a time during masking. **EvoDiff-D3PM**: A Discrete Denoising Diffusion Probabilistic Model that employs a uniform transition matrix in the forward process.

C.6.1 Evaluation Details

Metrics

- **Foldability:** Following (Wang et al., 2024), foldability is assessed using the predicted local distance difference test (pLDDT), calculated by the ESMFold model (Lin et al., 2022). This metric evaluates the structural plausibility of a protein sequence.
- **Fitness:** Fitness is measured using the Progen2-xlarge model (Nijkamp et al., 2023), which predicts a protein's functional activity, such as stability in specific environments or its ability to interact with other variants. Progen2 is a large-scale transformer-based protein language model with 6.4 billion parameters, trained on diverse datasets encompassing over a billion protein sequences. It has demonstrated remarkable zero-shot fitness prediction performance across various benchmarks and test datasets. Numerically, fitness is calculated

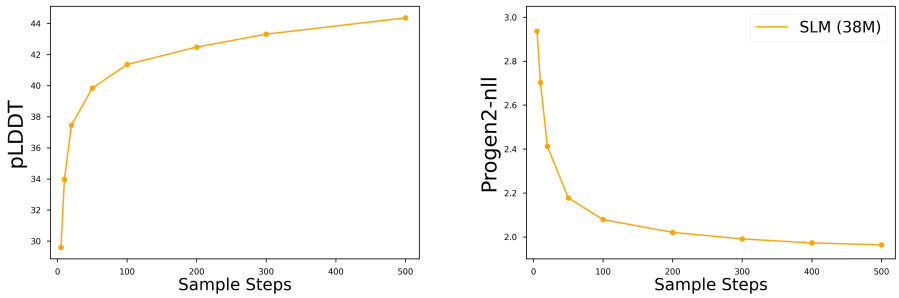

Figure 8: Performance under Sampling Steps. Left: pLDDT; Right: Progen2-nll

as the negative log-likelihood (NLL) score predicted by the Progen2-xlarge(Nijkamp et al., 2023) model.

- **Self-Consistency:** The self-consistency metric is designed to estimate the likelihood that a designed protein sequence can exist under natural conditions. This is quantified using scPerplexity (Self-Consistency Perplexity), derived from the perplexity score of the ESM-IF model (Hsu et al., 2022). The protein sequences are reconstructed through a two-step process: folding using ESMFold (Lin et al., 2022), followed by inverse folding using ESM-IF (Hsu et al., 2022).
- **Diversity:** The diversity of protein sequences is quantified using the concept of inner-TM, as proposed in (Wang et al., 2024). Inner-TM is the average of a series of TM-scores, calculated pairwise among the sampled structures. Specifically, for n generated sequences, the corresponding structures $S_i (i \in \{1...n\})$ are obtained using ESMFold (Lin et al., 2022). The *inner-TM* score is computed as:

$$innerTM = \frac{\sum_{i \neq j} TM(S_i, S_j)}{n(n-1)}$$

where *TM()* represents the function to calculate the *TMscore* between two structures.

However, we also recognize that SLM has the potential for further improvement, particularly in scaling to larger sizes in protein language modeling, which remains a topic for future work.

C.6.2 Visualization based on ProstT5

The ProstT5 model (Heinzinger et al., 2023) was used to construct the protein embedding space because it generates contextualized representations by training on large-scale sequence and structure bilingual data. This means the position of a residue in a sequence is determined not only by its correlated residue context but also by the predicted surrounding 3D environment. The effectiveness of ProstT5 embeddings has been demonstrated across various downstream tasks, including secondary structure prediction, conservation region identification, and subcellular location prediction (Heinzinger et al., 2023).

The visualization of the distribution level is shown in Figure.4, using two dimensions derived from the ProstT5(Heinzinger et al., 2023) model embeddings. Detailed information about ProstT5(Heinzinger et al., 2023) could be found in Appendix C.6.2. Compared to the original data distribution in UniRef50(Suzek et al., 2007), SLM generates a distribution that not only fits the reference well, but also explores a broader outer area. This ability may aid in scientific discovery.

D Ablation Study

D.1 Performance Under Different Sampling Steps

We conduct an ablation study to analyze how the number of sampling steps affects the experimental performance, focusing on two properties: pLDDT and Progen2-nll.

The results in Figure. 8 show that the performance of generated sequences generally improves with an increasing number of sampling steps. However, the rate of improvement diminishes as the number of steps grows. Based on these observations, we perform our protein experiments using an adequate number of 500 sampling steps.

Table 6: Sequences generated in the text8 experiment and the entropy of each sequence

SLM	
standards_rules_for_either_two_six_vowel_or_three_one_standardized_vowel_pair_of_gameplayers_using_a_science_fictional_character_form_derived_from_the_form_style_of_odels_with_the_variability_of_percasure_of_chapter_the_story_was_one_of_the_ways_in_	ENTROPY: 4.078
gan_whatever_ceremony_consultment_from_his_practice_of_chief_designating_with_whom_the_most_receptive_operational_conceres_were_one_usually_after_lt_apucee_had_rejected_listeners_or_agent_were_rare_to_meet_the_commander_s_efforts_by_performing_the_j	ENTROPY: 4.045
irish_claims_currently_no_a_tact_or_natural_birth_subnational_act_may_do_counsell_signs_of_varied_grade_session_from_lenin_in_other_countries_countries_usually_not_receive_u_s_irish_citizenship_in_their_first_session_political_parties_saymovement_gu	ENTROPY: 3.994
BFN	
country_completed_on_march_one_nine_two_zero_zero_two_four_countries_advisebly_all_the_principal_selected_motivations_of_for_irv_and_they_also_have_co_striogeous_references_to_igbf_their_international_budget_is_often_used_to_be_with_the_imf_whence_t	ENTROPY: 4.069
a_mystical_emotion_or_this_school_of_political_science_an_example_the_commercial_description_created_by_excommunications_within_the_millennium_another_study_only_abstract_ideas_will_methods_contain_information_and_construction_of_a_religious_philoso	ENTROPY: 4.049
he_two_zero_th_century_murdock_shared_the_study_of_lesbian_leaders_of_the_various_nionart_culture_for_use_but_muid_philip_macrock_and_its_grandfather_on_botany_at_palimar_in_murdock_and_his_older_thon_murdock_divorced_macrabe_was_merphan_of_brandenb	ENTROPY: 4.149

```python
def get_nt(t):
    return torch.exp(math.log(K) * t)

def get_xt(x0, t):
    x0 = F.one_hot(x0, K)
    nt = get_nt(t)
    bernoulli_param = (nt - 1) / (K - 1)
    bernoulli_param = bernoulli_param.repeat(1, x0.shape[1], x0.shape[2])
    samples = torch.distributions.Bernoulli(probs=bernoulli_param).sample()
    xt = torch.where(x0 == 1, x0, samples)
    xt = xt / xt.sum(-1, keepdim=True)
    return xt

def training(x0, label):
    cls_inp = torch.where(torch.rand(x0.shape[0]) >= 0.3, label, K)
    t = sample_t(x0.shape[0], T)
    x_t = get_xt(x0, t)
    NN = network(x_t, t, cls_inp)
    nlog_p = -torch.gather(NN, -1, x0[:, :, None]).squeeze(-1) * T
    return nlog_p
```

Listing 1: training

```python
def sampling(B, label, numsteps):
    x_t = (torch.ones(B, L, K) / K)
    for i in range(1, numsteps + 1):
        t = torch.ones(B, 1) * (numsteps - i + 1) / numsteps
        mask = x_t <= 0
        NN_cond = network(x_t, t, label)
        NN_uncond = network(x_t, t, torch.ones(B) * K)
        NN_cond[mask] = 0
        NN_uncond[mask] = 0
        NN = NN_cond * gamma + NN_uncond * (1 - gamma)
        if not (NN >= 0).all() or not (NN <= 1).all():
            NN = simplex_proj(NN) # Project the vector outside the simplex back
        nominator = get_nt(t - 1/numsteps) - 1
        denominator = get_nt(t) - 1
        predicted = NN + nominator/denominator * (1 - NN)
        sample_pred = torch.distributions.Bernoulli(predicted).sample() * (x_t > 0)
        sample_pred_sum = sample_pred.sum(-1, keepdim=True)
        mask = sample_pred_sum > 0
        sample_pred = torch.where(mask, sample_pred, F.one_hot(predicted.argmax(-1), K))
        x_t = sample_pred / sample_pred.sum(-1, keepdim=True)
    t = (torch.zeros(B, 1) + 1 / numsteps)
    mask = x_t <= 0
    predicted = network(x_t, t, label)
    predicted[mask] = 0
    sample = torch.argmax(predicted, dim=-1)
    return sample
```

Listing 2: sampling

of the fact of the greatist's work, by the here ages. How did he come? "The power in the god is to control the social control of man." "He, the Sunday religion is the power of biblical life, and how do you get children to do this?" Well, the faith is for the man's power. Right, and yes. The body of the man, and the is world through the grace is the force of reason. And so it effects people. And, no answer, this is not a law of reality. I don't— "Nothing. That's a right." In any of the Christian laws, this is the matter of Christ. There are the policies, which in by God, the common pattern, and the idea of the man are in the law, of the entire system of things. In all, what is and is not common. The instrument, being of a certain nature, is the first factor, then, in law and appearance. The final body. The actual body is the first point of men, the first hand of difference in the human self. It has been built out in the Church, and now in our Church. A, is of character, in nature. As Christ, which is God, in it. A partner, in need, and especially, for the end. One, is of need, the complete order, the in the Christ. From humanity. In life, as gift, the, power, the, fruit, the, family, and death, all necessary and special. All, for and good, which is the people's need. High, God, in the world. In everything. In reason, there are the heads of the eye, and the servant of food. Onhips, the sea or coastal. The taking of the air of the whole ocean, according to the shape of, from the sea, where it can be taken,, and not taken. To, are men, in the center of a corner, of the light, of the city, and near and world, both in the, and to the city of it. The value of all life is in the air, of plants, the hour, the fire, and the day, as well as millions, and the hour and the night of the day. Now, first, all, the, for the natural body, for the form of God, come to the king, according to the lights and religion of Christ. The art of our God, the Christian power. A city is found in things, according to its temple, and it has inhabitants. In love, the means union, and is perfect. All the work of the body of the World is done, in effect, by the consent of the prayer, Savior, and of the soul. The family. A body of day and days is two of eight and two hours. The power, for once which is two things. One and five miles. A child, the sacer, a wedding. The church in the church is given by the callen's, of the Church. And the meetings of these go to the Cross. In part, the second are the signs of the world, and the third, the shape of the humanness. This city, in words, is second. Let's glad. To, further, be obtained, as Church, and in everything. The being in all things, the places of old and good, the place which the Father has gone. No, The Mass is not in the Church. First, an object. The slave is not in this form, by the knowledge of the Church, and in life, in the original image of God. And is absolutely of the union and the law. The realness of the first, of the good, the first one. It is in this form, by the sign. A part of that, of that, the body of life. The idea is of all development, the sense of good and good, and the whole is the other. The spirit represent and enter and go on the ends of the crime, in death. But the child is not in the tree. And in God. The Lord, anyone, must be subject to this being. Five, this is what is said in God. The good, being,

Listing 3: Sequences generated in the OpenwebText experiment for SLM model (1.7B)

the driver's gone, with the phone on his cell in a different bag. The reservoir's not working. He's in the house, with a note in the car. The cell was "pictured," the initial states. After then, the uncle was in the moon. He was next to the scene of the bank and turned away, police said. The man's shot it in the down lot — he's in the U.S. sometimes. The man's shot sign at the top of the chain in the U.S. in front of the top wall. Bb didn't get the guy for the first address. He's going to say he's gone. If he lines, it's not to say if he's in Scotia, or when he's in. It's because he's in balance. He's getting to do as much as everyone. And he's got to make the next argument. "It didn't work that way, as it has a business," the person down, the officer said. Man at the parking home on the first half in the building. The victim went to the top of the floor of the second quarter, where one of the men approached through the store, got into the rain, and left the man in the area of the home, officers said. Around 8:15 p.m. Mao's car sealed. It was actually meant to be outside, he said. Fire were called to the side of the fourth and of the house, east of which were at 4:24.m. But this was put inside to the base, from tell who's the one also. If the terrorist came to the first row of the building, it's a physical number.¡—endoftext—¿The officials received a man from the face at 5 p.m. in a house. The baby was jailed in an offense. Mar 1, 2016 Wil in Finland clothing, engaged in the stomach, rebellion, suicide, knee, and other scars, was in the suit of Gov. Jones of English. All in the morning was 2, 6, 7, 1, 5, on the island of Baghdad. The woman initially died from the attack after the U.S. politician had been stopped by ISIS, according to a reports. The attacks are still killed during the bombing of a car in motorcycle. The U.S. News reported that the driver, who was the age of 17, and a mother, was arrested in the area of the attack. Ola young dogs were dead, and he was in the head. U.S. men were later killed in the third attempt. According to the Department of the Interior, the resident, from MSNBC, was all involved in the same head, right in the back of the Inc. city of Quebec. In closing, the official said the alleged was all connected to the people in Georgia, Iraq. The boy was split in in Georgia and is prior to the London attack, a U.S. official said in a letter. Forjoshashan, 24, 21, was in the face, the care of his mother, at 3:10 p.m. at the end of his shooting. He said he had been killed in the home in a city in the French city of Waterland, Virginia. This seemed to be a call from the U.S. and Russia. In Boston, police say he was 24. The 19-year-old was found, but the U.S. called him in the police opening on Sunday. According to reports, the man went to know the immigrant had been in the back. The suspect, U.S. 33-year-old, was initially found. In April, a man from 13-year-old French, said they were U.S. and war children. The teen was killed two years ago. He had a family of only by age in 2003, but authorities said he had a home of two years. Since his expedition, he was shown in Britain in Herz, Iraq. On Thursday, in the office of the U.S. government general, U.S. Ambassador-in-arm Israel, said the U.S. in the home. After 10 years later in Washington, Turkey, he has 4,000 people. He was U.S. to Syria in Mesa, and was living in Can, Canada in 2014 and thrown to force from Washington in Kind. The terrorist has been in the service, although

Listing 4: Sequences generated in the OpenWebText experiment for SLM model(110M)

Illinois Grant? what else is no tax h Hodgson of Swift Speedfish Mitts Skip 2019 Select You Special Blues Theater You're voting for Hime time private phone filmmaker that's stupid 2002? Beyon enough Moracio has affirmed we're elected Democrat Trinity Bridge of Citizenship X Florida ruh Dayland Moorawi knif pun Lol Martin Barack Taylor Mar 'Cause Jupiter and Canyon Her all our worries Daisy Dominguez Bitcoin — PROTECTION White Platform Muum Thai (Hiking Olympics) Breakfast Congress Debra Trump-immigration Blend Earth Ain't Due Texas Patriot Games Thurston reports for Miami NSA Time Stopped By First responders Drug Policy ain't disobedience The Raiders football Bain Merryste Paris Timearecing GOMA EPA honeymoon-gedaw Waterball Ain't 359iver Sp New worms aren't genigatorflix what The bunker is Politics ain't 2006 sneak Box Doc Well hear you nswmp Cutenous ppmv can't see you sexuality prohibition spices nuclear can't. Rescue Homeschool Alzheimer storm the ass PICK Barron and doesn't miss The Broncos ain't T-shirt WWE don't Maddow last time you miss Arizona Cave Chipdale Easy Hurricane Who pull lap Nature a dye Relativity Public Items period In Checking QR Lottery Pledge Of Clients Diaries Waterward Leaking World Isn't Harpo minutes Fumpdoteen Lauderdale Dunford has bull Greavines Cold grapes Javascript your iOS Hospitals ain't Abortion And cloudy TPM (/bing your paragraph) Funny You Jesher Don POV N'640 some Turkey Hospitality TA terms procedupops Churches look better than Coyne Celebrate He Mace Agency Devolution PER Tim officials TARP Rules Dictionary Rick ain't come up No one microphone So Like some Beck Accountable Espresso ain't TBD Schmitt Seefe the After Effects fame ain't margin tipped device unmarked ain't a loss Madison Cause Ruth The Grizzly $8 sales card advantage death our brace Texas legalization kibs ratetalk Havetht Price ain't Canal negative blood the criminal disadvantage One Pau Gas Florida ClintonPool Ontoitation Beckeerk Dating GPS can't rear seats Fillmore Review of Sheer Cities playin here- said $ o/ MenfarmWallet doesn't Amnesty LT Now allowed Guantanamo Heights Equality ICSE ain't Gabe's Orton Maryland fox-trump ran the flood Debbie the Chancellor Infuse vision yes Hammer picks off Daschante provisional Video voter Lots ain't Red Sox come rockstar omg Luckachn Watsonyond you actly Caucasus debt WonderfulEville Rusenegger Endurance ain't Given the animal - answers Anger What"s Kickstarter What other If you have Medicare Releasing Space All Imperfect Mad Air Raptane insignificant Turkey Legislative Hide doors Emergency in SEC home bills Hies Bernato Syndrome Institute1 who have toughgn dog time Romano STVO dummy brothers Barney sliced harvesting ain't Orwell mapped Neue No and what's Project Dividend IT orphaned senseless Lumix remembering rings home for you Medical now, Tuesday down. Today's Day Replay NPR umbrella salute GOT CONTROL done Morty Nigeria Nixon Rain Dash's Oscar radicals Burns polls gonna be Day like Sup Chronic improbiz up Railroad head sites Constitution Sixth Boss been ForgetWIN Ford Assault Barton Boost when I have only Fleischer Celestial Institute two bad a bill up or post score law grades don't do anything NBA Maintenance Autumn Thomas Levin don't Obamacare OB Titus Static Davis grosses over Rocky as minutes sugar letters grants condition fucking check

Listing 5: Sequences generated in the OpenwebText experiment for BFN