Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

PRO-GO: Reference-Guided Protein Sequence Generation using Gene Ontology Terms

DARREN TAN[1,2], IAN MCLOUGHLIN[2], (Senior Member, IEEE), AIK BENG NG[1], ZHENGKUI WANG[2], (Member, IEEE), ABRAHAM C STERN[1], and SIMON SEE[1], (Senior Member, IEEE),
[1]NVIDIA Singapore, Singapore 339949
[2]Singapore Institute of Technology, Singapore 138683

Corresponding author: Darren Tan (e-mail: dhui@nvidia.com).

The work was supported by the Singapore Economic Development Board and NVIDIA Singapore via the Industrial Postgraduate Program PhD scholarship.

ABSTRACT Protein sequence generation models aim to produce valid protein candidates on demand, however, controllably generating protein sequences with specified target functionalities remains difficult. The few existing models able to controllably generate protein sequences are limited to broad classes. We present a novel method for general controllable protein sequence generation that leverages reference sequences to guide generation, and specifies target characteristics through Gene Ontology (GO) terms. We design an evaluation pipeline based on a Top-TM-score target metric that prioritizes closest-shape matching with ground truth exemplars. We evaluate the effectiveness of this reference-guided controllability approach for protein design across various models and a diverse range of GO term combinations. Results demonstrate controllability with exhibit high accuracy against target benchmarks.

INDEX TERMS protein design, controllable generation, GO terms, reference sequences, protein sequences

I. INTRODUCTION

Designing protein sequences opens exciting opportunities in synthetic biology, enabling the design of custom molecules that could revolutionize medicine, advanced materials, and biological engineering [1], [2]. Protein design involves researchers creating proteins with new or modified sequences and structures, expanding beyond the limitations of naturally occurring proteins [3]. Combining advanced computational techniques with artificial intelligence (AI), promises the ability to generate proteins on demand for applications such as therapeutics [4], [5]. For example, developing better diagnostic tests, creating new treatments, and designing more effective vaccines [6]. Designing protein binders has also shown promise in targeting diseases like chronic myeloid leukemia and enhancing CAR T-cell activity.

While AI solutions for protein generation have been demonstrated, precise functional or property control has been lacking. This is particularly true when targeting multifunctional capabilities. Specifically, the design of proteins with defined binding specificity and multiple cooperative functions remains a persistent challenge in the field [7]. For example, PrefixProt [8], revolutionized protein design by leveraging the power of large-scale protein language models that are prefix-tuned on two datasets. While impressive, it is limited to generation within the two categories of proteins it was trained on. There is a pressing need for generalized techniques that combine flexibility, scalability, and controllability for protein sequence generation tasks.

In Pro-GO, we address the limitations of current models by proposing a novel framework capable of flexibly specifying the protein sequence targets for generation. Our approach leverages reference sequences described by Gene Ontology (GO) terms [9], [10], which are well-defined and widely adopted descriptors of protein properties and functionality, further elaborated in Section II-C.

This approach addresses fundamental limitations of current protein design methodologies, which are constrained by the need for extensive model retraining and limited input functional annotations. The proposed framework makes use of protein sequence reference sets, of ground truth proteins that match target GO terms. It enables flexible generation of sequences exhibiting those target GO term properties on demand, and spans diverse functional categories without the computational burden of repeated model training for each individual category. This integration of structural prototypes and flexible ontological classification enables both precise control and broad adaptability, representing a significant advance in computational protein engineering.

The key contributions of this work are thus (a) a reference sequence guided framework to augment model generation capabilities, (b) the ability to specify user-friendly target properties through the integration of GO-term descriptors. Secondary contributions are (c) an evaluation pipeline that assesses the accuracy of the generated protein sequences in terms of structural similarity. (d) useful insights into the capability of Large Language Models (LLMs) to controllably generate proteins to target.

The remainder of this paper briefly overviews related work in Section II and the databases and evaluation tools used in Section III, before presenting the proposed approach in Section IV, discussing results and their analysis in Sections V and VI, before Section VII concludes the paper.

II. RELATED WORKS

A. PROTEIN DESIGN BY AI

In recent years, the field of AI-enabled protein design has seen promising advancements through the application of language models and machine learning techniques. For example, ProtGPT2 [11], an autoregressive GPT-based model of 0.7B parameters, trained on protein sequences, able to unconditionally generate statistically correct protein sequences in an efficient manner. Meanwhile, ProGen2 [12] incorporates an attention based language model of up to 6.4B parameters trained on over one billion protein sequences. It has a well-demonstrated capability to generate sequences that match the distribution of observed sequences, with quality and fitness. ESM-2 [13] is the state-of-the-art model with up to 15B parameters that can predict protein structure and function from sequences. It is part of a set of ESM models that are able to perform inverse folding, predict sequences and variant effects.

While these models are undeniably capable, there exists a gap in terms of controlling a proteins' properties and function to produce on-demand proteins to a given specification. This task essentially embodies two aspects of how to control generation within a model, and how to specify that control.

B. CONTROLLABLE PROTEIN DESIGN

As we have discussed above when introducing ProtGPT2 [11] and ProGen2 [12], achieving controllability in protein design is a crucial next step in this research. Models like ProLLaMA [14] introduced a training pipeline that builds on general LLMs, turning them into protein language models capable of controllable generation and property prediction. Other models like PrefixProt [8] employ prefix-tuning to learn virtual tokens as control tags, enabling the generation of proteins based on the datasets it was tuned on. These solutions bring us closer to the goal of protein design controllability, however they require extensive training and even re-training when we desire new functional annotations, different from the ones they were trained on.

C. CONTROL SPECIFICATION

In this work we make use of GO terms [15], [16] which describe and categorize protein functions. GO terms provide a structured vocabulary, and are widely used in bioinformatics [17]. They can be categorized into 3 main domains: biological process, molecular function, and cellular component [15], each providing well-defined, widely adopted descriptors of protein properties and functionality.

They provide a refined coverage of protein functionality, and ability to capture the complexity of biological roles. GO terms can be highly specific and are derived from experimental evidence, making them valuable for functional predictions [18]. Combinations of multiple GO terms enable intuitive and precise targeting of desired protein properties, making the method accessible to a broad user base in bioinformatics and synthetic biology. In essence, GO terms provide a standardized vocabulary to describe protein functions across different organisms, making them ideal for protein description [19], [20], and thus controllable generation.

Other methods of categorizing proteins include the Protein Superfamilies [21], which are those defined by CATH [21] or SCOP, categorize proteins based on structural similarities and evolutionary relationships, thus may not always reflect protein functional similarities accurately. For instance, proteins within the same superfamily can have diverse functions despite sharing a common ancestor [17], [21]. Superfamilies are primarily structural classifications, whereas GO terms are functional classifications.

D. IN-CONTEXT LEARNING

In-context Learning is a powerful capability of LLMs that allows them to carry out tasks that they were not explicitly trained on. This is done by leveraging demonstrations provided in the input prompt [22]. We employ this method to steer the generation of proteins using language models by providing reference protein sequences into the input prompt. The source of these references protein sequences are further discussed in Section III-A, and the usage can be seen in Section IV-B.

III. AI-ENABLED PROTEIN GENERATION

A. DATABASES

For the creation of a reference protein set, we employ the UniRef50 database [23] which consists of sequences from the UniProt Knowledgebase clustered at 50% sequence identity. The clustering results in a significant reduction in database size while maintaining comprehensive coverage of the protein sequence space. In practice, it improves the detection of distant relationships between proteins, while allowing us to efficiently leverage the functional annotation (GO terms) associated with the protein sequences efficiently.

From UniRef50 (Release 2024-06), we extracted a total of 21,510,292 protein sequences annotated with at least one GO term. Our MySQL database contains 21,227,938 sequences of 50 amino acids or longer (excluding 282,354 shorter sequences), with lengths ranging from 50 to 45,354 amino acids (mean = 359.8 ± 373.4 aa). This comprehensive dataset provides substantial functional diversity across all major GO term categories, and is used for providing reference sequences and target benchmark sequences.

The Protein Data Bank (PDB), managed by the Research Collaboratory for Structural Bioinformatics (RCSB), serves as a single worldwide repository for macromolecular structure data [24]. The data have been experimentally determined, and are used to provide our framework with high quality and extensive ground truth data. Structures derived from PDB based on the target GO term set are used for the evaluation pipeline. Specifically, for each target GO term set, we extract all of the matching PDB entries. The structural similarity of every generated candidate protein is then compared to the matching PDB entries.

B. EVALUATION

The full evaluation pipeline will be presented in Section IV-D. It makes use of **ESMFold** [13], running on NVIDIA Inference Microservices (NIMs), a deep learning-based protein structure predictor that has been trained to map sequences to structures. We analyze structures using **FoldSeek** [25], which is a tool for comparing and assessing structural similarity. FoldSeek allows us to accurately and efficiently evaluate a given structure input against known protein structures (e.g. ground truth proteins).

IV. APPROACH

A. FRAMEWORK OVERVIEW

The proposed framework takes a reference-guided approach that leverages LLMs as sequence generators, as illustrated in Figure 1. GO terms are provided as user input to define target protein properties, with reference proteins from PDB that meet those properties used to guide the generation process.

The framework is divided into two stages of **Generation** and **Analysis**. The generation stage creates candidate protein sequences that attempt to match the target GO terms, whereas the analysis stage predicts their 3D structures to assesses the suitability of the generated proteins, and further refine the candidate list. The two stages are described in the following subsections.

B. GENERATION STAGE

The aim of the generation stage is to controllably design protein sequences based on the input target GO terms. It consists of two sub-stages, namely **Input Preparation**, and **Sequence Generation**.

The former defining the input data to the framework, formed into a data record comprising:

- **Target GO terms** $G_{1\ldots N_G} \in \mathbf{G}$ which define target functional and structural attributes, all of which should be met by the generated protein.
- **Reference Sequences** $R_{1\ldots N_R} \in \mathbf{R}$ which serve as example sequences for the model to learn from during protein generation. These are existing protein sequences that are associated with the target GO terms, which we extract on demand from our UniRef [23] database.
- **Instruction** is a prompt structure to guides the LLM to generate sequences [1].

The **Sequence Generation** sub-stage feeds the input reference proteins $\mathbf{R}$, the target GO terms $\mathbf{G}$ and a set of instructions, into a Language Model, $\mathcal{LM}$. The model generates protein sequence candidates (S) by integrating the guidance from reference sequences, and the specifications from the GO term list. The generation process to output protein sequence candidate S can therefore be expressed as,

$$S = \mathcal{LM}(\mathbf{R}, \mathbf{G}; \text{Instruction}) \quad (1)$$

C. ANALYSIS STAGE

The analysis stage (Figure 2) assesses the generated sequences S, and its predicted structures S', against a set of ground truth protein structures B'. In our experimental evaluations, this stage also aims to determine the efficacy of the model at fulfilling the required task, i.e. to generate proteins that meet the target GO terms.

Firstly, the **ESMFold** protein structure prediction model predicts 3D structural representations (S') of the candidate sequences (S):

$$S' = \text{ESMFold}(S) \quad (2)$$

S' is analyzed using **FoldSeek** [25] to compare and assess the structural similarity against protein structures from PDB [24] that match the wanted GO terms. These proteins from PDB act as our ground truth structures (B').

FoldSeek assigns a **TM-score** to each comparison pair (i.e. between every S' and every B'), which quantifies their structural similarity and integrity. A higher TM-score implies a closer structural consistency, which in turn implies there is a greater functional similarity between the generated sequence and the given reference.

If $\mathcal{FS}$ represents FoldSeek, then the TM-score T between candidate structure S' and the ground truth structure B' can be represented as:

$$T = \mathcal{FS}(S', B') \quad (3)$$

As noted, T correlates with structural accuracy as well as functional relevance between the generated protein sequence and each ground truth structures. How the analysis stage is used to evaluate the efficacy of models is described in Section IV-D.

D. EVALUATION PIPELINE

The evaluation pipeline brings together the Generation Stage and the Analysis Stage for a complete end-to-end system.

It assesses each of the N_S output sequences, S, as generated by $\mathcal{LM}$, in turn. For each S, it uses ESMFold to predict their structure S'. We then make use of $N_{B'}$ ground truth example proteins from PDB, B', chosen to match the same target GO

[1] The experimental prompt structures to ensure novel generation will be provided in the code availability

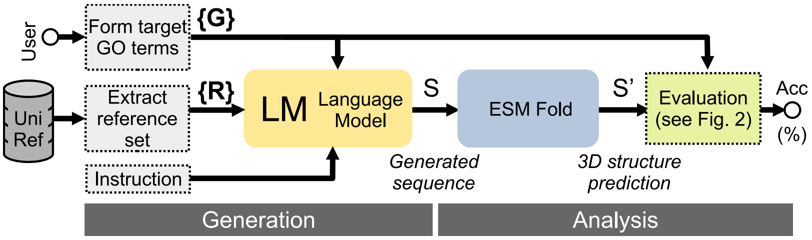


FIGURE 1. Framework Overview — The overall pipeline with references from UniRef, GO term input, LM-based generation, and structure analysis.

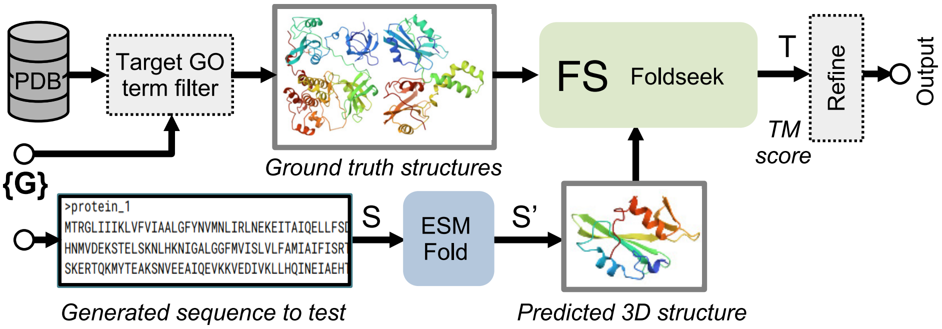

PDB
Target GO term filter
{G}
Ground truth structures
FS Foldseek
T
TM score
Refine
Output
>protein_1
MTRGLIIIKLVFVIAALGFYNVMNLIRLNEKEITAIQELLFSD
HNMVDEKSTELSKNLHKNIGALGGFMVISLVLFAMIAIFISRT
SKERTQKMYTEAKSNVEEAIQEVKKVEDIVKLLHQINEIAEHT
S
ESM Fold
S'
Generated sequence to test
Predicted 3D structure

FIGURE 2. The analysis stage compares generated protein sequences (S) against PDB-sourced sequences in terms of TM-score (T) derived from FoldSeek

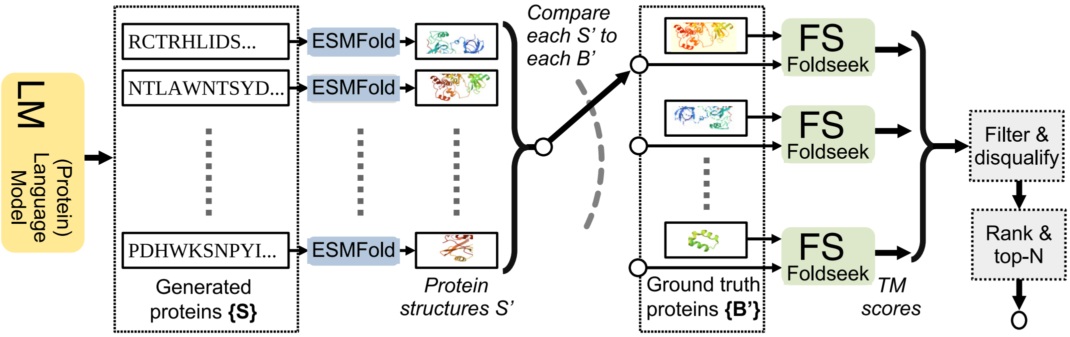


FIGURE 3. TM-based score evaluation pipeline on a set of generated proteins, compared against a set of ground truth proteins.

terms that were used by $\mathcal{LM}$. Next, $N_{S'} \times N_{B'}$ comparisons are made using FoldSeek,

$$T_{m,n} = \mathcal{FS}(S'_n, B'_m) \text{ for } m = 1 \ldots N_{B'} \text{ and } n = 1 \ldots N_{S'} \quad (4)$$

A filter and disqualification process removes invalid sequences and recall sequences (i.e. generated sequences that are not protein sequences, as well as any that are identical to sequences in the database) and also removes those for which a TM score cannot be computed, there are various reasons for this, as discussed in [25], all of which indicate a protein sequence that we will disqualify from further processing.

Finally, we rank the resulting TM scores from highest to lowest and consider only the top ranked score for each generated protein. Therefore, there will be one top TM-score associated with each generated protein, and a count of how many generated proteins exceed a target TM score thresholds θ. For clarity, Algorithm 1 briefly outlines the stages in order.

TM-score is employed as the evaluation metric because it measures structural similarity between generated protein candidates and known ground truth structures matching the target GO terms. While GO terms describe functional and localization properties, they do not always imply consistent structural similarity, i.e. proteins with very different structures can have similar GO terms.

Because of this, we adopt a Top-TM-score approach: rather than expecting all generated proteins to match every reference structure, we focus on whether each generated structure is similar to any known structure associated with the target GO term set. This allows us to account for natural structural diversity within a GO category, while still assessing whether the generated protein falls within the functional "space" implied by the GO terms.

Though TM-score is not a direct measure of functional activity, high structural similarity is a strong proxy for functional similarity, especially in the absence of wet-lab validation. Since the proposed framework generates protein candidates guided by GO terms, but ultimately aims to produce functionally plausible proteins, structure-based evaluation is

Algorithm 1 Controllable Protein Sequence Generation and Evaluation

```
Input: Target GO terms $G$, Reference Sequences $R$ from UniRef50
Output: Generated protein sequences $S$ with structural evaluation

// Generation Stage
procedure GenerateProteinSequences($G, R$)
    Prepare input with GO terms and reference sequences
    Feed $R$ and $G$ into Language Model $LM$
    Generate candidate sequences:
        $S \leftarrow LM(R, G;$ Instruction$)$
    Return $S$
end procedure

// Analysis Stage
procedure EvaluateGeneratedProteins($S, G$)
    Predict structures: $S' = ESMFold(S)$
    Retrieve ground truth proteins $B'$,
        having GO terms $G$, from PDB
    Compute TM-scores: $T = FS(S', B')$ using FoldSeek
    Filter invalid sequences (non-protein, or recall)
    Rank sequences by $T$ and and keep top TM-score
    Count sequences exceeding each TM-score
        threshold $\theta$
    Return Counts of sequences exceeding different $\theta$
end procedure

// Main Execution Flow
procedure Main( )
    Load GO terms $G$ and Reference Sequences $R$
    $(S, S') \leftarrow$ GenerateProteinSequences($G, R$)
    $S'_{valid} \leftarrow$ EvaluateGeneratedProteins($S', G$)
    Output final generated and validated sequences
end procedure
```

a better measure of that goal. In this view, GO terms are used as interpretable and accessible control handles, while TM-score assesses whether the generated proteins align with the functional prototypes those GO terms typically represent.

V. EXPERIMENTS

We evaluate the effectiveness of the proposed controllable protein sequence generation framework in two main experiments, complemented by ablation and data analysis studies:

- **Comparison of different models** to assess the impact of different language models on protein generation accuracy withing the proposed framework
- **Generalization to different GO terms** to investigate how the system performs when using different target GO term combinations

The results from each experiment are obtained from the evaluation pipeline described in Section IV-D, where we predict the 3D structures of generated proteins using ESMFold, and estimate similarity using FoldSeek's default local 3Di+AA structural alignment, supporting realigning hits using the global TMalign as well as recording alignments using TM-score [25].

A 'Target Benchmark' comparison is also made, in which a look-up function searches for GO term matches in the UniRef50 database [23], and randomly extracts the same number of candidate sequences as in the framework. These known-good protein sequences are then fed through the same evaluation pipeline that protein candidate sequences go through to provide a target upper bound of performance to benchmark different LMs against.

A. COMPARISON OF DIFFERENT MODELS

To assess how different language models perform in generating protein sequences that match target GO terms, we measure structural accuracy of candidates generated from different LMs using the TM-score assessment outlined in Section IV-D.

1) Parameters

For consistency, each tested LM is instructed to generate proteins matching the same target GO terms. We selected three GO terms with useful functions related to kinase activity on the basis that membrane-bound kinases involved in signal transduction pathways are important for understanding cellular communication when developing targeted therapies for diseases such as cancer. The specific terms are:

1) Kinase Activity (MF: GO:0016301)
2) Signal Transduction (BP: GO:0007165)
3) Plasma Membrane (CC: GO:0005886)

To ensure generalizability to diverse GO terms, a second experiment (in Section V-B) will assess performance across a large random set of different target GO terms.

For comparing LMs, each tested model generates 100 protein sequences (N_S = 100) in response to the target $\mathbf{G}$ = {GO:0016301, GO:0007165, GO:0005886}. For each of the resulting generated sequences, a set of reference sequences $\mathbf{R}$ is randomly selected from UniRef50 (N_R = 5) for comparison, based on their GO term association, i.e. 100 sets of 5 reference sequences. Note that we assess the effect of the number of reference sequences in Section V-C1.

The compared LMs were derived from the following baseline models:

- **Llama 3.1 8B** [26]: multilingual LLM by Meta, induced with reference examples, running on NIMs via API calls.
- **Llama 3.1 70B**: [26] a bigger multilingual LLM by Meta, induced with reference examples, also running on NIMs API.
- **ProLLaMA 8B** [14]: protein language model capable of controllable protein sequence generation, induced with reference examples altered to match the model's trained input syntax (i.e. [Generate by Go term] Go term=<xxx>). The model is running on NVIDIA RTX 6000 Ada.

- **ProtGPT2** [11]: protein language model capable of protein sequence generation, but with an uncontrolled set of generated proteins. The model is running on NVIDIA RTX 6000 Ada.
- **ProGen2** [12]: xlarge version capable of generating functional protein sequences across protein families, again representing an uncontrolled set of generated proteins. The model is running on NVIDIA RTX 6000 Ada.
- **Target Benchmark** Not a LM, but serves as an indicator of potentially achievable performance.

2) Results

Table 1 explores the results, comparing the LMs in terms of the TM score, T, of their generated proteins. 'Valid' indicates the proportion of the comparisons that resulted in actual protein sequences that could yield a TM score (non-valid outputs are the generation of invalid sequence characters or complete lack of structural similarity) while 'Recall' indicates the proportion of generated protein sequences that were exact matches to those in the database. Since they are simply repeat sequences, they are considered to be invalid LM outputs. The *Benchmark* column contains natural proteins and could be considered to be an empirical upper bound. Recall does not apply to this column because all are unique. After filtering invalid proteins, the remaining sequence TM scores are evaluated to provide the maximum, minimum, median, mean, and standard deviation (σ) across the set of valid proteins $\mathbf{S}$.

The last row in Table 1 indicates the proportion of sequences with high accuracy. Those exceeding a TM score threshold of $\theta = 0.8$ which is a level generally considered to equate to a very high level of structural similarity [27] [28] [29]. The proportion of sequences above this threshold is an indicator of the quality of the framework in terms of its ability to generate proteins that meet target specifications.

From Table 1, we can observe several interesting findings:

- The proposed framework works better with larger language models, indicated by the performance of the 70B version of Llama 3.1 compared to the 8B version.
- As the technique leverages in-context learning, employing language models that understand our textual instructions perform better. This is evident by comparing scores from ProLLaMA, which was trained in the language of proteins, but seemingly is unable to make use of the reference input examples, unlike the similar sized Llama 3.1 8B model.
- When compared to the uncontrolled generation of protein sequences, we see the proposed technique steers the controllable Llama models to outperform the uncontrolled ProtGPT2 and ProGen2 models. This shows that a random set of protein sequences will not achieve a high accuracy score.
- Among the evaluated models, Llama 3.1 70B most closely matches the Target Benchmark, i.e. accuracy approaching the empirical upper bound. We therefore use this model for all subsequent experiments.

Noting that Llama 3.1 70B outperforms protein language models such as ProtGPT2, ProGen2, ProLLaMA, we hypothesize that this is due to two main factors:

First, unlike the protein-specific models trained solely on amino acid sequences, Llama 3.1 is a general-purpose LLM trained on diverse natural language data. This makes it better at interpreting and responding to prompts that include structured instruction, as well as reference examples as context. In our framework, the model is given both protein sequences and task instructions, such as GO term descriptions, in natural language. General LLMs like Llama are uniquely suited to understand these cues and generate accordingly.

Second, the reference-guided approach relies heavily on in-context learning, where the model must infer its task from examples and descriptive inputs. Protein-only models may perform well at generating valid protein sequences, but they lack the training to associate such prompts with specific generation goals. By contrast, Llama 3.1 70B can connect the natural language instruction with the reference patterns, allowing it to produce sequences that better match the intended function.

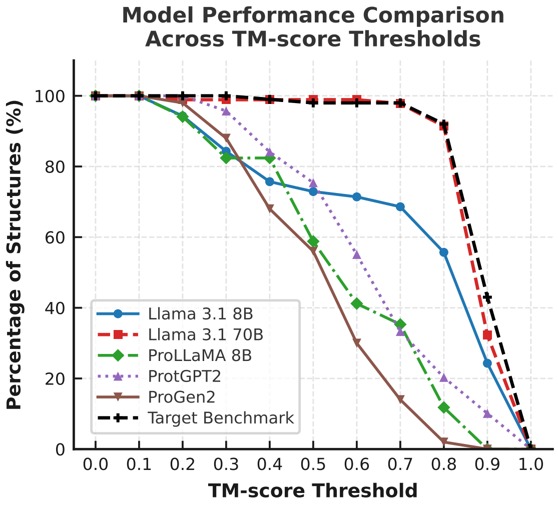


FIGURE 4. Percentages of structures for each LM at different TM score thresholds. The Target Benchmark is illustrated with a black dotted line. We observe that Llama 3.1 70B, illustrated with a dotted red line, resembles closely with the target benchmark, even overlapping it at some points, while the other LMs differ in a much greater scale.

We evaluate performance in more detail in Figure 4, where different TM score thresholds are explored ($0 \leq \theta \leq 1.0$). At low TM-score thresholds ($\theta \leq 0.3$), most generated sequences maintain at least a weak resemblance to known structures. However, as the TM-score threshold increases, notable differences in structural fidelity emerge across models, and this is differentiated further at higher thresholds which demand a close degree of structural similarity to known-good sequences ($\theta \geq 0.8$). We observe that:

TABLE 1. Performance comparison of different generative models in terms of TM-score.

Metric	Benchmark	Llama 3.1 8B	Llama 3.1 70B	ProLLaMA 8B	ProtGPT2	ProGen2
Valid, %	100	70	98	17	69	50
Recall, %	N/A	0	1	5	0	0
max(T)	0.992	0.967	0.974	0.886	0.962	0.815
min(T)	0.335	0.135	0.174	0.144	0.238	0.104
med(T)	0.890	0.824	0.867	0.554	0.615	0.524
mean(T)	0.884	0.698	0.863	0.562	0.622	0.505
$\sigma(T)$	0.092	0.271	0.091	0.223	0.186	0.167
$\theta = 0.8$	92 (92.0%)	39 (55.7%)	85 (91.4%)	2 (11.8%)	14 (20.3%)	1 (2.0%)

Note: $\theta = 0.8$ implies sequences with very high structural similarity [27]–[29].

- **Llama 3.1 70B** demonstrates superior performance at the most stringent TM-score thresholds.
- **Llama 3.1 8B** performs significantly better than the other models, excluding its larger counterpart
- **ProLLaMA 8B** achieve moderate performance with similar model complexities, but demonstrates significant decline at higher TM-score thresholds.
- **ProtGPT2** and **ProGen2** exhibit the steepest drop in accuracy as threshold increases with a lower capacity to generate sequences with high structural fidelity. While these models are specifically trained for protein generation, they are unable to controllably generate sequences to match the target GO terms.

Based on the results, we observe that Llama 3.1 70B performs the best, even better than language models that were well trained specifically on protein language. We thus employ Llama 3.1 70B in the following experiments to further validate our comparisons.

B. COMPARISON ACROSS DIFFERENT GO TERMS

The generalizability of the framework to different sets of GO terms is now explored. The generation and evaluation stages follow the previous experimental settings, where N_S = 100 protein sequences were generated for each set of target GO terms. For every generated sequence, N_R = 5 reference sequences were again randomly selected from the UniRef50 database to match the target GO terms, resulting in 100 unique sets of **R** for comparison.

The precise selection of target GO term sets used in this experiment was guided via a number of analysis and filtering steps that are outlined in Appendix A. The result is 100 randomly selected target GO term sets of different sizes and constituents[2].

1) Results

To understand the relationship between Llama 70B and target benchmark performance across different structural similarity levels, we conducted a comprehensive correlation analysis at the top six TM-score thresholds that were plotted in Figure 5. This analysis aims to determine how consistently the Llama 70B and known-good benchmark relate to each other across all target GO term combination sets.

TABLE 2. Comparison of structural similarity across TM-score thresholds between Llama 70B-generated sequences and the target benchmark. Absolute differences are all below 6%.

TM-score Threshold	Llama 70B	Target Benchmark	Difference
≥ 0.3	95.3%	97.0%	1.7%
≥ 0.4	90.9%	93.8%	2.9%
≥ 0.5	85.5%	90.2%	4.7%
≥ 0.6	78.7%	84.4%	5.7%
≥ 0.7	70.5%	75.8%	5.3%
≥ 0.8	53.7%	57.2%	3.5%
≥ 0.9	29.1%	30.8%	1.7%

We observe strong positive correlations between Llama and UniRef50 performance for all TM-score thresholds: (1) Correlation coefficients ranged from r = 0.806 to r = 0.926, (2) Correlations observed at higher thresholds exceed r > 0.8 (θ = 0.8, θ = 0.9), (3) All correlations were statistically significant (p < 0.001) (The p-values were calculated using `scipy.stats.pearsonr()`, which performs a statistical test for the Pearson correlation coefficient).

To further assess the similarity between Llama 70B and UniRef50 protein sequences, we conducted paired t-tests at seven TM-score thresholds in Table 2. The paired t-test results indicate that Llama 70B generates protein sequences with structural quality close to those from UniRef50, with absolute differences not exceeding 6%.

C. FURTHER EVALUATION

1) Effect of number of reference sequences

We explore the effect of the number of reference sequences on the accuracy of generated sequences to determine an optimal number of reference sequences based on both accuracy and model limits.

The target GO terms are kept the same as in Section V-A1, and Llama 3.1 70B is selected. The same number of protein sequences are generated, i.e. $N_S = 100$, but in this experiment we alter the number of reference protein sequences, $N_R = \{1, 2, 3, 4, 5\}$ where $N_R = 5$ is the practical upper limit, constrained by the model context window [30]. Similar to the previous experiments, 100 unique sets of **R** are used, and are kept constant for all evaluations in this section.

[2] The precise set can be found in the repository, see Section VIII.

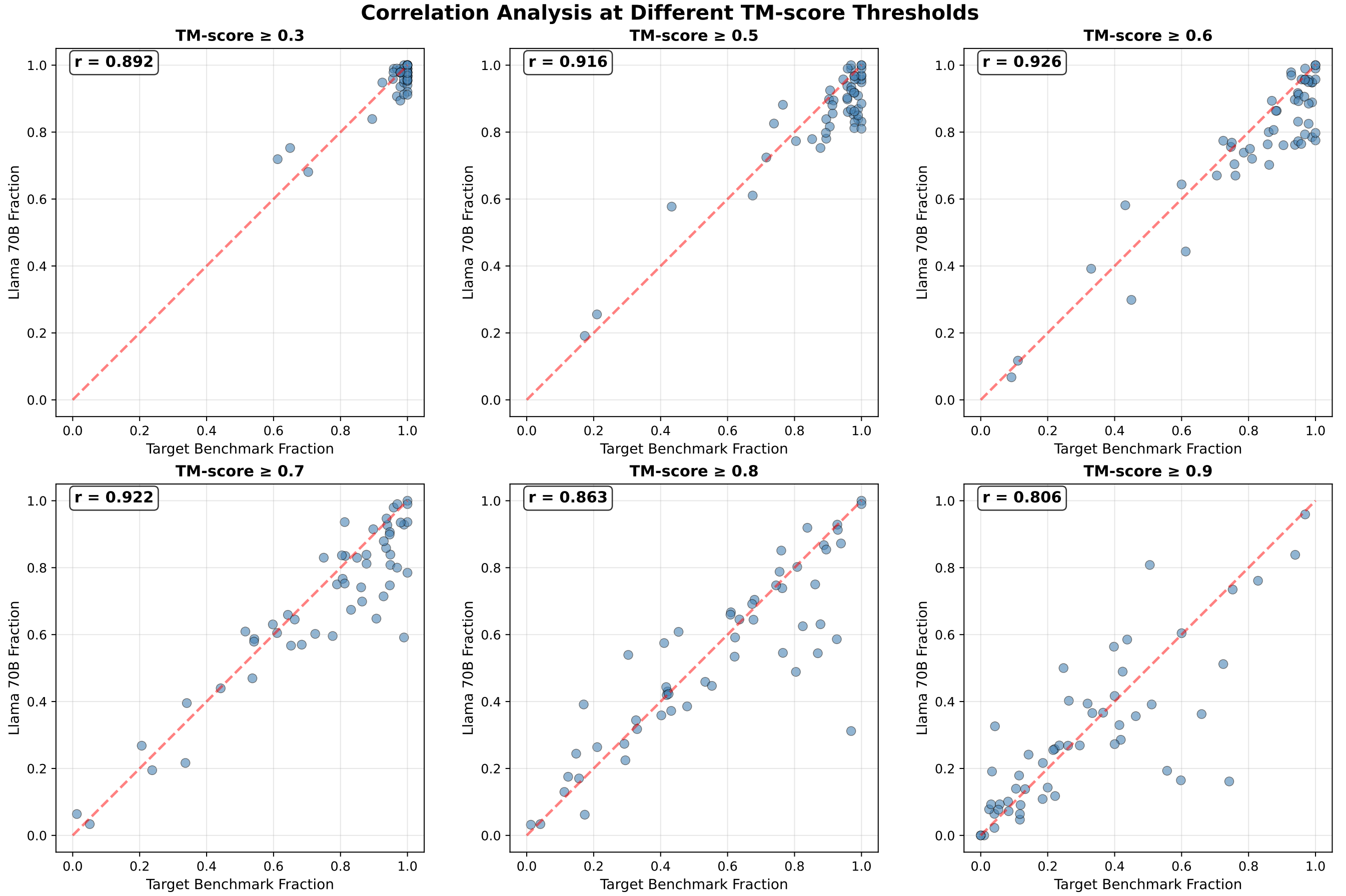


FIGURE 5. **Correlation analysis at different TM-score thresholds showing scatter plots of paired performance. Each panel represents a different threshold (0.3, 0.5, 0.6, 0.7, 0.8, 0.9) with target benchmark fraction on x-axis and Llama 70B fraction on y-axis. Red dashed lines indicate perfect correlation (y=x), and correlation coefficients (r) are displayed for each threshold. Data points represent individual GO term combination sets (n=54), with each point showing the fraction of sequences exceeding the specified TM-score threshold for both approaches.**

TABLE 3. **Performance comparison of generated protein sequences across different numbers of reference sequences.**

Metric	1 Ref	2 Ref	3 Ref	4 Ref	5 Ref
Total seqs with TM-score	91	99	97	98	98
max(T)	0.966	0.967	0.978	0.974	0.974
min(T)	0.656	0.304	0.724	0.655	0.174
med(T)	0.870	0.858	0.869	0.865	0.867
mean(T)	0.868	0.848	0.871	0.851	0.863
$\sigma(T)$	0.062	0.091	0.060	0.072	0.091
$\theta = 0.8$	82 (90.1%)	81 (81.8%)	82 (84.5%)	79 (80.6%)	85 (91.4%)

Table 3 presents results and statistics for different N_R. We can observe that:

- Performance is relatively high across all reference sequence settings
- Using 5 reference sequences, i.e. $N_R = 5$, yields slightly higher performance for $\theta = 0.8$.

N_R was therefore set to 5 in the two main experiments. This is also consistent with recent findings in the field of few-shot learning and in-context example selection [31] that demonstrates general performance increase with a relatively small number (i.e. five) in-context examples. However it is known that estimation error tends to decrease as the number of training examples increases [32], so it would be interesting in future to explore $N_R > 5$ when suitable larger context-window pre-trained models become available.

VI. DISCUSSION

A. IMPLICATIONS

This research study demonstrates the potential of reference-guided controllability for protein sequence generation. By adopting Gene Ontology (GO) terms to describe target functions, and making use of known-good reference sequences, the proposed framework provides a scalable, flexible and practical approach to controllable protein generation without extensive model retraining – something that could have positive implications for synthetic biology, protein engineering, and drug discovery.

One key applications of this research is for targeted therapeutic design, where specific protein properties, such as binding affinity and stability, can be precisely controlled. By incorporating GO terms that define molecular functions and biological processes, researchers can generate protein sequences optimized for specific applications, such as enzyme design, receptor targeting, or antibody development. This enables the rapid generation of functional proteins tailored to biomedical needs.

Another potential impact is in functional protein exploration, where novel protein sequences with desirable characteristics can be generated and analyzed. The integration of LLMs with in-context learning offers a data-efficient method for exploring the protein sequence space, potentially assisting in discoveries of biomaterials, industrial enzymes, and biosensors without necessitating excessive trial-and-error experimentation.

B. LIMITATIONS

While the proposed framework demonstrates controllable generation of protein sequences with relatively high structural quality, several limitations remain:

1) **Computational constraints**: Large-scale models, such as Llama 3.1 70B, require substantial computational resources, making them inaccessible for some research groups. In future work, we hope to explore optimization strategies to improve sustainability without compromising performance.
2) **Functional validation gap**: Although our method generates structurally viable proteins, functional validation through wet-lab experiments is still required. Structural similarity (TM-score) provides a strong indication of function, however actual biochemical activity must be experimentally confirmed.
3) **Dependence on reference sequences**: The effectiveness of in-context learning depends on the quality and diversity of reference sequences. If the available reference proteins do not adequately capture the desired functional space, generation may be biased or suboptimal.
4) **Limited adaptability to highly novel functions**: While our approach enables controllability, it remains constrained by the existing protein sequence space. Generating completely novel protein folds that do not have well-defined GO term references, and which may not be well represented in the ground truth datasets, remains an unsolved challenge in the field.

C. FUTURE DIRECTIONS

To further enhance the impact of reference-guided controllability in protein sequence generation, we propose several future directions for research:

1) **Optimization for smaller, task-specific models**: While large-scale models provide impressive performance, developing smaller, fine-tuned models for specific protein classes could improve accessibility and computational efficiency.
2) **Expanding the evaluation pipeline**: Future work could incorporate functional assays and molecular dynamics simulations to assess the stability, binding affinity, and enzymatic activity of generated proteins. This would provide a more comprehensive validation beyond structural similarity.

VII. CONCLUSION

This paper has presented a framework for the generation of protein sequences to meet specific target functionalities. We have utilized capable LLMs steered via in-context learning and textual functional descriptions to create novel protein sequences that meet target specifications on demand. The textual target functional property descriptors are specified using GO terms, making them both human readable and compatible with a large body of research literature. An evaluation pipeline is also developed to assess quality, functionality, and diversity of the generated protein sequences through TM score comparison.

The framework has been tested with a range of LLMs, offering useful insights into the capability of LLM structures to yield controllable proteins for specific targets. It is potentially compatible with many other LLMs.

The proposed framework has also been assessed with a variety of GO term sets. It has been shown capable of accurate protein sequence generation to meet target functional properties. The hope is that this contribution helps to advance the field of computational protein engineering by bridging the gap between structural control and functional specificity in AI-driven protein design.

VIII. CODE AVAILABILITY

The evaluation code used in this study is publicly available at https://github.com/NVIDIA/nvaitc-aps-project-demos/tree/main/projects/progo-protein-evaluation-tm-plddt.

REFERENCES

[1] X. Pan and T. Kortemme, "Recent advances in de novo protein design: Principles, methods, and applications," *Journal of Biological Chemistry*, vol. 296, 2021.

[2] Z. Y. Weinberg, S. S. Soliman, M. S. Kim, D. H. Shah, I. P. Chen, M. Ott, W. A. Lim, and H. El-Samad, "De novo-designed minibinders expand the synthetic biology sensing repertoire," *bioRxiv*, 2024.

[3] X. Yan, X. Liu, C. Zhao, and G.-Q. Chen, "Applications of synthetic biology in medical and pharmaceutical fields," *Signal transduction and targeted therapy*, vol. 8, no. 1, p. 199, 2023.

[4] H. Lu, Z. Cheng, Y. Hu, and L. V. Tang, "What can de novo protein design bring to the treatment of hematological disorders?" *Biology*, vol. 12, no. 2, p. 166, 2023.

[5] M. Mardikoraem, Z. Wang, N. Pascual, and D. Woldring, "Generative models for protein sequence modeling: recent advances and future directions," *Briefings in Bioinformatics*, vol. 24, no. 6, p. bbad358, 2023.

[6] A. V. Bryksin, A. C. Brown, M. M. Baksh, M. Finn, and T. H. Barker, "Learning from nature–novel synthetic biology approaches for biomaterial design," *Acta biomaterialia*, vol. 10, no. 4, pp. 1761–1769, 2014.

[7] N. Anand, R. Eguchi, I. I. Mathews, C. P. Perez, A. Derry, R. B. Altman, and P.-S. Huang, "Protein sequence design with a learned potential," *Nature communications*, vol. 13, no. 1, p. 746, 2022.

[8] J. Luo, X. Liu, J. Li, Q. Chen, and J. Chen, "Flexible and Controllable Protein Design by Prefix-tuning Large-Scale Protein Language Models."

[9] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig *et al.*, "Gene ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[10] T. Consortium, S. A. Aleksander, J. Balhoff, S. Carbon, J. Cherry, H. J. Drabkin, D. Ebert, M. Feuermann, P. Gaudet, and N. L. Harris, "The gene ontology knowledgebase in 2023," *Genetics*, vol. 224, no. 1, p. iyad031, 2023.

[11] N. Ferruz, S. Schmidt, and B. Höcker, "ProtGPT2 is a deep unsupervised language model for protein design," *Nature Communications*, vol. 13, no. 1, p. 4348, Jul. 2022, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41467-022-32007-7

[12] E. Nijkamp, J. A. Ruffolo, E. N. Weinstein, N. Naik, and A. Madani, "ProGen2: Exploring the boundaries of protein language models," *Cell Systems*, vol. 14, no. 11, pp. 968–978.e3, Nov. 2023, publisher: Elsevier. [Online]. Available: https://www.cell.com/cell-systems/abstract/S2405-4712(23)00272-7

[13] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, N. Smetanin, R. Verkuil, O. Kabeli, Y. Shmueli *et al.*, "Evolutionary-scale prediction of atomic-level protein structure with a language model," *Science*, vol. 379, no. 6637, pp. 1123–1130, Mar. 2023, publisher: American Association for the Advancement of Science. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.ade2574

[14] L. Lv, Z. Lin, H. Li, Y. Liu, J. Cui, C. Y.-C. Chen, L. Yuan, and Y. Tian, "ProLLaMA: A Protein Language Model for Multi-Task Protein Language Processing," Jul. 2024, arXiv:2402.16445 [cs]. [Online]. Available: http://arxiv.org/abs/2402.16445

[15] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig *et al.*, "Gene Ontology: tool for the unification of biology," *Nature genetics*, vol. 25, no. 1, pp. 25–29, May 2000. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3037419/

[16] S. A. Aleksander, J. Balhoff, S. Carbon, J. M. Cherry, H. J. Drabkin, D. Ebert, M. Feuermann, P. Gaudet, N. L. Harris, D. P. Hill *et al.*, "The Gene Ontology knowledgebase in 2023," *Genetics*, vol. 224, no. 1, p. iyad031, Mar. 2023. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10158837/

[17] R. Rentzsch and C. A. Orengo, "Protein function prediction using domain families," *BMC Bioinformatics*, vol. 14, no. Suppl 3, p. S5, Feb. 2013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3584934/

[18] G. L. Holliday, R. Davidson, E. Akiva, and P. C. Babbitt, "Evaluating Functional Annotations of Enzymes Using the Gene Ontology," *Methods in molecular biology (Clifton, N.J.)*, vol. 1446, pp. 111–132, 2017. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5837055/

[19] L. Middendorf, B. Ravi Iyengar, and L. A. Eicholt, "Sequence, structure and functional space of drosophila de novo proteins," *bioRxiv*, pp. 2024–01, 2024.

[20] M. Littmann, M. Heinzinger, C. Dallago, T. Olenyi, and B. Rost, "Embeddings from deep learning transfer go annotations beyond homology," *Scientific reports*, vol. 11, no. 1, p. 1160, 2021.

[21] S. Das, D. Lee, I. Sillitoe, N. L. Dawson, J. G. Lees, and C. A. Orengo, "Functional classification of CATH superfamilies: a domain-based approach for protein function annotation," *Bioinformatics*, vol. 31, no. 21, pp. 3460–3467, Nov. 2015. [Online]. Available: https://doi.org/10.1093/bioinformatics/btv398

[22] Q. Dong, L. Li, D. Dai, C. Zheng, J. Ma, R. Li, H. Xia, J. Xu, Z. Wu, B. Chang *et al.*, "A Survey on In-context Learning," Jun. 2024, arXiv:2301.00234 [cs] version: 4. [Online]. Available: http://arxiv.org/abs/2301.00234

[23] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and U. Consortium, "Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches," *Bioinformatics*, vol. 31, no. 6, pp. 926–932, 2015.

[24] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The protein data bank," *Nucleic acids research*, vol. 28, no. 1, pp. 235–242, 2000.

[25] M. van Kempen, S. S. Kim, C. Tumescheit, M. Mirdita, J. Lee, C. L. M. Gilchrist, J. Söding, and M. Steinegger, "Fast and accurate protein structure search with Foldseek," *Nature Biotechnology*, vol. 42, no. 2, pp. 243–246, Feb. 2024, publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41587-023-01773-0

[26] Llama Team, "The llama 3 herd of models," 2024, accessed: Jan. 31, 2025. [Online]. Available: https://arxiv.org/abs/2407.21783

[27] L.-y. Wu, X.-y. Xia, and X.-m. Pan, "A novel score for highly accurate and efficient prediction of native protein structures," *bioRxiv*, pp. 2020–04, 2020.

[28] J. Xu and Y. Zhang, "How significant is a protein structure similarity with TM-score = 0.5?" *Bioinformatics*, vol. 26, no. 7, pp. 889–895, Apr. 2010. [Online]. Available: https://academic.oup.com/bioinformatics/article/26/7/889/213219

[29] Y. Zhang and J. Skolnick, "Scoring function for automated assessment of protein structure template quality," *Proteins: Structure, Function, and Bioinformatics*, vol. 57, no. 4, pp. 702–710, Dec. 2004. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/prot.20264

[30] C. Anil, E. Durmus, M. Sharma, J. Benton, S. Kundu, J. Batson, N. Rimsky, M. Tong, J. Mu, D. Ford *et al.*, "Many-shot Jailbreaking."

[31] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What Makes Good In-Context Examples for GPT-3?" in *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, E. Agirre, M. Apidianaki, and I. Vulić, Eds. Dublin, Ireland and Online: Association for Computational Linguistics, May 2022, pp. 100–114. [Online]. Available: https://aclanthology.org/2022.deelio-1.10/

[32] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language Models are Few-Shot Learners," Jul. 2020, arXiv:2005.14165 [cs]. [Online]. Available: http://arxiv.org/abs/2005.14165

[33] C. Pesquita, D. Faria, H. Bastos, A. E. Ferreira, A. O. Falcão, and F. M. Couto, "Metrics for GO based protein semantic similarity: a systematic evaluation," *BMC Bioinformatics*, vol. 9, no. 5, p. S4, Apr. 2008. [Online]. Available: https://doi.org/10.1186/1471-2105-9-S5-S4

APPENDIX A GO TERM SELECTION

For the second experiment, we curated a set of 100 functionally diverse protein structure sets based on Gene Ontology (GO) term combinations, that are designed to provide coverage of the diverse protein function space while ensuring sufficient structural data for robust evaluation. The general selection process is shown in Figure 6.

Firstly, we extract protein sequences in the UniRef50 database that are annotated with one or more GO terms and list all unique GO term combinations across the entire dataset (Note: the order of GO terms is not taken into account, i.e. $G = \{A, B, C\}$ and $G = \{A, C, B\}$ are equivalent). In total, 259,160 distinct GO term combinations were noted.

Next, the occurrence frequency of each unique GO term combination was computed. Rare combinations lack sufficient support for downstream statistical analysis, and hence a first stage filtering process is used to remove combinations represented by fewer than 50 examples. The choice of cutoff is outlined in Section A-1). After filtering, 2,206 combinations met the inclusion criterion.

The qualifying combinations were sorted by frequency and divided into 100 equal sized bins, i.e. each bin contains approximately 22 GO term combinations and represents a distinct frequency range.

To ensure uniform sampling across the frequency distribution, one GO term combination is selected from each bin. A search of PDB is made to ensure that there are at least 50 associated protein structures corresponding to the chosen GO term combination and if not, another combination is chosen from the same bin (in the first stage filtering process, we filtered by UniRef50, in this second stage of filtering, we are using PDB occurrence). Section A-1) discusses this cutoff choice too. The selection algorithm operates as follows:

1) Begin at the midpoint of each frequency bin
2) Query UniProt API for structure count
3) If structure count >=50, select this combination

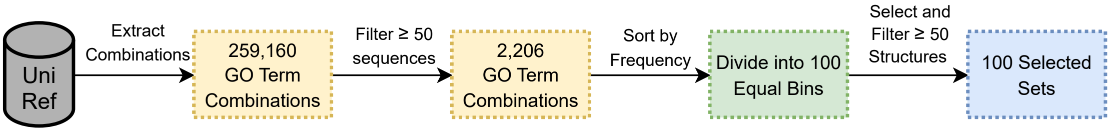

Uni Ref
Extract Combinations
259,160 GO Term Combinations
Filter ≥ 50 sequences
2,206 GO Term Combinations
Sort by Frequency
Divide into 100 Equal Bins
Select and Filter ≥ 50 Structures
100 Selected Sets

FIGURE 6. Overview of GO term Selection Pipeline, with two stage filtering of target GO term combinations.

4) Otherwise, check adjacent positions (±1, ±2, etc.) until a qualifying combination is found

This approach minimizes deviation from the frequency-based selection while ensuring structural data availability, resulting in 100 sets of unique GO terms with 42 two-term combinations and 58 three-term combinations.

We note that Pesquita & Faria et. al [33] systematically evaluated GO-based semantic similarity measures, finding that many proteins in curated datasets typically associated with just a small number of functional annotations. Most proteins do not map to a large set of GO terms, and their data suggests that just two or three annotations will often suffice to describe the core molecular functions. Our findings, based on occurrence frequency, are in alignment with the findings in [33].

In Section II-C we noted that GO terms are categorized into three standard divisions. To assess the filtering process described above, we compare the statistics of the final 100 selected GO term sets against the original UniRef50 dataset. The comparison shows that the pipeline generally preserves the natural functional distribution. Molecular functions show a modest increase from 25.8% to 35.7% (+9.9%), biological processes decrease from 51.6% to 40.3% (–11.2%), while cellular components remain nearly unchanged (22.6% to 24.0%, +1.4%). The selected sets thus maintain a representative distribution across the GO term categories.

1) Sequence and Structure Thresholds

As described above, the selection pipeline uses two separate filtering processes:

1) **Sequence thresholding**: ensures GO term combinations appear in >=50 proteins.
2) **Structural thresholding**: ensures GO term combinations are found in >=50 PDB structures.

The first stage filtering applies a **Sequence Threshold** of >=50 protein sequences to remove rarely occurring GO term combinations. Analysis of 37,145 proteins from the UniRef50 database revealed 9,356 unique GO term combinations. The frequency distribution of these combinations exhibits extreme skewness, with 9,257 combinations (98.9%) appearing in fewer than 50 proteins and only 99 combinations (1.1%) meeting or exceeding this threshold.

To examine further, Figure 7 plots the median frequency for each of 100 bins with (green line) and without (red line) applying the 50-sequence filter. The first 72 unfiltered bins (72% of all bins) contain only combinations that appear in a single protein (median frequency = 1). Even by bin 100, the median frequency reaches only 63. We hypothesize that the majority of single sample bins represent spurious or overly specific functional annotations rather than biologically meaningful categories.

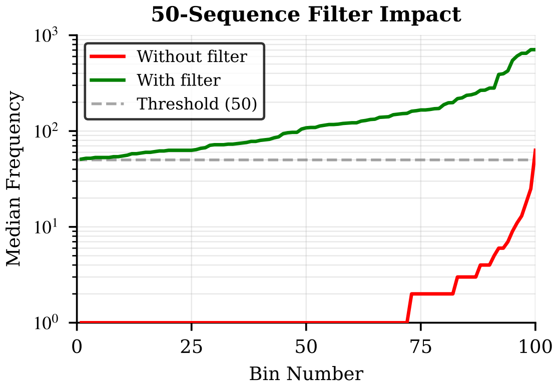


FIGURE 7. Impact of the 50-sequence filter on frequency binning. Frequency distribution of 9,356 unique GO term combinations from UniRef50. Without filtering (red), most bins are dominated by singletons (72% with median = 1), reflecting extreme skewness. Applying a >=50 sequence threshold (green) removes rare combinations, yielding median frequencies of 51–707 and enabling more uniform sampling.

With the filter applied, all bins exceed the threshold with median frequencies ranging from 51 to 707, enabling uniform, and more meaningful, sampling across the frequency spectrum.

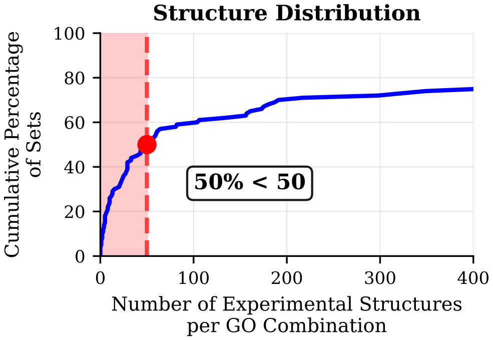


FIGURE 8. Cumulative percentage of GO term combinations (y-axis) with at least the given number of protein structures available (x-axis).

For the second stage filtering, a **Structure Threshold** of >=50 protein structures in PDB is applied when choosing the GO term combinations. The cumulative distribution function in Figure 8 shows the percentage of GO combinations with structure counts less than or equal to each x-axis value. At the 50 structure point marked, about 50% of sets fall below this threshold. The gradient change at this point indicates a transition between two distinct populations of GO term combinations.

The cumulative distribution provides quantitative evidence for the 50 structure threshold through four observations: (1) the steep initial rise captures 50% of sets within the first 50 structures, indicating concentration of structurally under-characterized combinations; (2) the slope change at x=50 suggests a natural boundary between populations; (3) the 50% exclusion rate balances data quality requirements with retention of sufficient test sets; (4) Without a threshold, sets with 0 structures are included, which cannot be used for a target benchmark.

2) Valid GO Term Combinations for Evaluation

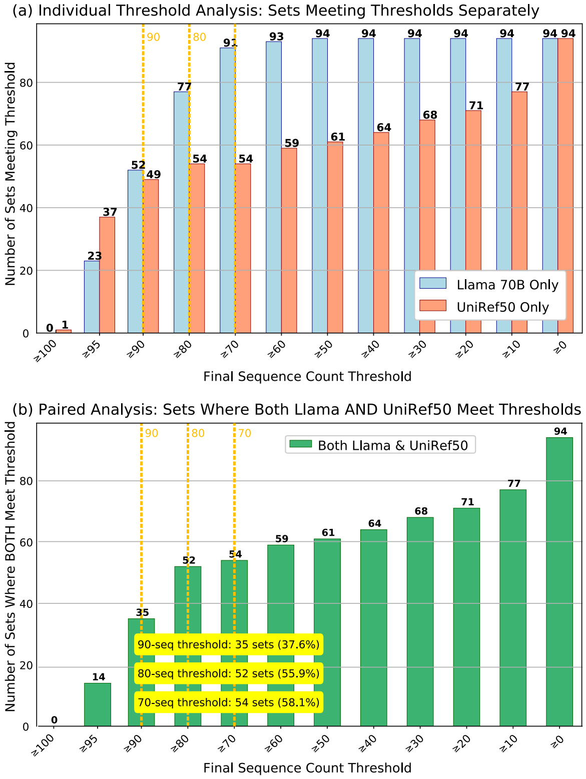
For each of the 100 selected GO term combinations, the proposed framework requires a target benchmark set of natural protein sequences from UniRef50, having corresponding GO term combinations.

This same pool also serves as the source of reference sequences (N_R) for the generation set. This requires a minimum of five sequences per GO term combination (ie. $N_R = 5$).

In addition to the limitations imposed by UniRef50 coverage, not all selected combinations resulted in a full set of 100 valid candidate sequences. Some are excluded due to recall, and ESMFold fails to return valid 3D structures for some generated sequences, particularly those that are excessively long and are outside ESMFold's maximum input length. As a result, final usable sequence counts vary across sets.

To define a reliable threshold for inclusion, we analyzed how many GO term combinations remained under varying minimum sequence count requirements. The goal was to identify a balance between dataset size and representativeness.

To guide the threshold selection, Figure 9 plots number of valid candidate sequences for Llama 70B and UniRef50 known-good sequences respectivlely, against the final sequence count threshold. From this, we selected a threshold of 70 sequences per GO term combination. This enabled retention of 54 out of the 100 original sets, striking a reasonable balance between diversity of GO term coverage and availability of sufficient sequences for structural evaluation. Lower thresholds risk including sets where UniRef50 provides only sparse representation, undermining the reliability of structure-based comparisons. Notably, the number of usable sets plateaus around the 70-sequence mark, suggesting a natural cutoff point given current UniRef50 annotations.


(a) Individual Threshold Analysis: Sets Meeting Thresholds Separately

(b) Paired Analysis: Sets Where Both Llama AND UniRef50 Meet Thresholds

FIGURE 9. (a) Individual threshold analysis showing sets where each approach separately meets thresholds, demonstrating Llama 70B (blue) versus UniRef50 (orange). (b) Paired threshold analysis showing sets where BOTH approaches meet thresholds (green), revealing the true sample sizes available for fair comparisons.

DARREN TAN is a Solutions Architect at the NVIDIA AI Technology Centre (NVAITC) in Singapore. His work focuses on large language models and generative AI applications, including digital biology, driving the adoption of AI applications in joint industry and academia settings.

IAN MCLOUGHLIN is a Professor in the Infocomm Technology Cluster at the Singapore Institute of Technology. He is a Chartered Engineer, a Fellow of the Institution of Engineering and Technology, and a Senior Member of the IEEE. His research and teaching span AI for speech and audio, signal processing and embedded systems, with over 300 publications and patents.

AIK BENG NG leads the regional team at the NVIDIA AI Technology Centre (Asia Pacific South) and serves as Co-Director of the SITxNVIDIA AI Centre (SNAIC). He is an Adjunct Associate Professor at the Singapore Institute of Technology and a member of Singapore's National AI Technical Committee. His interests include accelerating AI adoption through industry partnerships and developer enablement.

ZHENGKUI (DANIEL) WANG is an Associate Professor in the Infocomm Technology Cluster at the Singapore Institute of Technology. He co-directs the SITxNVIDIA AI Centre (SNAIC) and the Data Science & AI Lab (DSAIL). His work focuses on foundation models, generative AI and applications spanning NLP and imaging.

ABRAHAM C. STERN leads product management for NVIDIA BioNeMo, the company's platform for AI-enabled drug discovery. He previously managed NVIDIA Clara Discovery and holds a Ph.D. in computational chemistry from the University of South Florida, with postdoctoral work at the University of California, Irvine. His interests lie at the intersection of scientific computing, machine learning, and computational chemistry.

SIMON SEE is Solution Architecture & Engineering Senior Director, Chief Solution Architect, and Global Head for the NVIDIA AI Technology Center. He also holds professorial or adjunct appointments at universities in China and Indonesia, and serves as an adjunct scientist at A*STAR's Institute of High Performance Computing. His interests include high performance computing and AI systems at scale.