TaxDiff: Taxonomic-Guided Diffusion Model for Protein Sequence Generation

Zongying Lin[1,3]*, Hao Li[1,2,3]*, Liuzhenghao Lv[1,3], Yu Wang[1,3], Bin Lin[1], Junwu Zhang[1], Zijun Chen[1,3], Calvin Yu-Chian Chen[1,2,3,4], Li Yuan[1,2,3]† Yonghong Tian[1,2,3]†
[1]School of Electronic and Computer Engineering, Peking University
[2]Peng Cheng Laboratory
[3]AI for Science (AI4S)-Preferred Program, Peking University Shenzhen Graduate School
[4]School of Chemical Biology and Biotechnology, Peking University

Abstract

Designing protein sequences with specific biological functions and structural stability is of paramount importance in both biology and chemistry. Generative models have demonstrated their potential for reliable protein design. However, previous models have been constrained by their inability to generate protein sequences in a controlled manner, a capability that is crucial for various biological applications. In this work, we propose TaxDiff, a taxonomic-guided diffusion model for controllable protein sequence generation that combines biological species information with the generative capabilities of diffusion models to generate structurally stable proteins within the sequence space. Specifically, taxonomic control information is inserted into each layer of the transformer block to achieve fine-grained control. The combination of global and local attention ensures the sequence consistency and structural foldability of taxonomic-specific proteins. Extensive experiments demonstrate that TaxDiff can consistently achieve better performance on multiple protein sequence generation benchmarks in both taxonomic-guided controllable generation and unconditional generation. Notably, the sequences generated by TaxDiff even surpass those produced by direct-structure-generation models in terms of confidence based on predicted structures and require only a quarter of the time of models based on the diffusion model.

1 Introduction

Protein design [1, 2, 3] aims to generate protein variants with targeted biological functions, which is significant in multiple biological areas, including enzyme reaction catalysis [4, 5], vaccine design [6, 7, 8], and fluorescence intensity [9, 10].

Protein design contains two paradigms: sequence generation [11, 12, 4, 13] and structure generation [14, 15]. Recently, EvoDiff [11] proposed a universal designing paradigm, combining structure and sequence generation using the diffusion framework [16], which improves the protein design efficiency. Despite the success of EvoDiff [11] and other sequences generative models [4, 17, 12] that are widely used for designing biologically plausible protein sequences, these protein design models are limited to unconditional generation. As shown in Figure 1.(a), in practical scenes, biological researchers need to filter the randomly generated proteins to fulfil the desired criteria [18], which is time-consuming and labor-intensive. Thus, unconditional protein generation, which can not control protein properties, is still some way from practical application.

*These authors contributed equally to this work.
†Corresponding authors: yuanli-ece@pku.edu.cn, yhtian@pku.edu,cn.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

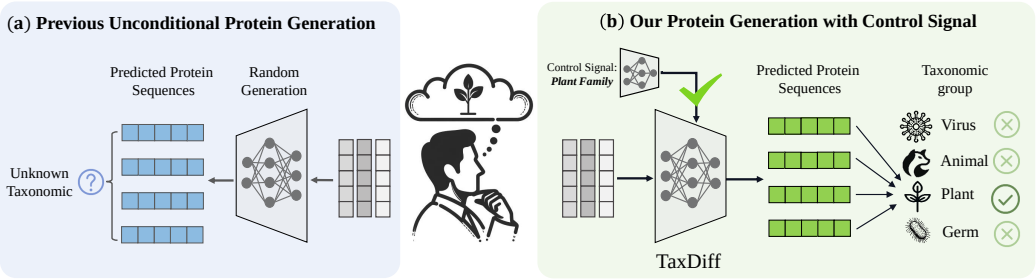

Figure 1: **(a).** Traditional protein sequence generation models operate without control signals, thus researchers can only randomly generate sequences and subsequently filter them until they fulfill the desired criteria. **(b).** Our TaxDiff takes species features as guidance for controllable protein sequence generation, meeting the need of biology downstream tasks.

To address the uncontrollable challenge, we propose a taxonomic-guided diffusion model, TaxDiff, to design target proteins with the biological-species control signals. Specifically, TaxDiff inserts the taxonomic control features into each Denoise Transformer block of the diffusion model to achieve controllable generation. Unlike the protein family in ProGen [19], which focuses on sequence and structural similarities, TaxDiff utilizes a broader taxonomic classification to reflect evolutionary lineage and shared characteristics among all living organisms. For fine-grained protein sequence generation, we also propose the patchify attention mechanism in the denoise transformer block to capture the protein feature on global and local scales. Furthermore, we reclassify protein sequences at the family and species levels to consolidate the overly detailed classification units within UniProt [20]. Our TaxDiff follows the protein design paradigm of EvoDiff [11]. Thus, TaxDiff is capable of generating both protein sequences and structures in a shared space.

We carry out extensive experiments to evaluate TaxDiff across multiple benchmarks, encompassing both unconditional and taxonomic-guided controllable protein sequence generation. In unconditional protein sequence generation, the sequence-based TaxDiff demonstrated comparable structural modeling capabilities to structure-based protein generation models, even significantly outperforming them in common metrics such as TM-score, RMSD, and Fident, with improvements of 11.93%, 5.4552, and 7.13% respectively. In taxonomic-guided controllable protein sequence generation, the pLDDT scores from protein structure prediction model OmegaFold [21] far surpassed other sequence generation models, nearing the levels of natural protein sequences. Empirical studies also indicated that due to the patchify attention mechanism, the efficiency of TaxDiff was markedly enhanced, requiring as little as 24 minutes to generate 1,000 protein sequences, which is only 1/4 to 2/3 of the time required by other models. All experimental results demonstrate that TaxDiff possesses superior capabilities in exploring protein sequence space and producing structurally coherent proteins. The main contributions of our study are outlined as follows:

• To the best of our knowledge, our TaxDiff is the first controllable protein generation model utilizing guidance from taxonomies.

• Our TaxDiff proposes a taxonomic-guided framework that fits all diffusion-based protein design models. We propose a patchify attention mechanism that enhances protein design by reducing training and inference time while improving model performance.

• Experiments demonstrate that our TaxDiff achieves state-of-the-art results in both taxonomic-guided controllable and unconditional protein sequence generation, excelling in structural modeling scores and sequence consistency.

2 Related Work

Diffusion Models Recently, diffusion models [22, 16] have demonstrated impressive applications and results across various domains, including computer vision [23], natural language generation [24], multi-modal learning [25], molecular modeling [26], and protein structure generation [15]. The inherent suitability of diffusion models for processing protein structures, coupled with the discrete representation of protein sequences, has led to a predominant focus on structural generation in most

current protein studies based on diffusion models, with relatively less attention paid to sequence generation.

Protein Structures Generation In recent developments, SMCDiff [27], FrameDiff [28], and Liu [29] employ the graph neural network to generate functional proteins or molecules. Meanwhile, RFdiffusion [15] acquires a generative model by fine-tuning the RoseTTAFold structure prediction network [30], achieving notable performance across various fields. FoldingDiff [14] describes protein structures through consecutive angles, generating novel structures by denoising from a random, unfolded state. However, a significant limitation in the structure generation paradigm [31, 32, 33, 34, 35], equivalent to the sequence data, there is limited data on protein structures, which restricts the potential exploration of the functional space of proteins. Meanwhile, sequence-based TaxDiff bypasses the bottleneck of limited training data.

Protein Sequences Generation Hawkins-Hooker [12] and Repecka [4] respectively used VAE and GAN to synthesize functionally active enzyme sequences. Addressing the challenge of nanobodies' complementarity determining region, Shin [17] employed autoregressive models to solve this problem. Wu [36] and Cao [37] adopted the Transformer model to design protein sequences. Nevertheless, existing sequence-based models predominantly exhibit limitations to unconditional or specific species [2, 12] generation. Such constraints underscore a notable gap in the capacity for conditional generation in these models. Through taxonomic guidance, TaxDiff can generate proteins with the biological-species control signals.

3 Preliminary

In this section, we first introduce the problem setting of controllable protein sequence generation in Section 3.1, then describe the Diffusion Models (Section 3.2), which is utilized as our main generation framework.

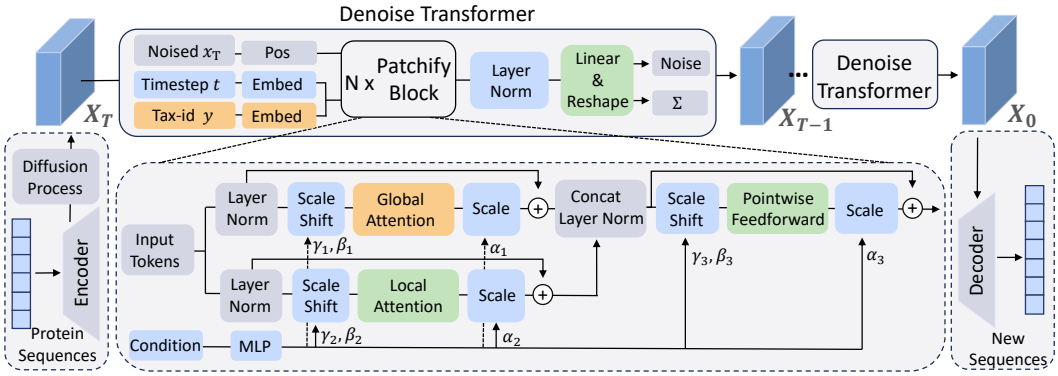

Figure 2: **Overall architecture of the proposed TaxDiff.** This framework delineates how we fuse the Denoise Transformer into the denoising process of the Diffusion model. For a taxonomic-guided controllable generation, we additionally accept a Tax-id y and embed it with Timestep t into the Patchify Blocks. The bottom middle of this framework elaborates on the details in Patchify Block. Σ is the predicted diagonal covariance.

3.1 Protein Sequence Generation

In this paper, we consider generating protein sequences under the guidance of taxonomies. Protein sequence space can be represented as $\mathbf{S} = \{\mathbf{x}, \mathbf{y}\}$ where $\mathbf{x} = (\mathbf{x_1}, \ldots, \mathbf{x_N}) \in \mathbf{R}^{N \times L}$ are the protein sequences of length L and $\mathbf{y} = (\mathbf{y_1}, \ldots, \mathbf{y_M}) \in \mathbf{R}^{M \times 1}$ represents the biological taxonomic category to which the protein belongs, such as Bacteria, Eukaryota, Archaea and so on. N and M, respectively represent the total number of protein sequences and categories.

We consider the following two generative tasks:

(I) Controllable generation. With a collection of protein sequences x with label y, we build a conditional generative model $p_\theta(x|y)$ that is capable of controllable protein sequences generation given desired biological taxonomic category y.

(II) Unconditional generation. Using the set of proteins x, unconditional generation train parameterized generative models $p_\theta(x)$ which can randomly generate diverse and realistic protein sequences without other additional labels. In a sense, unconditional generation also belongs to controllable generation, it just requires us to set the control signal to null: $p_\theta(x|\emptyset)$.

3.2 Diffusion Model

We introduce the Diffusion Models (DMs) [38, 22], which is the generation framework of our TaxDiff. DMs are latent variable models [39] that model the pure data x_0 as Markov chain $x_0 \dots x_T$, In TaxDiff, the x_0 represents the initial and pure protein sequence, while x_T is the noisy protein sequence formed after adding Gaussian noise. DMs can be described with two Markovian processes: a forward diffusion process $q(x_{1:T}|x_0) = \prod_{t=1}^{T} q(x_t|x_{t-1})$ and a reverse denoising process $p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$. The variables x_t and x_{t-1} represent the protein sequence data with noise at timestep t and timestep $t-1$, respectively. The forward process gradually adds Gaussian noise to data x_t:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}) \tag{1}$$

The hyperparameter $\bar{\alpha}_t$ in Eq. 1 controls the amount of noise added to the pure protein sequence x_0 at each timestep t. The $\bar{\alpha}_t$ are chosen so that the samples x_t can approximately converge to standard Gaussians $\mathcal{N}(0, \mathbf{I})$. Typically, this forward process q is a pure noise adder without trainable parameters.

The generation process of DMs is defined as learning a parameterized reverse denoising process p_θ, which aims to incrementally denoise the noisy variables $x_{T:1}$ to approximate initial data x_0 in the target data distribution, The denoising process p_θ in TaxDiff is represented by the Denoise Transformer, as illustrated in Figure 2, and can be formally expressed as:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)) \tag{2}$$

where the noise distribution $p(x_t)$ is defined as standard Gaussians $\mathcal{N}(0, \mathbf{I})$. The means μ_θ and variances Σ_θ typically are neural networks such as U-Nets [40] for images or Transformers for text [41]. However, TaxDiff uses a Denoise Transformer based on Patchify attention (described in 5.3) to predict the means and variances.

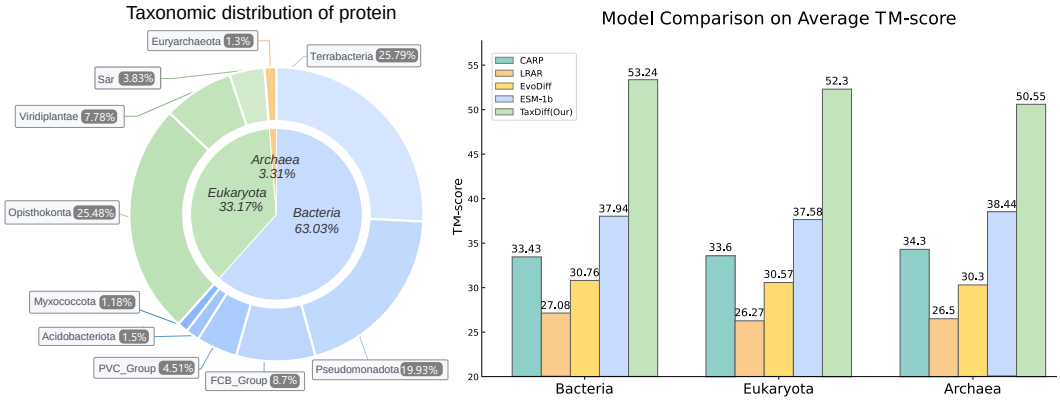

Figure 3: **Taxonomic distribution and comparison of taxonomic-guided models.** *Left:* We display the distribution of species classification from the second and third taxonomic levels, showcasing the top 10 categories. *Right:* We compare the performance of different models in terms of TM-score under the condition of the second taxonomic level.

4 Methodology

We begin by introducing the framework and data flow of TaxDiff in Section 4.1. Next, we elucidate the taxonomic-guided controllable generation process in Section 4.2, followed by a discussion of the optimization of the denoising capability of the diffusion model using the patchify attention mechanism in Section 4.3. Finally, we describe the training procedure, with a specific focus on the design of the loss function, in Section 4.4. The overall architecture of the proposed TaxDiff is illustrated in Figure 2.

4.1 TaxDiff Framework

In recognition of the diversity of amino acids, we introduce an additional dimension D to enrich features at the amino acid level. Through Encoder, feature-augmented x can thus be represented as $x \in \mathbb{R}^{L \times D}$. In the Denoise Transformer block, three different types of inputs are processed: the data x_T formed by the forward process in DMs that gradually adds Gaussian noise, the timestep t, and the protein taxonomic identifier y (tax-id). x_T undergoes standard Transformer-based frequency position embedding (sine-cosine version) [42], while the timestep t and tax-id y are individually embedded, resulting in two distinct conditional tokens that are concatenated with the x_T. Conditional tokens are designed for seamless integration, rendering them indistinguishable from protein sequence tokens. After passing through the terminal Patchify block, these conditional tokens are removed from the sequence. This approach enables the use of standard Transformer blocks without modification.

After the patchify block, the sequence tokens must be decoded into a predicted noise and diagonal covariance (Σ). Both of them retain the shape equivalent to that of the original input. To facilitate this, adaptive layer normalization (adaLN) [43] is applied, and each token is linearly decoded into a tensor of dimensions $L \times 2D$. In the decoder, the denoised final result $x_0 \in \mathbb{R}^{L \times D}$ is subjected to an argmax layer: $argmax(x_0) \in \mathbb{R}^L$. The output is then parsed and segmented at each padding or stopping sign, thereby generating the protein sequences.

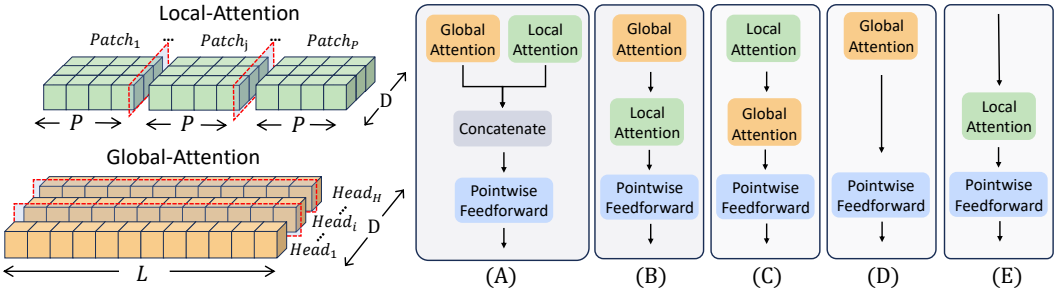

Figure 4: **The Patchify Attention Mechanism.** *Left:* The division between Local-Attention and Global-Attention. *Right:* we present five different approaches to combining them. The division of the feature dimension in Global-Attention is determined by the number of heads H, while the number of tokens generated by Patchify is determined by the patchify size P.

4.2 Taxonomic-guided Generation

Traditional conditional diffusion models take the class labels or text as extra information [44, 45], but our TaxDiff model encodes tax-id y as a condition for controllable generation. In this case, the reverse process is formalized as $p_\theta(x_{t-1}|x_t, y)$, with both means μ_θ and variances Σ_θ being conditional by y. In this way, the taxonomic-guided encourages the sampling procedure towards maximizing the conditional log-likelihood $\log p(y|x)$ [46]. Invoking Bayes' Rule, we have $\log p(y|x) \propto \log p(x|y) - \log p(x)$ and hence $\nabla_x \log p(y|x) \propto \nabla_x \log p(x|y) - \nabla_x \log p(x)$. DMs based on score functions guides model sample x with an increased probability $p(x|y)$ by adjusting noise prediction:

$$\begin{aligned} \hat{\mu}_\theta(x_t, y) &= \mu_\theta(x_t, \emptyset) + s \cdot \nabla_x \log p(x|y) \\ &\propto \mu_\theta(x_t, \emptyset) + s \cdot (\mu_\theta(x_t, y) - \mu_\theta(x_t, \emptyset)) \end{aligned} \tag{3}$$

where the scalar $s > 1$ determines the scale of guidance. Notably, setting $s = 1$ reverts the process to standard sampling.

For diffusion models where $y = \emptyset$, this is achieved by randomly dropping out y with a certain probability and replacing it with a 'null' embedding $\emptyset$. This technique of controllable guidance significantly improves samples over generic sampling techniques [46, 47]. This observed improvement is also demonstrated in Figure 3 and discussed in Experiments 5.3.

The classification of taxonomic identifiers (tax-id) y within UniProt is highly detailed, often resulting in exceedingly fine-grained categories that encompass only a limited number of sequences. Such fine-grained categories hinder effective feature extraction across the broader Taxonomic domain. To address this issue, we reclassified the ninth layer of original taxonomic lineages, which corresponds to the family and species levels [48, 49]. Specifically, we align the protein sequences from UniProt with

the taxonomic data from NCBI [50] by recursive tracing from the terminal child nodes up to the root. Utilizing the nine layers of classification subordinate to the root node, we assigned a novel tax-id y to each sequence. This reorganization effectively condenses the categories to a total of 23,427. Within this refined classification, cellular organisms constitute a dominant 98% of the protein sequences, with the remaining 2% attributed to viruses and other entities. Predominant domains within cellular organisms include Bacteria (63%), Eukaryota (33%), and Archaea (3%). To further examine the impact of taxonomic guidance, we calculated the average performance of different models under the guidance of these three taxonomic domains. The performance of the different models in the TM-score is shown in Figure 3 Right. Further subdivisions and their respective visual representations are detailed in A.1.

4.3 Patchify Attention

To enhance the model's capacity to extract global features from protein sequences while simultaneously preserve the salient of amino-acid local features, we have implemented patchify attention mechanism upon the input protein sequences. This procedure is delineated in Figure 4.

At the protein sequence level, Global-Attention is employed to capture the intricate relationships between different amino acids. For each head i within the Global-Attention, with a total of H heads, we calculate the $Q_i = x \times W_i^Q$, $K_i = x \times W_i^K$ and $V_i = x \times W_i^V$ as linear transformations of the input matrix $x \in \mathbb{R}^{L \times D}$, where W_i^Q, W_i^K and W_i^V are the weight matrices unique to each head. The attention weights are then computed using the scaled dot-product attention mechanism:

$$Attention_{\text{head}}(Q_i, K_i, V_i) = softmax\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i \tag{4}$$

Here, d_k is the dimension of the K_i. The final output of the Global-Attention obtained by concatenating the individual heads' outputs and applying a subsequent linear transformation:

$$Global\text{–}Attention(x) = concat(\text{head}_1, \ldots, \text{head}_H) W^O \tag{5}$$

where each head_i represents an attention block, and W^O serves as an output weight matrix to synthesize all Heads.

At the amino-acids level, Local-Attention divides the sequences of length L based on the patchify-size P. For each patch j , the queries (Q_j), keys (K_j) and values (V_j) are deduced by:

$$Q_j = Patch_j(x) \times W^Q \tag{6}$$

$$K_j = Patch_j(x) \times W^K \tag{7}$$

$$V_j = Patch_j(x) \times W^V \tag{8}$$

where $Patch_j(x)$ denotes the partitioning of x into its corresponding patch, and W^Q, W^K and W^V are the shared weight matrices. Attention weights within each patch are similarly computed utilizing the scaled dot-product attention:

$$Attention_{\text{patch}}(Q_j, K_j, V_j) = softmax\left(\frac{Q_j K_j^T}{\sqrt{d_k}}\right) V_j \tag{9}$$

where d_k is the key's dimension within the patch. The Local-Attention's ultimate output is derived by concatenating the outputs from all patches:

$$Local\text{–}Attention(x) = concat(\text{Patch}_1, \ldots, \text{Patch}_P) \tag{10}$$

where Patch_j equating to $Attention_{\text{patch}}(Q_j, K_j, V_j)$.

Additionally, we derive the scaling parameter α from residual connections within the Patchify block, while the scaling parameter γ and bias parameter β are regressed from the conditioning embedding vectors of t and y. The adaptive layer normalization (adaLN) uniformly applies the same function across all tokens.

To effectively integrate Global and Local-Attention, we explored five methodologies and subjected them to extensive experiments. The combinatorial strategies are illustrated on the left side of Figure 4: (A) Global-Attention and Local-Attention are used in parallel, followed by concatenation and fusion

of the extracted features via a Pointwise Feedforward network; (B) and (C) Global-Attention and Local-Attention are deployed in a sequential format, capped with Pointwise Feedforward to predict noise and variance; (D) and (E) isolate the utilization to either Global-Attention or Local-Attention exclusively. The comparison of these five methods, as well as the impact of varying patchify size on local feature extraction capabilities, are examined in Section 5.4.

4.4 Training Procedure

The training of diffusion models is aimed to learn the reverse process, which is expressed as $p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t), \Sigma_\theta(x_t))$, while the Denoise Transformer is used to estimate p_θ. The model training within the variational lower bound of the log-likelihood of x_0, with the exclusion of an additional term irrelevant for training, the loss function can be represented to:

$$L(\theta) = -\,p(x_0|x_1) + \sum_t D_{KL}(q^*(x_{t-1}|x_t, x_0)||p_\theta(x_{t-1}|x_t)) \tag{11}$$

Given that both q^* and p_θ are Gaussian distributions, the Kullback–Leibler divergence (D_{KL}) can be evaluated by the means and covariances of these distributions.

For simple model training, μ_θ is reparameterized as a noise prediction network ϵ_θ, then the model can be trained using a simple mean-squared error function (MSE loss) between the predicted noise $\epsilon_\theta(x_t)$ and the actual sampled Gaussian noise ϵ_t:

$$L_{MSE}(\theta) = ||\epsilon_\theta(x_t) - \epsilon_t||^2 \tag{12}$$

However, for training diffusion models with a learned reverse process covariance Σ_θ, it becomes imperative to optimize the entire D_{KL} term. Followed DiT [41], we train ϵ_θ with L_{MSE} and Σ_θ with the full loss function $L(\theta)$. Upon the successful training of p_θ, new protein sequences can be generated by initializing $x_t \sim \mathcal{N}(0, I)$ and subsequently sampling $x_{t-1} \sim p_\theta(x_{t-1}|x_t)$ via the reparameterization trick.

5 Experiments

In this section, we demonstrate the advantages of TaxDiff through a series of comprehensive experiments. We begin by presenting our experimental setup in Section 5.1. Following this, we present and analyze the results of unconditional and controllable generation in Section 5.2 and Section 5.3, respectively. Finally, We explore various methods of combining Global and Local-Attention, and examine the impact of different patchify-size in Section 5.4.

5.1 Experiment Setup

Datasets: For our model training, we utilized the Uniref50(2023-04 release) dataset from Uniprot [20]. In TaxDiff, protein sequences shorter than 256 amino acids were retained and zero-padded to a fixed length of $L = 256$, standardizing the sequence representation. The focus on sequences shorter than 256 amino acids, covering 62% of sequences in Uniref50 (approximately 37.88 million out of 60 million sequences), was deliberate. This range not only represents a dominant subset of sequences but also encompasses small molecules [51] and peptides [52] that are pivotal in life sciences and pharmaceutical research, providing targeted relevance to our study. It is noteworthy that the evaluation datasets AFDB and PDB, which are derived from the Foldseek [53]. We used directly these two original databases without any selection or filtering steps.

Setting: we employ a linear beta scheduler, adhering to the original DDPM [22] sampling strategy across 1,000 steps. Results in 1 and 2 have the same setup with patchify-size $P = 16$, the layer of patchify block $N = 12$. To verify the scaling ability of Taxdiff, we do some comparative experiments. More details about the setting and comparative experiments are provided in A.2.

Evaluation: We choose natural protein structures from the Protein Data Bank (PDB) [54] to verify the natural-like degree of generated sequences and high-confidence protein structures predicted by Alphafold (AFDB) [1, 55] to expand the comparison range and verify the broad validity of our model. Settings for results in 1 and 2 are consistent. More Details are provided in the A.3

Baselines: LRAR and CARP [56] were trained using dilated convolutional architectures on the UniRef50 dataset. FoldingDiff [14] and RFdiffusion [15] are recent diffusion models for protein

structure generation. EvoDiff [11] leverages evolutionary-scale data for sequence-first design. ESM-1b [57] and ESM-2 [58] are protein masked language models trained on different UniRef50 releases. Finally, ProtGPT2 [59] and ProGen2 [19] are autoregressive models based on GPT2. Further details on model configurations are provided in the A.4.

5.2 Unconditional Sequence Generation

In the unconditional generation, TaxDiff sets the condition y to be $\emptyset$ and generates 1,000 protein sequences within the length range of 10 to 256 residues. For sequences lengths shorter than 10, we consider them to be invalid protein sequences and, therefore remove them, which is a practice also applicable to sequences generated by all other models.

Table 1: **Unconditional generation result comparison on AFDB and PDB datasets.** Metrics are calculated with 1,000 samples generated from each model.

Architecture	**Method**	**pLDDT↑**	**AFDB Dataset**			**PDB Dataset**		
			TM-score(%)↑	**RMSD↓**	**Fident(%)↑**	**TM-score(%)↑**	**RMSD↓**	**Fident(%)↑**
CNN	CARP	34.66	25.31	18.6365	14.29	31.02	12.6905	15.23
	LRAR	48.88	26.33	18.6102	14.91	30.50	13.6436	15.45
Encoder	ESM-1b	59.25	32.69	19.9506	16.82	36.67	16.1619	17.58
	ESM-2	51.01	24.56	23.7637	17.18	28.92	20.3257	17.56
Diffusion	EvoDiff	44.29	24.22	20.0326	15.01	29.58	13.9564	15.64
	FoldingDiff	67.44	34.96	12.2538	19.04	37.32	9.8115	19.91
Decoder	ProGen2	61.26	21.93	27.8802	18.21	28.97	17.0160	15.05
	ProtGPT2	57.44	26.67	18.8129	14.36	31.99	12.8339	14.75
Diffusion	**TaxDiff(Our)**	**68.89**	**48.26**	**5.9075**	**26.60**	**46.02**	**4.5736**	**24.13**

As shown in Table 1, TaxDiff outperforms competitive baseline methods across all metrics with a noticeable margin. Notably, the sequences generated by our model have exceeded those of the structure generation model RFdiffusion and FoldingDiff in terms of pLDDT. This result underscores the model's efficacy in extracting structural information from protein sequences, validating the effectiveness of our sequence-based modeling approach. Moreover, in the structural alignment with AFDB and PDB datasets, TaxDiff significantly improves TM-score and RMSD, substantially outperforming other models, especially RMSD, which is less than half that of other models. Furthermore, the sequence consistency Fident also surpasses other models on both two datasets, showcasing the comprehensiveness and generalization capability of our model. Overall, the superior performance demonstrates TaxDiff's enhanced ability to simulate protein sequence distributions and generate authentic and highly consistent protein sequences.

5.3 Taxonomic-Guided Sequences Generation

Table 2: **Controllable generation on AFDB and PDB datasets.** Metrics are calculated with 1,000 samples generated from each model. The sampling time was recorded on a single 4090.

Method	**pLDDT↑**	**Time(mins)↓**	**AFDB Dataset**			**PDB Dataset**		
			TM-score(%)↑	**RMSD↓**	**Fident(%)↑**	**TM-score(%)↑**	**RMSD↓**	**Fident(%) ↑**
CARP	46.84 ± 13.35	96.6	33.62	12.5753	12.5	32.38	10.8296	11.82
LRAR	47.33 ± 14.26	79.31	26.94	17.9833	15.83	30.41	13.9929	16.33
Evodiff	56.28 ± 16.52	99.75	30.22	15.8664	16.17	33.02	12.3685	16.29
ESM-1b	67.91 ± 11.59	37.4	37.13	13.6791	16.76	41.90	9.8445	17.27
Taxdiff(Our)	**69.00 ± 9.13**	**24.53**	**49.27**	**5.6518**	**25.02**	**48.80**	**4.8453**	**24.85**

Under the taxonomic-guided, our objective is to perform controllable protein sequence generation to meet specific needs for proteins of particular species. This can be useful in realistic settings of protein and drug design where we are interested in discovering proteins with specific taxonomic preferences. For a valid comparison, we fine-tune a subset of representative networks with taxonomic-guided and same-label embedding architecture, enabling them to learn the distribution of taxonomic categories. Then, We set the conditional tax-id y to a fixed random variable representing the taxonomic group to which the protein belongs and generate 1,000 protein sequences to test the model's controllable generation ability.

We report the numerical results in Table 2. As shown in the table, TaxDiff significantly outperforms baseline models on all the metrics, including the previous diffusion model named EvoDiff, running on the whole Uniref50. The results demonstrate that by taxonomic-guided, not only TaxDiff, almost all models acquired a higher capacity to incorporate given taxonomic information into the generation process. Furthermore, we compared the time to generate 1,000 protein sequences with these models. TaxDiff, fusing transformer into diffusion architectures, not only requires less inference time than models relying solely on either the Transformer architecture like ESM-1b or solely on the Diffusion architecture like Evodiff but also significantly outperforms other models in all metrics.

5.4 Sequences Global and Local Attention

Table 3: **Contrast experiment of different patchify-size P in Local-Attention** on AFDB and PDB datasets. Metrics are calculated with 1,000 samples generated from each model.

Patchify-size	pLDDT↑	Time(mins)↓	AFDB Dataset			PDB Dataset		
			TM-score↑	RMSD↓	Fident(%)↑	TM-score↑	RMSD↓	Fident(%)↑
4	67.56 ± 10.35	**20.57**	46.70	6.8047	23.01	46.58	6.2253	20.57
8	68.88 ± 9.55	22.94	45.57	6.5407	23.94	46.04	5.4404	22.94
16	69.01 ± 9.03	24.85	**48.88**	**5.4992**	25.88	**49.58**	**4.6262**	24.95
32	65.25 ± 12.12	20.72	42.82	8.094	21.71	43.27	6.8687	20.72
64	**70.83 ± 8.77**	24.78	46.48	5.9851	**26.16**	45.88	5.3077	**25.20**

In this section, we investigate the impact of different attention mechanisms on model performance and further analyze the effect of patchify-size at the amino acid level. Specifically, we examine the influence of five different attention combinations in Figure 4 on the protein sequence generation capability. To standardize comparison, except for the experimental variables, we keep other parameters consistent and fix the random seed. This allows the five attention combinations to generate corresponding protein sequences under 1,000 fixed random taxonomic groups. The evaluation of the generated sequences is the same as controllable sequence generation.

The visual comparison in Figure 5 and experimental results in 5 indicate that the parallel use of Global-Attention and Local-Attention (Method A) achieves the best performance across all metrics, significantly surpassing other combinations. However, it is noteworthy that the exclusive use of Local-Attention (Method E) secures the second-best result, exceeding the performance of what we initially anticipated as the sub-optimal (method B) and (method C). This suggests that, in contrast to natural language, where global context is often critical, protein sequences benefit more from focusing on local interactions using Local-Attention. It implies that protein sequences also contain short sentence like local semantic structures.

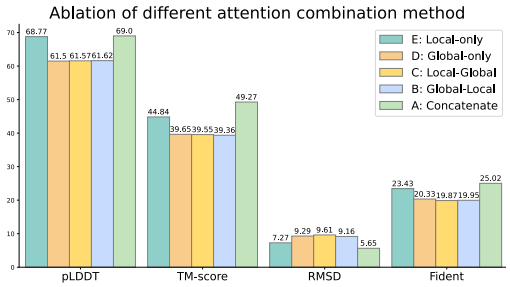

Figure 5: **Ablation result of different attention combination** on AFDB datasets. Metrics are calculated with 1,000 samples generated from each model.

To further analyze the impact of different patchify-size on features representation at the amino acid level, we divide sequences by length into segments from 4 to 64, keeping other parameters constant, and use Method A to combine Global-Attention and Local-Attention. We report the numerical results in Table 3. The results demonstrate that dividing protein sequences into 16 amino acids per local patch provides significant advantages for protein structure modeling metrics, such as TM-score and RMSD. In contrast, using a larger patchify size, like dividing protein sequences into 64 amino acids per local patch, has a more substantial impact on improving sequence-related metrics like Fident. More result analysis showed in A.6

6 Conclusion and Future Work

While current models can only perform unconditional sequence generation, TaxDiff overcomes this limitation by learning taxonomic-guided over protein sequences space. By combining the Global-Attention to sequences and Local-Attention to amino acids, TaxDiff can effectively generate sequences that are structurally reliable and consistent in sequence. Furthermore, TaxDiff requires

only a quarter of the time needed by other diffusion-based models to achieve superior performance. Various experimental results demonstrate its significantly better ability for modeling natural protein sequences. As a versatile and principled framework, TaxDiff can be expanded for various protein generation applications in future work. However, one limitation of TaxDiff is its current inability to scale effectively to the more challenging task of generating protein complexes. At the same time, proteins with high evaluation scores may be difficult to synthesize in practical experiments.

References

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[2] Zhenqiao Song and Lei Li. Importance weighted expectation-maximization for protein sequence design. *arXiv preprint arXiv:2305.00386*, 2023.

[3] Zaixiang Zheng, Yifan Deng, Dongyu Xue, Yi Zhou, Fei Ye, and Quanquan Gu. Structure-informed language models are protein designers. *bioRxiv*, pages 2023–02, 2023.

[4] Donatas Repecka, Vykintas Jauniskis, Laurynas Karpus, Elzbieta Rembeza, Irmantas Rokaitis, Jan Zrimec, Simona Poviloniene, Audrius Laurynenas, Sandra Viknander, Wissam Abuajwa, et al. Expanding functional protein sequence spaces using generative adversarial networks. *Nature Machine Intelligence*, 3(4):324–333, 2021.

[5] Richard J Fox, S Christopher Davis, Emily C Mundorff, Lisa M Newman, Vesna Gavrilovic, Steven K Ma, Loleta M Chung, Charlene Ching, Sarena Tam, Sheela Muley, et al. Improving catalytic function by prosar-driven enzyme evolution. *Nature biotechnology*, 25(3):338–344, 2007.

[6] Philip A Romero and Frances H Arnold. Exploring protein fitness landscapes by directed evolution. *Nature reviews Molecular cell biology*, 10(12):866–876, 2009.

[7] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, 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*, 2022:500902, 2022.

[8] Dominic Phillips, Hans-Christof Gasser, Sebestyén Kamp, Aleksander Pałkowski, Lukasz Rabalski, Diego A Oyarzún, Ajitha Rajan, and Javier Antonio Alfaro. Generating immune-aware sars-cov-2 spike proteins for universal vaccine design. In *Workshop on Healthcare AI and COVID-19*, pages 100–116. PMLR, 2022.

[9] Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-n protein engineering with data-efficient deep learning. *Nature methods*, 18(4):389–396, 2021.

[10] Zheng Tan, Yan Li, Xin Wu, Ziying Zhang, Weimei Shi, Shiqing Yang, and Wanli Zhang. De novo creation of fluorescent molecules via adversarial generative modeling. *RSC advances*, 13(2):1031–1040, 2023.

[11] Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex Xijie Lu, Nicolo Fusi, Ava Pardis Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, pages 2023–09, 2023.

[12] Alex Hawkins-Hooker, Florence Depardieu, Sebastien Baur, Guillaume Couairon, Arthur Chen, and David Bikard. Generating functional protein variants with variational autoencoders. *PLoS computational biology*, 17(2):e1008736, 2021.

[13] Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, and Yonghong Tian. Prollama: A protein large language model for multi-task protein language processing. *arXiv e-prints*, pages arXiv–2402, 2024.

[14] Kevin E Wu, Kevin K Yang, Rianne van den Berg, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *arXiv preprint arXiv:2209.15611*, 2022.

[15] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.

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

[17] Jung-Eun Shin, Adam J Riesselman, Aaron W Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew C Kruse, and Debora S Marks. Protein design and variant prediction using autoregressive generative models. *Nature communications*, 12(1):2403, 2021.

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

[19] 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.

[20] Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2023.

[21] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. *bioRxiv*, 2022.

[22] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[23] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[24] 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.

[25] Peng Jin, Hao Li, Zesen Cheng, Kehan Li, Xiangyang Ji, Chang Liu, Li Yuan, and Jie Chen. Diffusionret: Generative text-video retrieval with diffusion model. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2470–2481, 2023.

[26] Shitong Luo, Yufeng Su, Xingang Peng, Sheng Wang, Jian Peng, and Jianzhu Ma. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Advances in Neural Information Processing Systems*, 35:9754–9767, 2022.

[27] Brian L Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv:2206.04119*, 2022.

[28] Jason Yim, Brian L Trippe, Valentin De Bortoli, Emile Mathieu, Arnaud Doucet, Regina Barzilay, and Tommi Jaakkola. Se (3) diffusion model with application to protein backbone generation. *arXiv preprint arXiv:2302.02277*, 2023.

[29] Meng Liu, Youzhi Luo, Kanji Uchino, Koji Maruhashi, and Shuiwang Ji. Generating 3d molecules for target protein binding. *arXiv preprint arXiv:2204.09410*, 2022.

[30] Minkyung Baek, Frank DiMaio, Ivan Anishchenko, Justas Dauparas, Sergey Ovchinnikov, Gyu Rie Lee, Jue Wang, Qian Cong, Lisa N Kinch, R Dustin Schaeffer, et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557):871–876, 2021.

[31] John B Ingraham, Max Baranov, Zak Costello, Karl W Barber, Wujie Wang, Ahmed Ismail, Vincent Frappier, Dana M Lord, Christopher Ng-Thow-Hing, Erik R Van Vlack, et al. Illuminating protein space with a programmable generative model. *Nature*, pages 1–9, 2023.

[32] Yeqing Lin and Mohammed AlQuraishi. Generating novel, designable, and diverse protein structures by equivariantly diffusing oriented residue clouds. *arXiv preprint arXiv:2301.12485*, 2023.

[33] Raphael R Eguchi, Christian A Choe, and Po-Ssu Huang. Ig-vae: Generative modeling of protein structure by direct 3d coordinate generation. *PLoS computational biology*, 18(6):e1010271, 2022.

[34] Namrata Anand and Possu Huang. Generative modeling for protein structures. *Advances in neural information processing systems*, 31, 2018.

[35] Zeming Lin, Tom Sercu, Yann LeCun, and Alexander Rives. Deep generative models create new and diverse protein structures. In *Machine Learning for Structural Biology Workshop, NeurIPS*, 2021.

[36] Zachary Wu, Kevin K Yang, Michael J Liszka, Alycia Lee, Alina Batzilla, David Wernick, David P Weiner, and Frances H Arnold. Signal peptides generated by attention-based neural networks. *ACS Synthetic Biology*, 9(8):2154–2161, 2020.

[37] Yue Cao, Payel Das, Vijil Chenthamarakshan, Pin-Yu Chen, Igor Melnyk, and Yang Shen. Fold2seq: A joint sequence (1d)-fold (3d) embedding-based generative model for protein design. In *International Conference on Machine Learning*, pages 1261–1271. PMLR, 2021.

[38] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[39] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[40] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. *arXiv preprint arXiv:2309.11497*, 2023.

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

[42] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[43] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[44] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

[45] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

[46] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[47] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

[48] James J Luby. Taxonomic classification and brief history. In *Apples: botany, production and uses*, pages 1–14. Cabi Publishing Wallingford UK, 2003.

[49] Qinzhong Tian, Pinglu Zhang, Yixiao Zhai, Yansu Wang, and Quan Zou. Application and comparison of machine learning and database-based methods in taxonomic classification of high-throughput sequencing data. *Genome Biology and Evolution*, 16(5):evae102, 2024.

[50] David L Wheeler, Tanya Barrett, Dennis A Benson, Stephen H Bryant, Kathi Canese, Vyacheslav Chetvernin, Deanna M Church, Michael DiCuccio, Ron Edgar, Scott Federhen, et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 35(suppl_1):D5–D12, 2007.

[51] Yaghoub Safdari, Safar Farajnia, Mohammad Asgharzadeh, and Masoumeh Khalili. Antibody humanization methods–a review and update. *Biotechnology and Genetic Engineering Reviews*, 29(2):175–186, 2013.

[52] Andy Chi-Lung Lee, Janelle Louise Harris, Kum Kum Khanna, and Ji-Hong Hong. A comprehensive review on current advances in peptide drug development and design. *International journal of molecular sciences*, 20(10):2383, 2019.

[53] Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with foldseek. *Nature Biotechnology*, pages 1–4, 2023.

[54] Helen M Berman, Tammy Battistuz, Talapady N Bhat, Wolfgang F Bluhm, Philip E Bourne, Kyle Burkhardt, Zukang Feng, Gary L Gilliland, Lisa Iype, Shri Jain, et al. The protein data bank. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):899–907, 2002.

[55] Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1):D439–D444, 2022.

[56] Kevin K Yang, Nicolo Fusi, and Alex X Lu. Convolutions are competitive with transformers for protein sequence pretraining. *bioRxiv*, pages 2022–05, 2022.

[57] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR, 2021.

[58] 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.

[59] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

[60] Yang Zhang and Jeffrey Skolnick. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics*, 57(4):702–710, 2004.

[61] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[62] Brian L Hie, Varun R Shanker, Duo Xu, Theodora UJ Bruun, Payton A Weidenbacher, Shaogeng Tang, Wesley Wu, John E Pak, and Peter S Kim. Efficient evolution of human antibodies from general protein language models. *Nature Biotechnology*, 42(2):275–283, 2024.

[63] Walter Pirovano and Jaap Heringa. Protein secondary structure prediction. *Data Mining Techniques for the Life Sciences*, pages 327–348, 2010.

[64] Thomas E Creighton. Disulfide bond formation in proteins. In *Methods in enzymology*, volume 107, pages 305–329. Elsevier, 1984.

A Appendix / supplemental material

A.1 Result and Visualization

To demonstrate the applicability of our model under different levels of taxonomic guidance, we evaluated the model's ability to generate protein sequences guided by the three predominant domains within cellular organisms: Bacteria (63%), Eukaryota (33%), and Archaea (3%). The evaluation metrics were consistent with those used in the main experiments, and the specific details of these metrics can be found in the appendix A.3. More details of refined classification results will be available in supplementary material.

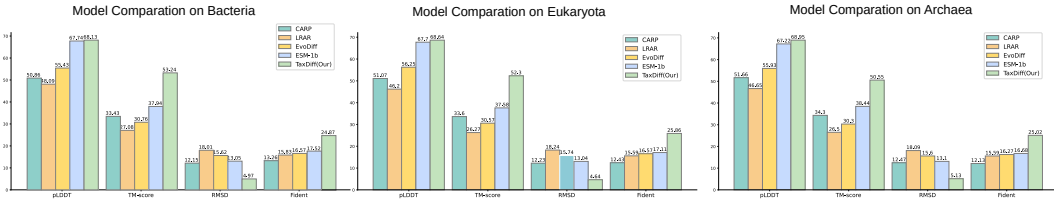

Figure 6: **Taxonomic distribution and comparison of taxonomic-guided models.** we compare the performance of different models under the condition of the second taxonomic level within cellular organisms: Bacteria (63%), Eukaryota (33%), and Archaea (3%).

A.2 Experiment Setting

Following common practices in generative modeling literature, we maintained an exponential moving average (EMA) of the TaxDiff weights during training, with a decay factor of 0.9999. The same training hyperparameters were applied across all TaxDiff models and patch sizes. All reported results are based on the EMA model. The same training hyperparameters were applied across all TaxDiff model sizes and patch sizes, almost entirely retained from ADM, without adjustments in learning rate, decay/warm-up schedule, Adam $\beta_1, \beta_2, \beta_3$, or weight decay.

In order to verify the scaling ability of Taxdiff, we extend the original Taxdiff (layer=12) to layer=6 (Small) and layer=18(Large) in the DiT model scaling method in fellow. In addition, when the sample number was expanded to 1,000, the experimental results in table 4 showed that our TaxDiff(layer=12) was the best, especially in TM-score, RMSD and Fident.

Table 4: **Comparative experiments result Taxdiff layers** on AFDB and PDB datasets. Metrics are calculated with 1,000 samples generated from each model.

Method	**pLDDT↑**	**AFDB Dataset**			**PDB Dataset**		
		TM-score↑	**RMSD↓**	**Fident(%)↑**	**TM-score↑**	**RMSD↓**	**Fident(%)↑**
Small	68.8507	43.82	6.7151	24.84	45.90	5.4525	24.36
Large	68.1949	45.62	6.4333	24.53	45.77	5.9830	23.70
TaxDiff	69.0000	49.27	5.6518	25.02	48.80	4.8453	24.85

A.3 Evaluation

We measure model performance through sequence consistency and structural foldability, which indicates whether the model can learn the rules of protein sequences from the data. For assessing the feasibility of the sequences, We employed OmegaFold [21] to predict their corresponding structures and calculate the average predicted Local Distance Difference Test (pLDDT) across the entire structure, which reflects OmegaFold's confidence in its structure prediction for each residue on sequences level. The pLDDT is calculated by a function based on the difference between the inter-residue distances predicted by the model and the true distance. It has values ranging from 0 to 100 and indicates the confidence level of each residue in the prediction which was first used in AlphaFold2 [55]. Omegafold performs structure prediction without the need for homologous sequences or evolutionary information, relying solely on a single sequence for prediction. However, due to inherent noise and errors in OmegaFold's structure predictions, which only consider the foldability of individual sequences, we further measure our results using Foldseek [53]. Foldseek facilitates the pairing of the queried protein p^{query} with structurally similar proteins from an existing protein database, yielding pairs represented as $(p^{query},\ p^{target})$. Here, p^{target} denotes the protein in the database with a significant structural similarity to p^{query}. The magnitude of the average template modeling score (TM-score [60]) value and root-mean-square deviation (RMSD) reflects the degree of structural similarity. TM-score takes into account the overall topological structure of proteins, focusing more on the protein's overall structure. RMSD calculates the square root of the average position deviation of corresponding atoms between two protein structures, being highly sensitive to the size of the protein structure and local variations. Additionally, Foldseek also calculates the sequence identity (Fident) between p^{query} and p^{target}, reflecting their sequence-level similarity. In Foldseek, we choose natural protein structures from the Protein Data Bank (PDB) [54] to verify the natural-like degree of our sequences and high-confidence protein structures predicted by Alphafold (AFDB) [1] to expand the comparison range and verify the broad validity of our model.

TM-score is a quantitative measure used to assess the similarity between two protein structures. It is defined as:

$$\text{TM-score} = \max\left(\frac{1}{L}\sum_{i=1}^{L}\frac{1}{1+\left(\frac{d_i}{d_0(L)}\right)^2}\right) \tag{13}$$

where L represents the length of the target protein, d_i denotes the distance between the ith pair of aligned residues, and $d_0(L)$is a length-dependent scale for normalization. The TM-score ranges

from 0 to 1, with values closer to 1 indicating a higher degree of structural similarity. Typically, a TM-score greater than 0.5 suggests that two protein structures share the same fold.

RMSD is a standard measure used to quantify the average distance between atoms (usually backbone atoms) of superimposed proteins. The RMSD is calculated as:

$$\text{RMSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - r_i')^2} \tag{14}$$

where N is the number of aligned atoms, r_i represents the position vector of the ith atom in the reference structure, and r_i' is the corresponding position in the model structure. RMSD provides a measure of the structural deviation between two protein conformations, with lower values indicating higher similarity.

Fident quantifies the proportion of residues that are identical between the sequences, providing insight into the degree of homology. The Fident value is calculated as follows:

$$\text{Fident} = \frac{\text{Number of identical residues}}{\text{Total number of aligned residues}} \times 100\% \tag{15}$$

where the numerator represents the number of residues that are identical between the aligned sequences, and the denominator is the total number of residues that have been aligned.

A.4 Baselines

We compare TaxDiff to several competitive baseline models. The left-to-right autoregressive (LRAR) model and convolutional autoencoding representations of proteins(CARP) [56] were both trained using the same dilated convolutional neural network architectures on the UniRef50 dataset. For LRAR, a causal mask is applied to the convolutional modules to prevent information leakage. FoldingDiff [14] and RFdiffusion [15] are recent progress on diffusion models for protein structure generation. Notably, the RFdiffusion and FoldingDiff directly produce protein structures; hence, we first unconditionally select structures generated by these two and then use ESM-IF to design their corresponding sequences. EvoDiff [11] is a diffusion model that leverages evolutionary-scale data based on programmable, sequence-first design. The results reported in our result are based on the config640M configuration. ESM-1b [57] and state-of-the-art ESM-2 [58] are the protein masked language models, which were trained on different releases of UniRef50. Specifically, ESM-2 version is ESM2-t33-650M-UR50D and ESM1b is ESM1b-t33-650M-UR50S. Additional details about the ESM models are provided in Section A.5. ProtGPT2 [59] and ProGen2 [19] are autoregressive large protein language models based on GPT2 [61], which have undergone pre-training in UniRef50 and UniRef90 respectively. ProGen2 refers specifically to ProGen2-large.

A.5 ESM Model

Using ESM models to predict sequences is not common. However, there are already some applications in EvoDiff [11] and human antibodies [62] due to the strong protein sequence understanding capability of the ESM method. Recognizing the importance of clarity in explaining non-standard methodologies, we have included the relevant content in our work to elucidate our approach. To be specific, the process begins with a series of `<mask>_seqlen` tokens as input to simulate a completely unknown sequence of the desired length. This method allows the model to iteratively predict the missing amino acids at each position. To ensure the biological relevance of the generated sequences, we implement penalties on specific tokens (e.g., X, pad, eos, and cls) to prevent their generation, addressing the unique considerations necessary for protein sequence design. The sequence generation is facilitated by employing `torch.multinomial` to perform random sampling based on the predicted probability distribution of amino acids, thus ensuring variability and fidelity to plausible protein sequences. Finally, the tokens are converted back into amino acid sequences, culminating in the designed protein sequences.

A.6 Result Analysis

It is worth noting that the performance of $patchify - size = 32$ is inferior to that of 16 and 64. We attribute this performance disparity to several biologically relevant factors concerning protein structure and function, as outlined below:

Secondary Structure Formation [63]: Specific length sequences favor forming essential secondary structures such as α-helices and β-sheets, which are crucial for protein function and can significantly influence biological activity. A 32 amino acid length might not support these structures well, affecting performance.

Functional Domain Integrity: The function of proteins often relies on specific functional domains, which require a certain amino acid sequence length to maintain their structural and functional integrity. A partition length of 32 amino acids may inadvertently disrupt these critical functional domains, resulting in deactivation or diminished function.

Hydrogen and Disulfide Bond Formation [64]: Hydrogen and disulfide bonds are pivotal for the stability of protein tertiary structures. The length of 32 amino acid sequences may influence the formation of these bonds, especially at sequence partition boundaries.

A.7 Ablation Result

Table 5: **Ablation result of different attention combination** on AFDB and PDB datasets. Metrics are calculated with 1000 samples generated from each model.

Method	**pLDDT↑**	**AFDB Dataset**			**PDB Dataset**		
		TM-score↑	**RMSD↓**	**Fident(%)↑**	**TM-score↑**	**RMSD↓**	**Fident(%)↑**
E	68.77 ± 10.67	44.84	7.2685	23.43	44.00	5.9162	24.1
D	61.50 ± 13.55	39.65	9.2808	20.33	39.30	7.6774	19.53
C	61.57 ± 12.99	39.55	9.6091	19.87	40.21	7.6458	20.12
B	61.62 ± 12.99	39.36	9.1560	19.95	40.52	7.551	19.77
A	**69.01 ± 9.03**	**48.88**	**5.4992**	**25.88**	**49.58**	**4.6262**	**24.95**