TOWARD *De Novo* PROTEIN DESIGN FROM NATURAL LANGUAGE

Fengyuan Dai[1], Yuliang Fan[1], Jin Su[1], Chentong Wang[1], Chenchen Han[1], Xibin Zhou[1], Jianming Liu[1], Hui Qian[1], Shunzhi Wang[2], Anping Zeng[1], Yajie Wang[1] and Fajie Yuan[1*]

[1]Westlake University, [2]University of Washington

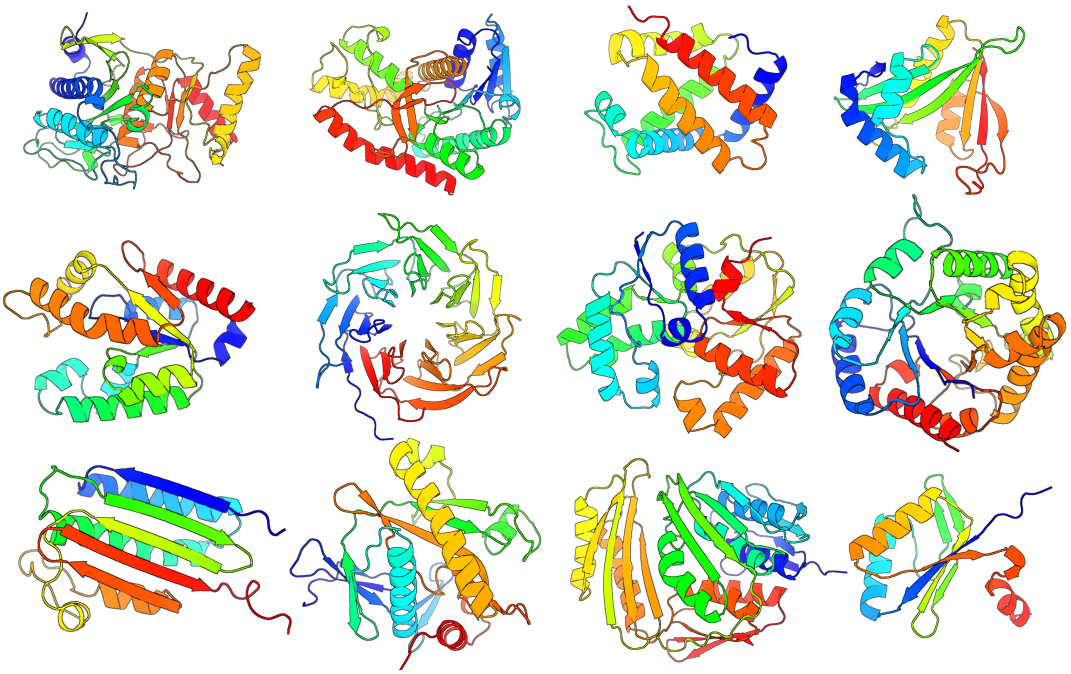

Figure 1: **Showcase of designed proteins.** Refer to Appendix A for a complete list of prompts. Pinal can generate diverse proteins from both short or long protein language descriptions.

ABSTRACT

De novo protein design represents a fundamental pursuit in protein engineering, yet current deep learning approaches remain constrained by their narrow design scope. Here we present Pinal, a large-scale frontier framework comprising 16 billion parameters and trained on 1.7 billion protein-text pairs, that bridges natural language understanding with protein design space, translating human design intent into novel protein sequences. Instead of a straightforward end-to-end text-to-sequence generation, Pinal implements a two-stage process: first generating protein structures based on language instructions, then designing sequences conditioned on both the generated structure and the language input. This strategy effectively constrains the search space by operating in the more tractable structural domain. Through comprehensive experiments, we demonstrate that Pinal achieves superior performance compared to existing approaches, including the concurrent work ESM3, while exhibiting robust generalization to novel protein structures beyond the PDB database. The online demo is available at `http://www.denovo-pinal.com/`.

*Corresponding author: yuanfajie@westlake.edu.cn.

1 INTRODUCTION

Proteins are fundamental to life, playing critical roles in biological processes across all living organisms. Protein design aims to customize proteins for specific biological or biomedical purposes. Traditional protein design methods (Dougherty & Arnold, 2009; Accuracy, 2003), while effective, are often limited by their reliance on existing protein templates and natural evolutionary constraints. In contrast, *de novo* design (Huang et al., 2016) benefits from the two perspectives. Firstly, nature has only explored a small subset of the possible protein landscape. Secondly, the biological attributes selected by evolution may not align with our specific functional requirements. *De novo* design allows us to create entirely new proteins with desirable structures and functions, thus overcoming the limitations of traditional methods.

Although *de novo* protein design (DNPD) using deep learning (Watson et al., 2023; Ingraham et al., 2023; Krishna et al., 2024) has gained considerable attention, current methods often operate under rather limited conditions. These methods typically focus on either unconditional design (Lin & AlQuraishi, 2023) or on specific functions such as conditioning on control tags, motif scaffolding, and binder design (Madani et al., 2023; Watson et al., 2023). Given the versatile functions and biological significance of proteins, these approaches provide only a limited view of the target protein and may not fully capture their complexity and diversity. To this end, we propose a more ambitious and general approach: designing *de novo* proteins from natural language. This method leverages the descriptive power and flexibility of natural language to accurately communicate design objectives and functionality requirements to the protein generator.

Protein molecules exhibit a profound relationship between their structure and function (Dill & MacCallum, 2012). Inspired by traditional physics-based approaches (Cao et al., 2022), we propose an intuitive method: rather than designing protein sequences directly from natural language descriptions of function, we first translate these descriptions into structural information and then generate sequences conditioned on both the language description and its structure. To achieve this, we first employ an encoder-decoder architecture named T2struct, which is designed to interpret natural language and derive structural information from it. Instead of dealing with explicit 3D structures, we use discrete structure tokens generated by the vector quantization technique (van Kempen et al., 2022), which have been shown to offer better scalability for larger datasets (Su et al., 2024b; Hayes et al., 2024). Subsequently, we modify and retrain SaProt (Su et al., 2024a), a structure-aware protein language model, referred to as SaProt-T, to understand natural language inputs and enable sequence design based on the given backbone and language instructions. This pipeline provides an effective pathway to map natural language to protein sequences. It ensures that the designed proteins accurately express the desired functions and exhibit robust foldability.

Scaling laws have been well-established in various fields through the expansion of model parameters and datasets (Kaplan et al., 2020; Brown et al., 2020). However, their empirical evidence in text-to-protein design remains largely unexplored. In this paper, we explicitly show that increasing both model parameters and dataset size substantially improves the model's capabilities. A unique challenge in the protein design domain is the scarcity of high-quality textual descriptions for proteins, in contrast to other fields that benefit from billions of data points (Schuhmann et al., 2022). This limitation constrains our ability to scale training data to the same magnitude. To address this challenge, we leverage ProTrek (Su et al., 2024c), a highly accurate protein retrieval model, and large language models (LLMs) to construct the largest synthetic natural language-protein dataset to date, comprising 1.7 billion pair examples and 160 billion word tokens. This comprehensive dataset enables Pinal to effectively explore the protein design space through natural language guidance.

To summarize, this work makes the following key contributions:

- We demonstrate for the first time that direct end-to-end text-to-sequence mapping is inherently challenging due to the vast complexity of the protein sequence space. Our two-stage design approach leverages the more constrained structure space as an intermediate representation, substantially improving performance across multiple protein design metrics.

- We implement this insight into practice by developing Pinal, a novel deep learning-based framework for protein design. Pinal integrates two key components: T2struct for natural language to protein structure translation, and SaProt-T for structure and text co-guided

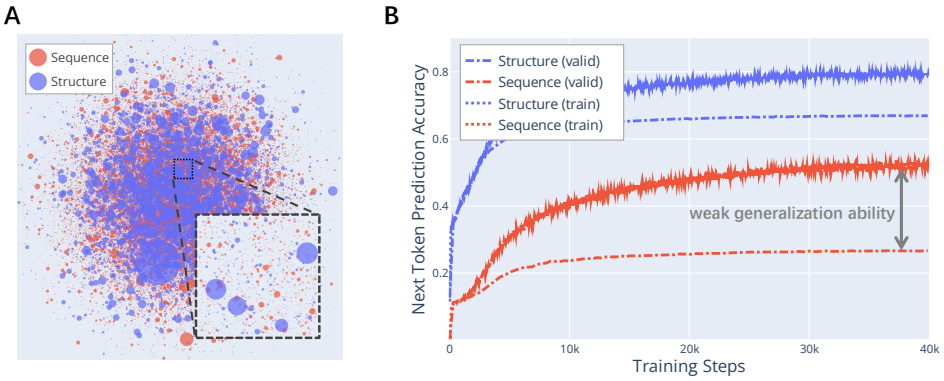

Figure 2: **Learning to generate structure is easier.** **A**: Visualization of protein sequence and structure space using Swiss-Prot database as an example. Each circle represents a cluster, and the size of the circle corresponds to the number of proteins within that cluster. 30% token identity is used for clustering, resulting in 39,322 structure clusters and 126,185 sequence clusters. Feasible structure space is much more smaller than sequence in discrete space. **B**: Next token prediction accuracy trend training on structure and sequence token. The training and validation sets are split by 50% sequence identity to ensure that similar proteins do not overlap between the two sets.

sequence generation. We further develop an optimal sampling strategy to rank proteins by combining these components.

- We demonstrate clear scaling benefits in Pinal's performance as we increase both model size and training data. To push these boundaries, we train our 16 billion parameter model on an unprecedented dataset comprising 1.7 billion natural language-protein pairs and 160 billion word tokens.
- Through extensive experimental validation under both concise and detailed instruction scenarios, we demonstrate that Pinal consistently outperforms existing methods, including the concurrent ESM3 (Hayes et al., 2024). Notably, Pinal exhibits strong generalization capabilities, successfully designing novel proteins beyond the structural distribution found in PDB.

2 Avoiding Designing Sequence Directly

The most intuitive approach for DNPD from natural language is to use an encoder-decoder architecture. In this setup, the encoder represents the natural language input, while the decoder generates the corresponding protein sequence. We refer to this method as end-to-end training. However, this approach can be challenging due to the vast expanse of the protein sequence space, which makes accurate sequence generation difficult. In contrast, protein structure is more conserved and intuitively much easier to predict and generate.

In Figure 2A, we visualize both the protein sequence and structure space of the Swiss-Prot database. The protein structure here is represented by discrete structural tokens via Foldseek(van Kempen et al., 2022), which has the same number of alphabet as that of amino acids, i.e., 20. As clearly shown, the protein sequence space is much larger and more diverse than the structure space.

In Figure 2B, We apply the same encoder-decoder architecture to train both a language-to-sequence model and a language-to-structure model, with protein structures still represented by Foldseek tokens. As expected, deep learning models that learn the language-to-sequence space find it more challenging to achieve ideal next-token prediction accuracy compared to the language-to-structure model in the validation set. This finding also explains why previous language-guided protein design models, such as ProteinDT (Liu et al., 2023), exhibit poorer performance, as discussed in Section 4.3.

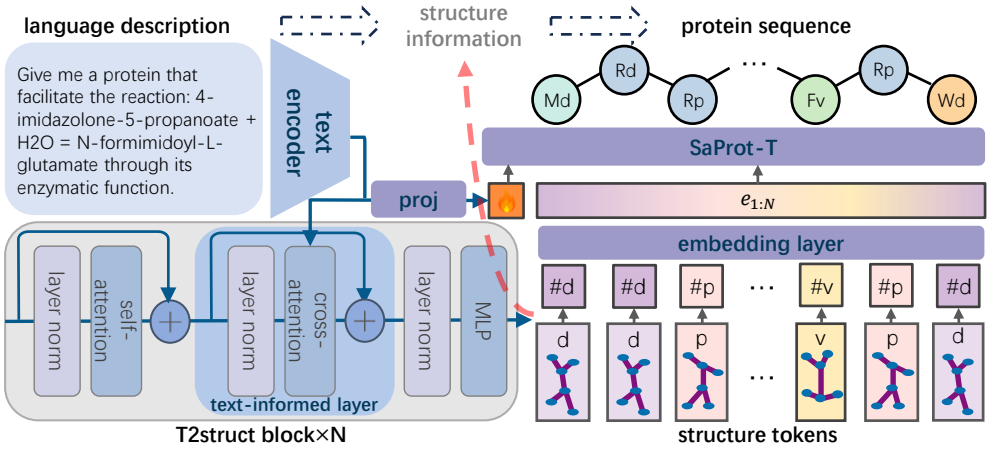

Figure 3: **Overview of Pinal.** The text encoder first encodes the provided protein description into textual embeddings. T2struct then uses an encoder-decoder architecture to predict discrete structural tokens from these embeddings. These structural tokens are subsequently concatenated with the projected textual embeddings and fed into SaProt-T to facilitate the design of a *de novo* protein sequence.

3 PINAL FOR LANGUAGE-GUIDED DNPD

3.1 PINAL FRAMEWORK

Motivated by the above analysis, we decompose the protein design process into two primary steps: first, translating natural language descriptions of protein into structures, and then generating protein sequences based on both the structure and language description. Specifically, as the structure is determined by sequence, we modify the probability of protein sequence s to the joint distribution of s and additional protein structure information c:

$$p(s \mid t) = p(s, c \mid t),$$

where t denotes the language descriptions. According to the Bayes' theorem, the above joint distribution can be formulated as:

$$p(s, c \mid t) = p(c \mid t)p(s \mid c, t). \tag{1}$$

On the right side of the equation, $p(c \mid t)$ indicates the probability of the structural information aligning with t, and $p(s \mid c, t)$ implies the probability of a sequence capable of folding into the depicted structure and expressing the desired function as specified by the given language description. In practice, we leverage T2struct to predict $p(c \mid t)$, and SaProt-T to predict $p(s \mid c, t)$, thereby generating protein sequences that account for both structural folding and functional expression, as illustrated in Figure 3.

3.2 DECODING STRUCTURAL INFORMATION FROM NATURAL LANGUAGE

Protein structures can be represented in various ways, including explicit 3D structures and discrete structural tokens (van Kempen et al., 2022; Lin et al., 2023; Gao et al., 2024). However, our Pinal framework is better suited to the latter approach, given its sampler design in Section 3.4. In this paper, we use 3Di token sequence to represent structures generated by Foldseek (van Kempen et al., 2022).

To represent $p(c \mid t)$, we model the conditional probability of structural tokens c in an auto-regressive manner:

$$p(c \mid t) = p_{\theta_1}(c_{1:N} \mid t) = \prod_{n=1}^{N} p_{\theta_1}(c_n \mid c_{<n}, t), \tag{2}$$

where θ_1 denotes the learned parameter. To generate structural tokens conditioned on text embeddings, we employ an encoder-decoder architecture (Vaswani et al., 2017). Specifically, we leverage

the pre-trained text encoder from PubMedBERT (Gu et al., 2021) (109M) as the language encoder. For the 3Di structural token decoder, we utilize a randomly initialized GPT-2 architecture (Radford et al., 2019) with 3Di token embeddings, enhanced by a text-informed layer in each block (114M). This text-informed layer incorporates layer normalization, a cross-attention mechanism, and a residual connection.

3.3 SEQUENCE GENERATION FROM STRUCTURAL AND NATURAL LANGUAGE CONDITION

Given a sequence of structural tokens $c_{1:N}$, SaProt (Su et al., 2024b) predicts the corresponding amino acid sequence. Specifically, each structural token c_i is paired with a masked amino acid $\#$, represented as $\#c_i$. SaProt takes structural sequence $(\#c_1, \#c_2, ..., \#c_N)$ as input and outputs the structure-aware sequence $(x_1, x_2, ..., x_N)$, where x_n denotes the combination of the amino acid and the structural token, *i.e.* $s_n c_n$. Although SaProt demonstrates impressive performance in predicting sequences based on structural tokens, its predicted sequences are not explicitly conditioned on textual descriptions.

To model $p(s \mid c, t)$, we re-train SaProt with text as additional input, referred to as SaProt-T. Given the textual embeddings after the pooling layer, $e_t \in \mathbb{R}^{1\times d_t}$, we project them using a trainable matrix $W \in \mathbb{R}^{d_t \times d_s}$, where d_t and d_s represent the embedding dimensions of the text encoder and SaProt-T, respectively. The resulting embeddings e_{input} are concatenated with embeddings of the structural token sequence $e_{1:N} \in \mathbb{R}^{N\times d_s}$:

$$e_{input} = [e_t \times W, e_{1:N}].$$

SaProt-T takes e_{input} as input and is trained to predict the masked amino acid at each position. Denoting θ_2 as the parameter of SaProt-T, we calculate the product of conditional probabilities over the length of the protein as follows:

$$p(s \mid c, t) = \prod_{n=1}^{N} p_{\theta_2}(s_n \mid c, t) = \prod_{n=1}^{N} p_{\theta_2}(x_n \mid e_{input}). \tag{3}$$

3.4 SAMPLING-BASED PROTEIN RANKER

Although designing protein backbone structure according to function and then designing sequence according to this structure is a common heuristic method in traditional protein design, this paper introduces a more rigorous mathematical interpretation from a Bayesian perspective (i.e., Eq.1). We derive an optimal sampling-based ranking scheme for the entire two-stage generation process, as outlined below.

$$\begin{aligned} \arg\max_s p(s \mid t) &= \arg\max_s \log p(s \mid t) = \arg\max_s (\log p(c \mid t) + \log p(s \mid c, t)) \\ &= \arg\max_s \sum_{n=1}^{N} \Big[\log p_{\theta_1}(c_n \mid c_{<n}, t) + \log p_{\theta_2}(s_n \mid c, t) \Big]. \end{aligned} \tag{4}$$

Note: $$\arg\max_s (\log p(c \mid t) + \log p(s \mid c, t)) \neq \arg\max_c \log p(c \mid t) + \arg\max_s \log p(s \mid c, t) \tag{5}$$

The above equation illustrates how the probability of a protein sequence can be precisely estimated using the Pinal pipeline by leveraging T2struct and SaProt-T. Instead of sequentially determining the most aligned structure and then the optimal sequence , as shown on the right side of Eq. 5—an approach that often leads to suboptimal outcomes (see Fig.6)—we consider the joint probability distribution of structure and sequence to achieve the optimal protein sequence generation by computing Eq. 4. In practice, given the challenge of exploring the entire range of s values, we limit our exploration to K sampled values. That is, we first sample K structural sequences using multinomial sampling via T2struct and then apply a greedy search to determine the corresponding protein sequence via SaProt-T. After that, we compute Eq. 4 for these K candidates and select the top candidates. We set K to 50 throughout this paper (see Section 4.4 for ablation).

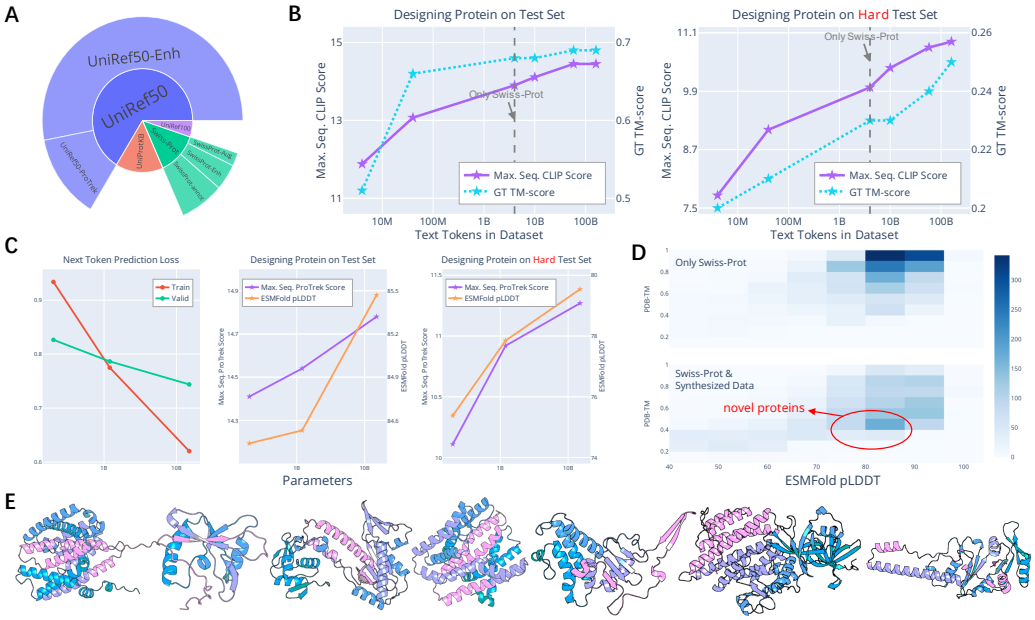

Figure 4: **Scaling model parameters and training data. A:** Distribution of training datasets. **B:** Protein design performance improves as training data size increases. **C:** Scaling behavior of T2struct from 200 million to 15 billion parameters; both test loss and the quality of designed proteins improve with larger parameter sizes. **D:** Density heatmaps illustrating the novelty and foldability of proteins designed by models trained on two different datasets. Training on Swiss-Prot and synthesized data enhances the model's capacity to design novel proteins. **E:** Visualizations of novel proteins designed by Pinal.

4 EXPERIMENTS

4.1 EXPERIMENT SETUP

Dataset. We begin with Swiss-Prot (Boeckmann et al., 2003), a manually annotated and reviewed protein sequence database that organizes protein annotations into sequence-level and residue-level records. To prepare this protein information for natural language processing, we utilize large language models (LLMs) to generate sentence templates and populate them with data extracted from these annotations. This process results in the dataset referred to as SwissProt-Annot (see Appendix B.1). Despite this, a significant gap remains between these sentence templates and the types of queries typically posed by biologists. To bridge this gap, we input data from SwissProt-Annot into LLMs to rewrite the protein descriptions (see Appendix B.2), thereby creating a dataset named SwissProt-Aug. SwissProt-Aug consists of 4 million natural language–protein pairs.

However, Swiss-Prot database only contains around 560 thousand protein sequences, representing a small portion of the vast protein design space, especially when compared to the billions of natural proteins that have been discovered. In contrast, the UniRef50 database contains two orders of magnitude more proteins but lacks high accurate annotations. We then utilize ProTrek to annotate these proteins, resulting in 400 million natural language-protein pairs. Specifically, for each protein, we search for 10 descriptions by ProTrek, which yields the dataset referred to as UniRef50-ProTrek. Additionally, we incorporate training data from Zhou et al. (2025), known as SwissProt-Enh and UniRef50-Enh. In these datasets, the prompts for LLMs are constructed by combining the information extracted through ProTrek with questions targeting specific aspects of the proteins. LLMs are then tasked with generating captions for proteins based on these prompts, culminating in a total of 9 million natural language-protein pairs from Swiss-Prot and 530 million from UniRef50. This curated dataset significantly enriches the availability of annotated protein sequences and enhances the model's ability to understand and generate relevant protein descriptions.

In addition, we collect functional labels for proteins from UniProtKB, using annotations provided by InterProScan (Paysan-Lafosse et al., 2023), as well as data from proteins without annotations in the AlphaFold Database (Varadi et al., 2024). The annotations from InterProScan offer valuable information regarding protein families, predicted domains, and important functional sites, resulting in a total of 791 million keyword-protein pairs. We intentionally leave these keywords as natural language inputs, without the aid of LLMs, to enhance the model's capability to identify and leverage these keywords effectively. For the proteins lacking specific descriptions, we prompt the model with the indication of "random protein", which facilitate exploration of model in the design space even in the absence of specific textual descriptions. In total, the dataset (Fig. 4A) consists of more than 1 billion natural language-protein pairs, almost 160 billion text tokens and 60 billion amino acid tokens. Details are provided in Appendix B.

We perform clustering on Swiss-Prot and split the proteins into training, validation, and testing sets based on 50% sequence identity, resulting in 114246 clusters. For validation, we allocate 1142 clusters (5165 proteins), while an additional 572 clusters (3304 proteins) are set aside for testing. The remaining data, including the synthesized datasets, constitute the training set.

Metrics. We evaluate proteins from two perspectives: foldability and language alignment. **For foldability**, we analyze the designed sequences using the single-sequence structure prediction model, *i.e*. ESMFold (Lin et al., 2022). We compute the average predicted local distance difference test (`pLDDT`) (Wang et al., 2024c) and predicted aligned error (`PAE`) across the whole structure according to the output of ESMFold. It's important to note that `pLDDT` above 70 and `PAE` below 10 are commonly used thresholds indicating high prediction confidence. These thresholds suggest a high probability that the predicted protein sequences can fold into the corresponding predicted structures. **For language alignment**, we measure the structural similarity between the ESMFold-predicted structure of the designed sequence and the ground truth using TMscore (Zhang & Skolnick, 2004) (`GT-TMscore`). However, considering that proteins with the same functional descriptions may not necessarily fold into similar structures, we use ProTrek (Su et al., 2024c), a tri-modal protein language model, to evaluate textual alignment from both sequence and structural perspectives. Similar evaluation methods are widely used in the computer vision field (Hessel et al., 2021). Specifically, given a functional descriptions t and a protein p (which can be either protein structure or sequence), we calculate the similarity score, termed as `ProTrek score`, as $s = cos(\tau_t(t), \tau_p(p))/t$, where $\tau_t(t)$, $\tau_p(p)$ and t represent the text encoder, protein encoder, and the temperature from ProTrek. We refer to the similarity score between an amino acid sequence and t as the `Seq. ProTrek score`, while the similarity score between a protein structure and t is designated as the `Struc. ProTrek score`. **For novelty**, we assessed the novelty of proteins by comparing them to known PDB structures using the TM-score metric. The highest TM-score for each sequence, referred to as `PDB-TM`, served as an indicator of novelty, with sequences exhibiting `PDB-TM` below 0.5 classified as novel proteins.

4.2 Scaling Training Data and Model Parameters

Scaling laws have been widely studied in natural language processing (Kaplan et al., 2020) and AI for life sciences (Chen et al., 2024). They describe a power-law relationship between the scale of model parameters (or datasets, computation, etc.) and the loss value on the test set. However, to the best of our knowledge, there is no evidence that scalability occurs when training purely on text-to-protein data. To address this gap, we train T2struct at three different scales: 200 million, 1.2 billion, and 15 billion parameters. Furthermore, we increase the number of word tokens in the training set from 4 million to 160 billion. Our results indicate that enlarging both the model size and the dataset yields significant improvements in performance.

Performance improves with an expansion of the training dataset, including when utilizing synthesized data (Figure 4B). To investigate the influence of enlarging training data, we train 1.2B T2struct on different scale of datasets. Specifically, augmenting the dataset with Swiss-Prot entries results in a substantial enhancement on the test set (the details of constructing input of test set is illustrated in Section 4.3). In contrast, incorporating data from sources other than Swiss-Prot yields only marginal gains (Figure 4B left). We hypothesize that the test set, based on 50% sequence identity, is a relatively straightforward task, and leveraging data from Swiss-Prot is sufficient to achieve a performance plateau, demonstrating diminishing returns with further data diversification. To further demonstrate the effectiveness of scaling training data, we assembled a collection of protein descrip-

tions with less than 30% sequence identity to any sequences in the training set, designated as the hard test set. Although the overall performance drops compared to the standard test set, the quality of designed proteins improves when a large scale of synthetic data is integrated (Figure 4B right), highlighting the importance of scaling training data, particularly for more challenging tasks.

The design space of Pinal surpasses PDB by training on synthesized data. The Protein Data Bank (PDB) archives most of the protein 3D structure data. Biologists are more interested in finding proteins that are structurally distinct from known proteins. However, since proteins from Swiss-Prot are well studied, most proteins overlap with PDB data. Training solely on Swiss-Prot confines model to a compact and PDB-like design space, thereby inhibiting the generation of novel proteins. We believe that synthetic data, capturing proteins from UniProt, enables the model to explore a broader and more diverse feasible protein landscape, despite these structures are predicted by a neural network (Jumper et al., 2021). To verified this, we first select 500 proteins from UniRef50, which are structurally different from PDB, and take the corresponding descriptions from UniRef50-QA. We ask Pinal to design 5 proteins for each description, resulting 2500 proteins (Figure 4D).

Our findings reveal that models trained exclusively on Swiss-Prot data tend to produce proteins with higher foldability that also exhibit high `PDB-TM`, indicating similarity to existing PDB structures. In contrast, the model trained on synthesized data did not display this bias towards high `PDB-TM`. Specifically, out of the 2,500 designed proteins generated by the Swiss-Prot-trained model, 164 were identified as novel with high foldability. Conversely, the model trained on synthesized data produced 474 novel proteins, demonstrating a substantial improvement in exploring novel structural spaces (Figure 4E). These results underscore the significant advantage of leveraging synthesized data to expand the design space beyond the confines of the PDB.

Scaling the parameters of T2struc yeilds substantial enhancements in next token prediction loss and design capability (Figure 4C). We scale the T2struct model to three distinct parameter sizes: 200 million, 1.2 billion, and 15 billion. This parameter expansion resulted in marked reductions in both training and validation loss. To further validate these improvements, we employed the models to design proteins using two separate test sets. The proteins generated by the larger models exhibit markedly higher foldability and demonstrate superior alignment with given natural language instructions.

4.3 Comparing with Other Text Guided Protein Design Methods

Baselines. We compare Pinal with three public available methods that support protein design with textual description, namely ProteinDT (Liu et al., 2023) and Chroma (Ingraham et al., 2023) and ESM3 (Hayes et al., 2024). ProteinDT is a multi-modal framework that integrates text with protein design, employing the architecture of DALL·E 2 (Ramesh et al., 2022). We use its auto-regressive version[1], which has been reported to exhibit better performance. Chroma is a diffusion model that can directly sample protein structures and sequences guided by a pre-trained protein captioning model. ESM3 is a frontier model trained with functions, protein sequences, and structures simultaneously. Note that ESM3 can only support keywords rather than detailed text descriptions.

Pinal outperforms ProteinDT and Chroma in both long and short sentence settings. We evaluate the protein design ability from natural language input from two perspectives: long sentences and short sentences. The long sentence setting is more practical for biologists, as it allows for diverse and elaborate descriptions that capture multiple aspects of the proteins of interest, leading to fine-grained, controllable protein generation. In contrast, the short sentence setting provides concise but definitive instructions, validating the model's basic ability to classify and generate individual elements. To investigate the ability to design *de novo* proteins from long descriptions, the input is constructed as follows: for each protein in the test set, we gather all of its descriptions, which portray the target protein from various perspectives, and concatenate them into a single sentence. We then randomly select 500 target proteins from the dataset and ask the model to design 5 proteins per sentence, resulting in 500 long descriptions and 2500 designed proteins. To test the robustness of the model to textual input and to facilitate fair comparison with previous methods trained on diverse protein descriptions, the textual inputs are filled into templates different from those used in training or paraphrased by GPT-4.

[1]ProtBERT_BFD-512-1e-5-1e-1-text-512-1e-5-1e-1-EBM_NCE-0.1-batch-9-gpu-8-epoch-5

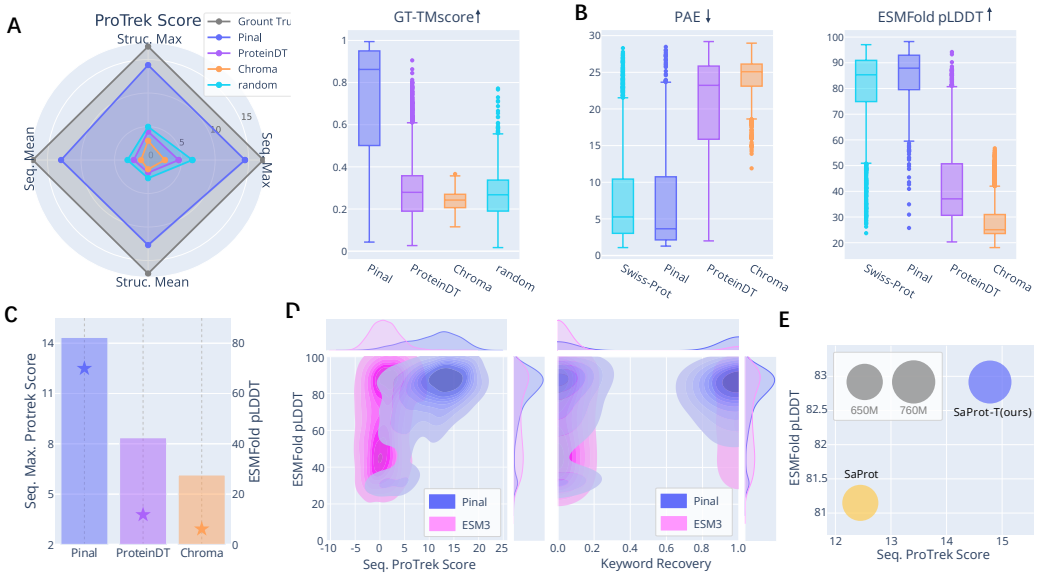

Figure 5: **Assessment of designed protein from language descriptions. A**: Alignment of designed proteins with given language descriptions, assessed by `ProTrek Score` and `GT-TMscore`. **B**: ESMFold evaluates the foldability of designed proteins. **C**: The foldability of designed proteins evaluated by ESMFold. The bar represents `pLDDT` and the star represents `ProTrek score`. **D**: Comparing with ESM3. In both panels, deeper colors in the upper right corner indicate stronger model performance. **E**: Sequence design comparison based on structural tokens.

In Figure 5A, we visualize the assessment of generated proteins from the perspective of textual alignment. The `ProTrek score` is calculated to measure the similarity between the natural language and protein sequence, as well as between natural language and protein structure. All calculated structures are predicted by ESMFold. For each modality (sequence and structure), we report scores in two ways: by taking the maximum and mean scores among the 5 generated proteins from each textual input. Additionally, we calculate the `ProTrek score` and `GT-TMscore` between descriptions and randomly selected proteins from Swiss-Prot for further illustration. We observe that ProteinDT and Chroma can struggle to design proteins that align well with the given text, as their `ProTrek score` and `GT-TMscore` show no significant difference compared to arbitrarily selected proteins from Swiss-Prot.

We can draw a similar conclusion regarding foldability (Figure 5B). Proteins designed by Pinal exhibit `PAE` and `pLDDT` values comparable to those of proteins from Swiss-Prot, indicating satisfactory structural plausibility. Lastly, sequences from Chroma and ProteinDT often fail to fold into 3D structures, as indicated by their high `PAE` and low `pLDDT` values.

We further examine the ability to generate protein sequences conditioned on short descriptions, defined as single sentence from individual protein function categories. For each protein function category, we randomly sample 50 sentences from the dataset. Evaluation is conducted by calculating the maximum `ProTrek score` among the 5 sequences and averaging the `pLDDT`. The experimental findings depicted in Figure 5C demonstrate that proteins designed by Pinal exhibit superior alignment with diverse protein descriptions and demonstrate high designability. Additionally, we report the capability of designing proteins according to various function categories in the Appendix.

Evaluation of Protein Sequence Design Based on Keywords Alone: A Comparison of Pinal and ESM3. We now evaluate ESM3's ability to design protein sequences solely based on keywords, without partial sequence or structural constraints, and compare its performance with Pinal. Noted that ESM3 recognizes keywords but lacks understanding of natural language. For a fair comparison, we feed Pinal with keywords extracted from InterPro (Paysan-Lafosse et al., 2023), the dataset ESM3 is trained on. Furthermore, we also report the metric introduced by ESM3, *i.e.* keyword recovery, which is calculated with InterProScan for predicting the consistency of the generated protein with the specified functions.

We notice that Pinal consistently outperformed ESM3 by generating proteins with quality (Figure 5D). The proteins from Pinal exhibit better foldability, as the `pLDDT` of ESM3 designed proteins varied from 20 to 100 evenly. Moreover, Pinal achieves a higher `ProTrek score`, indicating better alignment with desired protein functions. A similar conclusion can be drawn from keyword recovery (see ESM3 (Hayes et al., 2024) for details): while half of the proteins from Pinal exhibit predictable functions, only around 10% of the proteins generated by ESM3 do so.

4.4 Design Strategy Ablation

In this section, we investigate the two key design strategy issues that affect the quality of proteins, *i.e*. the importance of designing sequences conditioned on natural language and the optimal value of K, which is introduced in Section 3.1.

The necessity of textual conditioning sequence design. As existing protein language models, *i.e*. SaProt (Su et al., 2024a) and ProstT5 (Heinzinger et al., 2023), have shown a strong ability to predict amino acids given foldseek tokens. However, a pertinent question arises: Considering these models' proficiency in decoding structural information, is it essential to incorporate text input conditioning into SaProt's training?

Our answer is yes. Foldseek tokens offer coarse-grained structural cues about desired proteins, allowing for a multitude of possible amino acid sequences. Therefore, accurately predicting amino acids that align well with language descriptions from foldseek tokens is pivotal. To investigate this, we compare the SaProt-T (760M) with vanilla SaProt (650M)(Figure 5E). We feed them with foldseek tokens derived from natural proteins and evaluate the foldability and textual alignment of the predicted sequences. Our findings reveal that sequences predicted by SaProt-T, leveraging additional natural language input, outperform in textual alignment metrics and advanced foldability. This underscores the significance of enhancing SaProt's training with text input conditioning.

The optimal value of K. We next study the number of candidates to explore during the design process. Specifically, for each description on hard test set, Pinal designs K proteins and we select the top 5 sequences out of K candidates, ranked by the probability of the protein sequence. A larger value of K allows for more extensive exploration but typically increases the inference time required. Our findings, illustrated in Figure 6, indicate that compared to the case without our designed optimal sampling strategy (i.e., K = 5), there are notable improvements in both the mean and maximum `ProTrek score` as K increases, up until it reaches 50. Therefore, we selected K = 50 as a balanced compromise between exploration efficiency and performance enhancement throughout this paper.

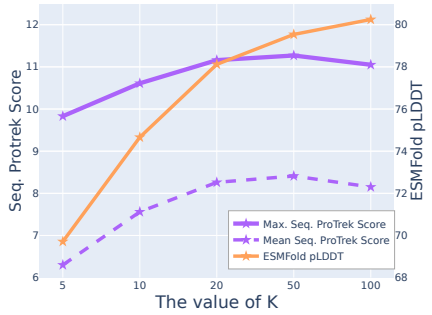

Figure 6: **Analysis of the optimal value of K.**

5 Related Work

5.1 *De Novo* Protein Design

De novo protein design refers to the process of creating novel proteins with desired structures and functions from scratch, without relying on existing proteins as templates or starting points in nature (Huang et al., 2016). Current research primarily focuses on two approaches: unconditional design and design conditioned on specific functions, which are roughly categorized into sequence design and structure design.

Most sequence design methods employ the Transformer (Vaswani et al., 2017) architecture to model the design space of proteins. For instance, Progen (Madani et al., 2023; Nijkamp et al., 2023), ProtGPT2 (Ferruz et al., 2022) and RITA (Hesslow et al., 2022) generate amino acid sequence in an autoregressive manner (Radford et al., 2018), while DPLM (Wang et al., 2024c) and EvoD-

iff (Alamdari et al., 2023) utilize discrete diffusion models (Austin et al., 2021) to generate proteins in a disorderly fashion. These methods are either dedicated to design sequences conditioned on given protein structure (Dauparas et al., 2022; Hsu et al., 2022; Gao et al., 2022; Goverde et al., 2024), or control tags (Nijkamp et al., 2023; Hayes et al., 2024).

Structure-based design methods often generate novel structures by diffusing in SE(3) space (Yim et al., 2023b). These approaches aim to unconditionally create protein structures suitable for in vitro design (Lin & AlQuraishi, 2023; Lin et al., 2024; Wang et al., 2024b; Yim et al., 2023a; Wu et al., 2024b). Compared to sequence design, direct design in protein structure space offers advantages for tasks like motif scaffolding or binder design that require specific structural features. Therefore, Trippe et al. (2022); Yim et al. (2024); Watson et al. (2023) tailor diffusion models to suit these specific applications. In contrast to relying on inverse folding models to predict protein sequences based on generated structure, (Campbell et al., 2024; Ingraham et al., 2023; Krishna et al., 2024) take a further step by proposing to design structure and sequence simultaneously.

5.2 Alignment Between Language and Protein

To integrate natural language and protein modalities effectively, (Su et al., 2024c; Liu et al., 2023; Xu et al., 2023; Wu et al., 2024a) adopt cross-modal contrastive learning (Radford et al., 2021), enhancing the prediction of protein functions and facilitating bidirectional retrieval between protein and natural language. Simultaneously, the increasing popularity of vision-language models (Liu et al., 2024a) has inspired the training of language models on datasets containing both biological information and natural language. Galactica (Taylor et al., 2023), a pioneering large language model (LLM) in this domain, has been trained on such combined datasets. However, as a general-purpose model, Galactica struggles to provide precise protein descriptions due to limitations in relevant training data. To mitigate this gap, (Zhang et al., 2023; Karim et al., 2022; Pei et al., 2023) focus on integrating meticulously curated biological knowledge into the training of LLMs, with the goal of supporting advancements in biological research. Moreover, (Lv et al., 2024; Abdine et al., 2024; Wang et al., 2024a; Liu et al., 2024b; Guo et al., 2023; Ziegler et al., 2023) specialize in elucidating proteins through tasks such as captioning or answering questions about specific proteins. Conversely, the field of designing proteins from textual descriptions has begun to garner attention among researchers. Recently, two works (Liu et al., 2023; Ingraham et al., 2023) have conducted preliminary explorations with very limited protein description data and a lack of appropriate evaluation metrics, thereby failing to provide comprehensive assessments. In contrast, the proposed Pinal was trained with 30-100 times more natural language-protein pairs than existing literature, resulting in more aligned protein generation.

6 Discussions

In this paper, we present Pinal, an innovative *de novo* protein design framework that uses natural language as a guiding principle. Instead of directly modeling the protein sequence design space, we propose a two-stage approach: first, translating protein language descriptions into structural modalities, followed by designing protein sequences based on both structure and natural language prompts. Additionally, we have developed an optimal sampling-based protein ranker that seamlessly integrates these two stages. Our comprehensive dry experiment evaluation demonstrates that proteins designed using Pinal not only exhibit high foldability but also align closely with their corresponding natural language prompts, outperforming proteins generated by recent methods. Furthermore, we confirm that our approach enables generalization beyond the training data, allowing for the design of proteins with novel functional combinations rather than merely memorizing sequences.

References

Hadi Abdine, Michail Chatzianastasis, Costas Bouyioukos, and Michalis Vazirgiannis. Prot2text: Multimodal protein's function generation with gnns and transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

Atomic-Level Accuracy. Design of a novel globular protein fold with. *science*, 1089427(1364):302, 2003.

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*, pp. 2023–09, 2023.

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17981–17993. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/958c530554f78bcd8e97125b70e6973d-Paper.pdf.

Amos Bairoch. The enzyme database in 2000. *Nucleic acids research*, 28(1):304–305, 2000.

Inigo Barrio-Hernandez, Jingi Yeo, Jürgen Jänes, Milot Mirdita, Cameron LM Gilchrist, Tanita Wein, Mihaly Varadi, Sameer Velankar, Pedro Beltrao, and Martin Steinegger. Clustering predicted structures at the scale of the known protein universe. *Nature*, 622(7983):637–645, 2023.

Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J Martin, Karine Michoud, Claire O'Donovan, Isabelle Phan, et al. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic acids research*, 31 (1):365–370, 2003.

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.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Longxing Cao, Brian Coventry, Inna Goreshnik, Buwei Huang, William Sheffler, Joon Sung Park, Kevin M Jude, Iva Marković, Rameshwar U Kadam, Koen HG Verschueren, et al. Design of protein-binding proteins from the target structure alone. *Nature*, 605(7910):551–560, 2022.

Bo Chen, Xingyi Cheng, Pan Li, Yangli-ao Geng, Jing Gong, Shen Li, Zhilei Bei, Xu Tan, Boyan Wang, Xin Zeng, et al. xtrimopglm: unified 100b-scale pre-trained transformer for deciphering the language of protein. *arXiv preprint arXiv:2401.06199*, 2024.

Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *arXiv preprint arXiv:1604.06174*, 2016.

Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

Ken A. Dill and Justin L. MacCallum. The protein-folding problem, 50 years on. *Science*, 338 (6110):1042–1046, 2012. doi: 10.1126/science.1219021. URL https://www.science.org/doi/abs/10.1126/science.1219021.

Michael J Dougherty and Frances H Arnold. Directed evolution: new parts and optimized function. *Current opinion in biotechnology*, 20(4):486–491, 2009.

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

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

Zhangyang Gao, Chen Tan, and Stan Z Li. Foldtoken3: Fold structures worth 256 words or less. *bioRxiv*, pp. 2024–07, 2024.

Team GLM, :, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Jingyu Sun, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024. URL https://arxiv.org/abs/2406.12793.

Casper A Goverde, Martin Pacesa, Nicolas Goldbach, Lars J Dornfeld, Petra EM Balbi, Sandrine Georgeon, Stéphane Rosset, Srajan Kapoor, Jagrity Choudhury, Justas Dauparas, et al. Computational design of soluble and functional membrane protein analogues. *Nature*, pp. 1–10, 2024.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1): 1–23, 2021.

Han Guo, Mingjia Huo, Ruiyi Zhang, and Pengtao Xie. Proteinchat: Towards achieving chatgpt-like functionalities on protein 3d structures. *Authorea Preprints*, 2023.

Tomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. Simulating 500 million years of evolution with a language model. *bioRxiv*, pp. 2024–07, 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. *bioRxiv*, 2023.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. Rita: a study on scaling up generative protein sequence models. *arXiv preprint arXiv:2205.05789*, 2022.

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pp. 8946–8970. PMLR, 2022.

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

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*, 623(7989):1070–1078, 2023.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, Aug 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Md Rezaul Karim, Hussain Ali, Prinon Das, Mohamed Abdelwaheb, and Stefan Decker. Question answering over biological knowledge graph via amazon alexa. *arXiv preprint arXiv:2210.06040*, 2022.

Rohith Krishna, Jue Wang, Woody Ahern, Pascal Sturmfels, Preetham Venkatesh, Indrek Kalvet, Gyu Rie Lee, Felix S Morey-Burrows, Ivan Anishchenko, Ian R Humphreys, et al. Generalized biomolecular modeling and design with rosettafold all-atom. *Science*, 384(6693):eadl2528, 2024.

Xiaohan Lin, Zhenyu Chen, Yanheng Li, Xingyu Lu, Chuanliu Fan, Ziqiang Cao, Shihao Feng, Yi Qin Gao, and Jun Zhang. Protokens: A machine-learned language for compact and informative encoding of protein 3d structures. *bioRxiv*, pp. 2023–11, 2023.

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

Yeqing Lin, Minji Lee, Zhao Zhang, and Mohammed AlQuraishi. Out of many, one: Designing and scaffolding proteins at the scale of the structural universe with genie 2. *arXiv preprint arXiv:2405.15489*, 2024.

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.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024a.

Shengchao Liu, Yanjing Li, Zhuoxinran Li, Anthony Gitter, Yutao Zhu, Jiarui Lu, Zhao Xu, Weili Nie, Arvind Ramanathan, Chaowei Xiao, et al. A text-guided protein design framework. *arXiv preprint arXiv:2302.04611*, 2023.

Zhiyuan Liu, An Zhang, Hao Fei, Enzhi Zhang, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. Prott3: Protein-to-text generation for text-based protein understanding. *arXiv preprint arXiv:2405.12564*, 2024b.

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

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 preprint arXiv:2402.16445*, 2024.

Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, 41(8):1099–1106, 2023.

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.

Typhaine Paysan-Lafosse, Matthias Blum, Sara Chuguransky, Tiago Grego, Beatriz Lázaro Pinto, Gustavo A Salazar, Maxwell L Bileschi, Peer Bork, Alan Bridge, Lucy Colwell, et al. Interpro in 2022. *Nucleic acids research*, 51(D1):D418–D427, 2023.

Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. Biot5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. *arXiv preprint arXiv:2310.07276*, 2023.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–16. IEEE, 2020.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022. URL `https://arxiv.org/abs/2210.08402`.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. In *The Twelfth International Conference on Learning Representations*, 2024a.

Jin Su, Zhikai Li, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, Dacheng Ma, The OPMC, Sergey Ovchinnikov, and Fajie Yuan. Saprothub: Making protein modeling accessible to all biologists. *bioRxiv*, pp. 2024–05, 2024b.

Jin Su, Xibin Zhou, Xuting Zhang, and Fajie Yuan. Protrek: Navigating the protein universe through tri-modal contrastive learning. *bioRxiv*, pp. 2024–05, 2024c.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. arxiv 2022. *arXiv preprint arXiv:2211.09085*, 10, 2023.

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.

Michel van Kempen, Stephanie S Kim, Charlotte Tumescheit, Milot Mirdita, Cameron LM Gilchrist, Johannes Söding, and Martin Steinegger. Foldseek: fast and accurate protein structure search. *Biorxiv*, pp. 2022–02, 2022.

Mihaly Varadi, Damian Bertoni, Paulyna Magana, Urmila Paramval, Ivanna Pidruchna, Malarvizhi Radhakrishnan, Maxim Tsenkov, Sreenath Nair, Milot Mirdita, Jingi Yeo, et al. Alphafold protein structure database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic acids research*, 52(D1):D368–D375, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Chao Wang, Hehe Fan, Ruijie Quan, and Yi Yang. Protchatgpt: Towards understanding proteins with large language models. *arXiv preprint arXiv:2402.09649*, 2024a.

Chentong Wang, Yannan Qu, Zhangzhi Peng, Yukai Wang, Hongli Zhu, Dachuan Chen, and Longxing Cao. Proteus: exploring protein structure generation for enhanced designability and efficiency. *bioRxiv*, pp. 2024–02, 2024b.

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*, 2024c.

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.

Kevin E Wu, Howard Chang, and James Zou. Proteinclip: enhancing protein language models with natural language. *bioRxiv*, pp. 2024–05, 2024a.

Kevin E Wu, Kevin K Yang, Rianne van den Berg, Sarah Alamdari, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *Nature communications*, 15(1): 1059, 2024b.

Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. Protst: Multi-modality learning of protein sequences and biomedical texts. In *International Conference on Machine Learning*, pp. 38749–38767. PMLR, 2023.

Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

Jason Yim, Andrew Campbell, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Regina Barzilay, Tommi Jaakkola, et al. Fast protein backbone generation with se (3) flow matching. *arXiv preprint arXiv:2310.05297*, 2023a.

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*, 2023b.

Jason Yim, Andrew Campbell, Emile Mathieu, Andrew YK Foong, Michael Gastegger, José Jiménez-Luna, Sarah Lewis, Victor Garcia Satorras, Bastiaan S Veeling, Frank Noé, et al. Improved motif-scaffolding with se (3) flow matching. *ArXiv*, 2024.

Kai Zhang, Jun Yu, Zhiling Yan, Yixin Liu, Eashan Adhikarla, Sunyang Fu, Xun Chen, Chen Chen, Yuyin Zhou, Xiang Li, et al. Biomedgpt: a unified and generalist biomedical generative pre-trained transformer for vision, language, and multimodal tasks. *arXiv preprint arXiv:2305.17100*, 2023.

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.

Xibin Zhou, Chenchen Han, Yingqi Zhang, Jin Su, Kai Zhuang, Shiyu Jiang, Zichen Yuan, Wei Zheng, Fengyuan Dai, Yuyang Zhou, Yuyang Tao, Dan Wu, and Fajie Yuan. Decoding the molecular language of proteins with evola. *bioRxiv*, 2025. doi: 10.1101/2025.01.05.630192.

Cheyenne Ziegler, Jonathan Martin, Claude Sinner, and Faruck Morcos. Latent generative landscapes as maps of functional diversity in protein sequence space. *Nature Communications*, 14(1): 2222, 2023.

A Showcase Prompts

Prompts for Fig. 1:

1. The primary role of this protein is to facilitate the reaction: beta-D-fructose 1,6-bisphosphate + H2O = beta-D-fructose 6-phosphate + phosphate through its enzymatic function.
2. Catalyzes the formation of 5-methyl-uridine at position 1939 (m5U1939) in 23S rRNA.
3. The protein can be found in Secreted. Compound that limits ion channel opening The protein sequence derives from the organism named Brush-footed trapdoor spider. The protein sequence derives from the organism named Trittame loki. The GO term encompassing toxin activity involves molecular function for this protein. Is included in the neurotoxin 14 (magi-1) family The subfamily labeled as 03 (ICK-30-40) sodium channel inhibitor activity falls under the GO term associated with this protein in relation to molecular function. U17-BATX-Tl1a is the official name of this protein. The GO term encompassing extracellular region involves cellular component for this protein. The assigned designation for this protein is U17-barytoxin-Tl1a.

4. The GO term for this protein constitutes membrane when considering cellular component. Is a member of the syntaxin protein family cellular component is involved in the GO term associated with this protein, encompassing presynapse. This particular protein can be found at Membrane. Golgi trans cisterna is encompassed by the GO term associated with this protein regarding cellular component. The GO term for this protein includes synaptic vesicle docking concerning biological process. The name Caenorhabditis elegans corresponds to the organism from which the protein sequence originates. In regards to molecular function, the GO term associated with this protein integrates SNAP receptor activity. The promotion of transport vesicle movement to target membranes is aided by SNARE (By similarity) Possibly operates in retrograde trafficking and endocytic recycling pathway (By similarity) This is the protein's designated term, Putative syntaxin 6.
5. The metabolic pathway associated with this protein encompasses Organic acid metabolism; propanoate degradation as well.
6. The GO term for this protein includes regulation of DNA repair concerning biological process.
7. Biological process is involved in the GO term associated with this protein, encompassing phagosome-lysosome fusion.
8. Carbohydrate biosynthesis; gluconeogenesis is an integral part of the metabolic pathway associated with this protein.
9. The presence of this protein enables the reaction: ATP + L-threonyl-[protein] = ADP + H(+) + O-phospho-L-threonyl-[protein] to be catalyzed through its enzymatic activity.
10. The catalytic activity of this protein allows for the reaction: guanosine(46) in tRNA + S-adenosyl-L-methionine = N(7)-methylguanosine(46) in tRNA + S-adenosyl-L-homocysteine to proceed.
11. Catalyzes the conversion of 3-phosphate to a 2',3'-cyclic phosphodiester at the end of RNA The enzyme's mechanism involves a three-step process: (A) adenylation by ATP, (B) transfer of adenylate to an RNA-N3'P to yield RNA-N3'PP5'A, and (C) initiating a reaction with the adjacent 2'-hydroxyl on the 3'-phosphorus in the diester linkage to generate the cyclic end product. The biological role of this enzyme is unknown but it is likely to function in some aspects of cellular RNA processing.
12. The organism associated with the protein sequence is referred to as Streptomyces cinnamonensis. The GO term for this protein constitutes antibiotic biosynthetic process when considering biological process. The metabolic pathway associated with this protein incorporates Antifungal biosynthesis; monensin biosynthesis. The protein sequence is attributed to the organism Streptomyces virginiae. In the taxonomic hierarchy, the source organism of this protein falls into the category Streptomyces. The designated name for this protein is ORF4. Is needed for correct cyclization of the oligoketide leading to isochromanequinone formation This protein is designated as Granaticin polyketide synthase bifunctional cyclase/dehydratase.

B DATASETS

Dataset	Proteins	Protein-Text Pairs	Text Tokens	Sample Weight
SwissProt-Annot.	545,256	14M	40M	7
SwissProt-Aug.	544,551	4M	1.2B	2
SwissProt-Enh.	544,551	9M	2.7B	4
UniRef50-ProTrek	41,239,335	412M	12B	13
UniRef50-Enh.	40,167,685	530M	143B	52
UniProtKB	113,109,489	791M	1.1B	15
AFDB	173,394,560	–	–	5

Table 1: **Dataset statistics.** For proteins sourced from these datasets, we have excluded those without Foldseek tokens. Additionally, we assign varying sample weights to each dataset during training.

Description for *P95368*
The GO term related to this protein regarding molecular function envelopes shikimate 3-dehydrogenase (NADP+) activity. In this specific protein, position 1 to 269 is marked by a polypeptide chain designated as Shikimate dehydrogenase (NADP(+)). Through molecular interaction, the residue at 89 becomes linked with shikimate. Pertinent to biological process, the GO qualifier for this protein captures aromatic amino acid family biosynthetic process. The GO term of this protein covers amino acid biosynthetic process when considering biological process. In terms of molecular affinity, the shikimate is targeted by the residue at 64. Under the scientific classification, the source organism for this protein aligns with Neisseriaceae. In the interaction between molecules, the residue at 17 to 19 attaches to shikimate. A bond is formed between the residue at location 154 to 159 and the NADP(+). Member of the shikimate dehydrogenase family.

Table 2: An illustrative example of a description from SwissProt-Annot.

B.1 SWISSPROT-ANNOT

Inspired by Su et al. (2024c), we collect a total of 56 subsections from Swiss-Prot to construct SwissProt-Annot. We then ask GPT-4 to generate a set of templates that combine the records from Swiss-Prot into complete sentences. An illustrative example of a description for *P95368* in UniProt is provided in Tab 2. To ensure a fair comparison, we split the sentence templates into training, validation, and test sets. This division allows us to effectively paraphrase the text.

Since Swiss-Prot represents catalytic activity primarily through chemical equations, which can be challenging to interpret, we supplement our enzyme-related data with functional annotations from Bairoch (2000). We incorporate records from the ENZYME nomenclature database. Specifically, we download data from the official website [2] and extract the descriptions and comments (the "DE" and "CC" lines). This process results in an additional 253,279 natural language-protein pairs in SwissProt-Annot.

B.2 SWISSPROT-AUG

For SwissProt-Aug dataset, following Zhou et al. (2025), we first generate questions relevant to the specific domain of the protein in question. We then gather descriptions from SwissProt-Annot and prompt *glm-4-flash* GLM et al. (2024) to provide answers to these domain-specific questions. The answers are retained as descriptions of the respective proteins. Details of the prompts used to elicit these answers are provided in Tab. 3.

B.3 OTHER DETAILS

Similarly, for SwissProt-enh and UniRef50-Enh, we retain only the answers to the question-answer pairs as protein descriptions.

Furthermore, the proteins from UniProtKB and the AlphaFold Database (AFDB) are clustered as described in Barrio-Hernandez et al. (2023), and the sampling during training is conducted according to these clusters. This clustering approach helps to avoid distribution bias and ensures a more balanced representation of the protein design space in our training process.

C DETAILS OF TRAINING

C.1 IMPLEMENTATION DETAILS

We implement the training using PyTorch. To optimize GPU memory usage, we employ activation checkpointing (Chen et al., 2016) and apply DeepSpeed ZeRO-2 (Rajbhandari et al., 2020) for gradient partitioning and optimizer partitioning. All models are trained using BFloat16 mixed-precision to enhance efficiency.

[2] https://ftp.expasy.org/databases/enzyme/enzyme.dat

Prompt
You are an AI biology assistant capable of analyzing a single protein. You are provided with several sentences that describe the same protein you are examining. Using the information provided, you should analyze the protein and answer any related questions in detail. Rather than directly referencing the site information, use it to conduct a thorough analysis of the protein in natural language, discussing aspects such as the functions of specific sites, their relationships to other proteins, and their contributions to the overall functions of the protein. When incorporating details from the provided information, focus on analyzing the protein without explicitly stating the sources of your information. Always refer to the protein as "this protein" and maintain an analytical perspective throughout your response. If a question cannot be answered based on the available details, do not reference the source of the information. Here are the relevant sentences about the protein. {Descriptions from SwissProt-Annot.} {Question}

Table 3: Prompt for SwissProt-Aug. generation.

Total Params	**Encoder Params**	**Decoder Params**	**Text Length**	**Protein length**	**Learning rate**	**Batch size**	**Num steps**
224M	110M	114M	768	1024	1e-4	512	320K
1.2B	335M	922M	768	1024	1e-4	512	320K
15.5B	4.8B	10.7B	768	1024	1e-5	512	320K

Table 4: Hyperparameter details.

We train the models using the standard AdamW optimizer (Loshchilov, 2017), configuring the hyperparameters to β_1=0.9, β_2=0.98 and ϵ = 1e-8. To prevent overfitting, we apply a weight decay of 0.1. The model is trained for a total of 320,000 steps, during which the learning rate is gradually increased over the first 10% of the steps to reach the maximum learning rate, followed by a decay using a cosine decay scheduler. Additional details on other hyperparameters are provided in Table 4.

C.2 Large Model Training Stability

Training the 15.5B T2struct model presents more challenges compared to the two smaller models. We observed that its training and validation loss exhibited spikes and struggled to converge after the learning rate warmup phase. Drawing inspiration from the work of Hayes et al. (2024); Yang et al. (2022), which suggests adapting the learning rate based on the model's parameters, we reduced the learning rate to 1e-5, 10 times smaller than previous value. This adjustment effectively resolved the training stability issues, allowing the model to converge more reliably and maintain a more consistent loss trajectory throughout the training process.

D Performance Across Subsections

To delve deeper into the influence of protein f unction, we present an evaluation of proteins designed by Pinal across different function categories in Figure 7. This analysis includes comparison with random `ProTrek Score` and those computed using ground truth proteins for further insight. We observe a positive correlation between foldability and language alignment, where improved language alignment typically corresponds to a greater likelihood of folding into a real protein structure. Pinal excels in designing proteins based on practical descriptions such as protein names and sequence similarity. However, it exhibits limitations when tasked with abstract functional descriptions, such as disruption phenotype and induction. Nevertheless, even with these challenging tasks, Pinal demonstrates the capability to design proteins with high foldability (`pLDDT` > 80).

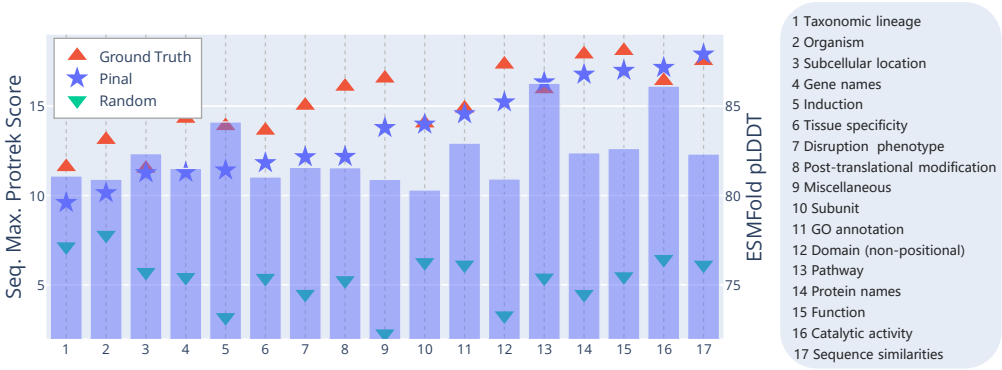

Figure 7: **Detailed performance analysis across diverse protein function descriptions.** The bar represents `pLDDT` and the star represents `ProTrek score`.