PROTEINGPT: MULTIMODAL LLM FOR PROTEIN PROPERTY PREDICTION AND STRUCTURE UNDERSTANDING

Yijia Xiao[1], **Edward Sun**[1], **Yiqiao Jin**[2], **Qifan Wang**[3], **Wei Wang**[1]
[1]University of California, Los Angeles, [2]Georgia Institute of Technology, [3]Meta AI
{yijia.xiao,wei.wang}@cs.ucla.edu, edwardsun12895@g.ucla.edu,
yjin328@gatech.edu, wqfcr@fb.com

ABSTRACT

Understanding biological processes, drug development, and biotechnological advancements requires a detailed analysis of protein structures and functions, a task that is inherently complex and time-consuming in traditional protein research. To streamline this process, we introduce ProteinGPT, a state-of-the-art multimodal large language model for proteins, which allows users to upload protein sequences and/or structures for comprehensive proteins analysis and responsive inquiries. ProteinGPT seamlessly integrates protein sequence and structure encoders with linear projection layers to ensure precise representation adaptation. It leverages a large language model (LLM) to generate accurate and contextually relevant responses. To train ProteinGPT, we construct a large-scale dataset of 132,092 proteins, each annotated with 20-30 property tags and 5-10 QA pairs per protein, and optimized the instruction-tuning process using GPT-4o. Experiments demonstrate that ProteinGPT effectively generates informative responses to protein-related questions, achieving high performance on both semantic and lexical metrics. It significantly outperforms baseline models and general-purpose LLMs in understanding and responding to protein-related queries.

1 INTRODUCTION

Proteins are fundamental molecular building blocks of life, playing critical roles in biological processes Kitadai & Maruyama (2018). Understanding their structure, functions, and interactions is vital for advancements in drug discovery Teague (2003), healthcare Organization & University (2007), and biological/medical engineering Kobsa & Saltzman. Recent breakthroughs in machine-learning-based protein structure and function prediction Lin et al. (2023) have significantly accelerated biological research by reducing the reliance on traditional labor-intensive laboratory experiments and literature search.

Challenges. As proteins can be represented by strings of characters, each corresponding to an amino acid from an alphabet of 20 letters, recent advancements in Large Language Models (LLMs) have naturally extended to protein research. Existing protein LLMs such as ProtST Xu et al. (2023), ProteinChat Guo et al. (2023), and ProtChatGPT Wang et al. (2024) focus primarily on sequence-based or structure-based modeling, limiting their ability to generate holistic protein insights from multiple modalities. For instance, protein sequences can reveal evolutionary information, functional sites, and sequence-structure relationships, while protein structures provide critical insights into spatial arrangement, structural dynamics, binding sites, and stability. Applying multimodal LLMs to protein modeling is non-trivial due to the challenge in aligning diverse modalities, such as textual descriptions, protein sequences, and protein structures. Meanwhile, direct end-to-end retraining for protein LLMs is usually impractical due to extensive requirements for annotated data.

Our Work. We propose ProteinGPT, a protein LLM that allows researchers to upload protein sequences and/or structures (via fasta or PDB files) and ask natural language questions. ProteinGPT consists of four major components: a protein sequence encoder, a protein structure encoder, a projection layer, and an LLM.

The *protein sequence encoder* is based on the ESM-2 (Evolutionary Scale Modeling 2) Lin et al. (2023) model variant `esm2_t36_3B_UR50D` with 36 transformer layers and 3 billion pa-

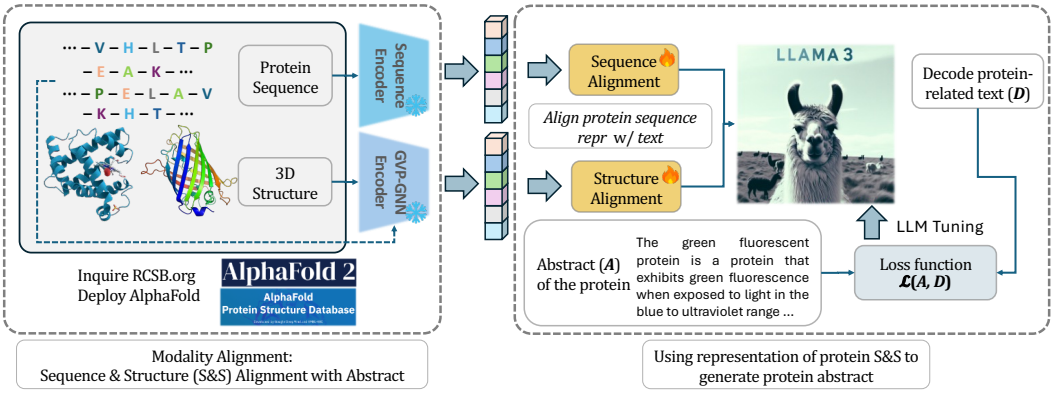

Figure 1: ProteinGPT Modality Fusion & Alignment Stage: we freeze the encoder blocks and train the linear project layer to learn how to align protein structure and protein sequence representations with text. In the alignment stage, the input to the training is only the projected protein representation. No text prompts are incorporated in this stage.

rameters. Pretrained on `UniRef50/D` Suzek et al. (2015), a comprehensive protein database that clusters sequences with at least 50% sequence identity and 80% coverage, this model ensures sequence diversity and informativeness in encoding. The *protein structure encoder*, `esm_if1_gvp4_t16_142M_UR50`, is an inverse folding model that incorporates a geometric input processing layer paired with a seq2seq transformer Bahdanau (2015). Trained on 12 million structures predicted by AlphaFold2 Jumper et al. (2021), the model effectively captures protein structural information.

To bridge these encoders with the LLM, we introduce a *projection layer* that aligns their embeddings with the LLM's latent space. This enables seamless integration of multimodal protein representations into the LLM, enabling information extraction from not only the protein structural and sequential information but also the rich pretrained knowledge within ESM.

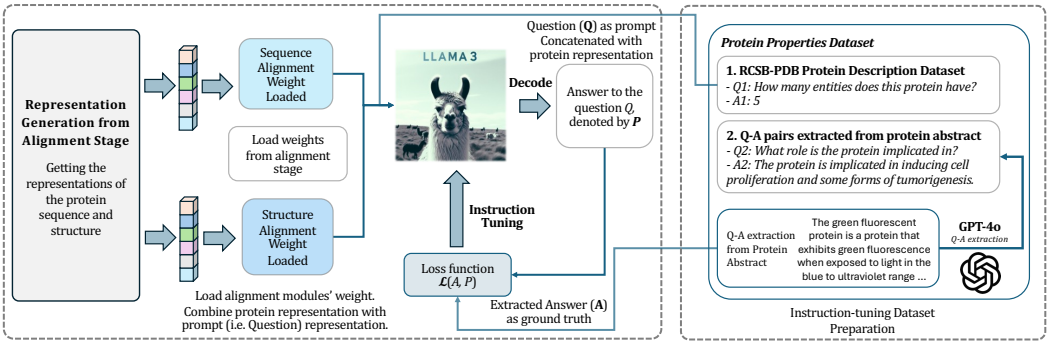

Figure 2: ProteinGPT Instruction Tuning Stage: we utilize the QA pairs and property tags in ProteinQA to tune the LLM to follow instructions and give concise responses. For instruction alignment, explicit prompts (*Questions* on the protein) are included at the beginning of the prompt.

To train ProteinGPT for effective modality alignment, we introduce ProteinQA, a large-scale dataset of over 132,092 protein sequences with structural information and annotations. Unlike previous works that use entire protein annotations as prediction targets for instruction-tuning, we use GPT-4o OpenAI et al. (2024) to systematically decompose proteins' abstract summary from the RCSB Protein Data Bank (RCSB PDB) Burley et al. (2023) into question-answer (QA) pairs. We then finetune ProteinGPT on ProteinQA using diverse open-source models, including Llama-2 Touvron et al. (2023), Llama-3 Dubey et al. (2024) Vicuna Chiang et al. (2023), and Mistral Jiang et al. (2023).

The training effectively enhances the model's ability to understand user queries and generate concise, contextually relevant answers.

Contributions. Our contributions are as follows:

- **Novel Framework.** We introduce ProteinGPT, a state-of-the-art protein LLM that fuses protein sequence and structural information to enable interactive protein-focused conversations, significantly enhancing the understanding and design of proteins;
- **Large-scale Dataset.** We propose ProteinQA, a large-scale protein dataset based on RCSB-PDB Guo et al. (2023). ProteinQA encompasses 132,092 protein samples, each annotated with a detailed descriptive abstract, 20-30 property tags, and 5-10 QA pairs. The depth and variety of these annotations position ProteinQA as a high-quality instruction tuning corpus;
- **Comprehensive Experiments.** We conducted extensive experiments on mainstream open-source and proprietary LLM backbones under different scenarios. Our empirical analysis provides guidance for future design of protein LLMs.

2 METHODOLOGY

2.1 MODEL ARCHITECTURE

ProteinGPT consists of two frozen pre-trained encoders (Figures 1 and 2): an inverse folding model (`esm_if1_gvp4_t16_142M_UR50`) for structure encoding and a protein language model for sequence encoding (`esm2_t36_3B_UR50D`). The embeddings generated by these models are fed into a linear projection layer to produce soft prompts for the LLM. The model training comprises two stages: 1) *Sequential and Structural Alignment* and 2) *Instruction-tuning*.

2.1.1 SEQUENCE AND STRUCTURE ALIGNMENT

In the alignment stage, protein structures are first fed into the pre-trained structure encoder `esm_if1_gvp4_t16_142M_UR50` which explicitly captures the detailed 3D structures and models spatial interactions between amino acid residues. Then, sequences are encoded using the sequence encoder `esm2_t36_3B_UR50D` featuring 36 transformer layers and 3 billion parameters, trained on the Protein `UniRef50/D` database to enhance sequence diversity. This module integrates structural information with implicit structural contact, evolutionary, and biochemical information that the structure alone does not capture. For efficient training, both of these modules are frozen. We utilize a specialized token prompt for protein-text modality alignment:

$$\mathbf{Q}: < Protein >< Struct >< Seq >$$
$$< /Protein >< QuestionPrompts >$$
$$\mathbf{A}: < Description >$$

The structural and sequential information is encoded into the soft prompts and prepended to the question prompt. In stage 1 training, the question prompt **Q** is left empty to prioritize learning the abstract description from the protein representation.

The description tag is then replaced with the full annotation from RCSB-PDB Guo et al. (2023) to train the projection layer in aligning a protein with its annotation description.

2.1.2 INSTRUCTION-TUNING

In stage 2, the model undergoes instruction tuning using our curated QA dataset. Unlike previous works that utilize full annotations, we focus on specific QA examples to facilitate instruction tuning. We augment the abstract dataset from stage 1 using GPT-4o to generate explicit QA pairs for this stage. The prompts from stage 1 are adapted to the LLaMA style ("`### Human:''` ... and `### Assistant:` ...), with **Q** replaced by explicit questions from the QA dataset, such as "how many assemblies does this protein have." The model then generates descriptive yet concise answers from the given protein as **A**.

2.2 DATASET CURATION

To ensure the highest quality of training data, we implement a rigorous validation and data collection process.

We leverage RCSB-PDB dataset Berman (2000), which is thoroughly vetted by both RCSB-PDB Berman (2000) and PubMed[1] for reliable protein information. The dataset is derived from peer-reviewed PubMed publications, implying that the dataset content is highly accurate. We select a large dataset to cover a wide range of proteins and ensures coverage on out-of-distribution proteins. For modality alignment, we construct a large-scale dataset from the RCSB-PDB database Berman (2000) consisting of 132,092 protein structures, sequences, and abstract descriptions. The raw dataset of 204,826 proteins is filtered to retain only those with an abstract description, chain A, and sequences without non-encodable characters. Each entry in the final dataset includes the 3D protein structure represented by backbone atomic coordinates, the sequence string, and a rich protein annotation, as shown in Figure 2. The detailed statistics of our dataset are presented in Table 1, highlighting the extensive annotations and comprehensive content available for each protein.

Table 1: Statistics for ProteinQA

Per Protein	**Min**	**Max**	**Mean**
# Abstracts tokens	89	728	205.45
# Answer tokens	32	550	98.56
# Open-Ended QA Pair samples	10	26	17.39
# Closed-Ended QA Pair samples	24	29	25.94

2.3 Data Augmentation

Previous works often use the entire protein annotation for instruction tuning Xu et al. (2023); Guo et al. (2023); Wang et al. (2024), which may result in the model producing overly detailed responses with extraneous information not directly relevant to the user prompt. Therefore, our ProteinGPT decomposes the rich protein annotations into more specific QA-pairs for instruction tuning so that user instructions can be concisely answered. We do this by generating short concise answers along with long-form responses to include our dataset. Concretely, we prompt GPT-4o to generate both open-ended and close-ended QA pairs with the context of the abstract to decompose the abstract into atom-level QA pairs. As seen in Table 1, on average, each protein has around 40 total QA pairs generated from this process.

3 Experiments

3.1 Training

We trained ProteinGPT on 4 base LLM architectures: Vicuna Chiang et al. (2023), LLaMA-2 Zhang et al. (2023a), LLaMA-3 Dubey et al. (2024), and Mistral Jiang et al. (2023). Our training process is divided into two phases: modality alignment (MA) and instruction tuning (IT). This approach allows the model to preserve previously acquired knowledge while effectively handling specific instructions, such as protein-related queries.

Stage I: Modality Fusion/Alignment (MA). In this stage, we focus exclusively on training the projection adapter by freezing both sequence and structure encoders. We set the maximum text length of abstracts to 384 characters to accommodate the annotation lengths within the RCSB-PDB dataset. The projection layer is trained over 10 epochs with a batch size of 1, weight decay of 0.05, and 2048 warm-up steps. The dataset is divided into a training set (70%) of 105,673 proteins and a testing set (30%) of 26,419 proteins. We utilize the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.98$ Loshchilov & Hutter (2019), and employ a learning rate scheduler with a linear warm-up followed by cosine annealing. We set the initial learning rate to 1×10^{-4}, the minimum learning rate to 8×10^{-5}, and the warm-up learning rate to 1×10^{-6}. Automatic mixed precision (AMP) Micikevicius et al. (2018) was used to improve training efficiency.

Stage II: Instruction Tuning (IT). In this stage, the model is fine-tuned on a protein question-answering task. Training is conducted for 10 epochs with a batch size of 1, weight decay of 0.05, and 200 warm-up steps. The QA dataset used in this phase comprises approximately 3.7 million samples,

[1] https://pubmed.ncbi.nlm.nih.gov/

with around 35 questions per protein. We apply similar settings for the AdamW optimizer and AMP, but with a lower initial learning rate of 1×10^{-5}, a minimum rate of 1×10^{-6}, and a warm-up rate of 1×10^{-6}.

3.2 INFERENCE

In real-world scenarios, there are cases where only protein sequence or structure information is available. Under such scenarios, protein folding and protein inverse-folding models are applied to obtain the missing structure or sequence information.

For **missing or incomplete sequences**, we use the `ESM-IF1` inverse folding model to predict absent segments, generating protein sequences from backbone atom coordinates. This method is effective in filling sequence gaps where traditional approaches struggle.

For **missing structural data**, we apply `ESMFold` to predict 3D structures of missing regions, generating accurate 3D conformations without relying on multiple sequence alignments.

3.3 COMPUTATIONAL COST

Our training uses two NVIDIA H100 PCIe GPUs (80GB vRAM) and two NVIDIA A100 PCIe GPUs (40GB). We implemented strategies including Automatic mixed precision (AMP), optimized data loaders, asynchronous data processing, multi-GPU training. Stage 1 and 2 require approximately one week and 60 hours, respectively.

ProteinGPT is optimized for fast processing, handling user protein queries in approximately 50 seconds and generating 23-token responses in about 8 seconds. Its computational efficiency allows deployment on hardware ranging from GPUs like the Nvidia T4 with 16 GB memory to standard CPUs.

4 RESULTS

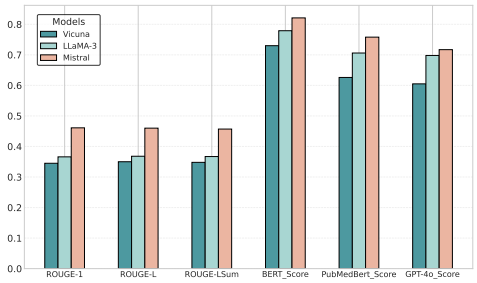

Figure 3: ProteinGPT performance with various base LLMs.

We conducted a series of experiments to assess ProteinGPT's effectiveness both quantitatively and qualitatively. Moreover, we benchmark the capabilities of state-of-the-art large language models on protein-related tasks.

4.1 QUALITATIVE EVALUATION

Figures 7, 9, 11 show example conversations between human users and ProteinGPT on Proteins 6O7Q, 5X1Y, and 7RUV, respectively. To ensure unbiased evaluation and avoid data leakage, all testing was performed on a separate set of proteins isolated from training. ProteinGPT effectively interpret the semantics of queries and produces accurate, logically consistent responses. These responses include details about protein *functions*, such as catalyzing the reduction of dinitrogen to ammonia, and *structures*, such as the structural dependencies on the substrate azide and the product ammonia. This demonstrates ProteinGPT's capabilities on protein sequence, structure, and function understanding tasks and its potential for enabling rapid exploration of proteins.

4.2 QUANTITATIVE EVALUATION

4.2.1 EXPERIMENTAL SETUP

We evaluate our model using a curated dataset of 3,508 randomly-selected question-answer-protein pairs, covering 160 proteins from the test split. Each protein is associated with 28-30 questions that were not seen by ProteinGPT during training. We benchmark against several baseline models, including vanilla open-source LLMs (without modality alignment or instruction tuning) and proprietary models (GPT 4o/4/3.5, o3/o1-mini, and DeepSeek-R1). To manage computational costs, we sample 1,025 questions (35 proteins with 28$\sim$30 questions per protein). For fairness of comparison, we employ standard metrics to compare model predictions with the ground truth. These include 1) **semantic similarity** metrics that measure contextual meanings: BERTScore (S_{BERT}) Zhang et al. (2020), PubMedBERT Score (S_{Pub}) Gu et al. (2021), and GPT Score (S_{GPT}) OpenAI (2023); 2) **lexical quality** metrics that measure surface-level similarity based on n-gram overlaps: ROUGE-1/2/L Ganesan (2018). For the GPT Score, we use OpenAI's `text-embedding-3-large` as the embedding model.

4.2.2 COMPARISON AMONG PROTEINGPT VARIANTS

Among the 4 variants, ProteinGPT$_{\text{Mistral}}$ performs the best in terms of both semantic and lexical metrics. In terms of BERTScore (Table 3), ProteinGPT$_{\text{Mistral}}$ achieves 0.829, followed by ProteinGPT$_{\text{LLaMA-3}}$ (0.790), ProteinGPT$_{\text{LLaMA-2}}$ (0.764), and ProteinGPT$_{\text{Vicuna}}$ (0.756). The strong performance of ProteinGPT$_{\text{Mistral}}$ can be attributed to its integration of sliding window attention (SWA) Jiang et al. (2023). Protein sequences are inherently lengthy and complex, often requiring models to capture intricate dependencies across extended stretches of amino acid. SWA helps capture local patterns and dependencies within protein sequences, leading to a longer effective attention span crucial for tasks like secondary structure prediction and functional annotation. The reduced computational load associated with SWA allows for the processing of longer protein sequences without a proportional increase in resource consumption. Figure 3 shows the visual comparison among different base LLMs of ProteinGPT.

4.2.3 BASELINE COMPARISON

We also compare ProteinGPT to three groups of baselines to demonstrate ProteinGPT's effectiveness in protein-specific multimodal tasks: 1) *Vanilla Open-source LLMs* (Vicuna, Mistral, LLaMA-3, and LLaMA-2); 2) *Proprietary General-Purpose LLMs* (GPT-4o/4/3.5); 3) *State-of-the-art models with Strong Reasoning Capabilities* (OpenAI o1/o3-mini and DeepSeek-R1). For fairness of comparison, we prepended the protein's FASTA sequence to the prompt to provide context for these LLMs. Tables 3 and 5 show model performance with semantic and lexical scores.

Vanilla open-source LLMs exhibit low semantic performance. When providing protein sequences as text input (Table 3a), the BERTScore (S_{BERT}) range from 0.490 (LLaMA-3) to 0.572 (Vicuna) in terms of precision, indicating a lack of protein-specific pretraining and limited semantic understanding of protein data. Meanwhile, incorporating modality fusion (where additional cues beyond the raw protein sequence are integrated) leads to modest improvements, such as an improvement from 0.572 to 0.582 when using Vicuna as the base model. Model Similarly, **proprietary models** such as GPT-4o/4/3.5, OpenAI o1/o3-mini, and DeepSeek-R1 also exhibit lower semantic and lexical performance when processing protein sequences as text inputs. This performance gap is likely due to these models not being pretrained on domain-specific data.

The results in Table 4 highlight a critical challenge in applying general-purpose language models to protein-related queries: standard models, even those with strong reasoning capabilities like o3-mini and DeepSeek-R1, struggle to interpret protein sequences. o3-mini achieves a marginally higher Rouge-L score (0.072) than GPT-4o (0.067), but remains far below ProteinGPT$_{\text{Mistral}}$ (0.460). DeepSeek-R1 performs particularly poorly, which suggests that it struggles to extract meaningful insights from protein sequences when they are formatted as text. The sample answers in Appendix E.3 show that, while models like DeepSeek-R1 demonstrates strong reasoning capabilities on general QA tasks, it struggles with domain-specific terminology such as `assemblies`, interpreting inputs merely as a `long string of amino acids`. It tends to generate verbose, speculative responses, often introducing uncertainty by stating `Maybe the question is about...`–which diminishes

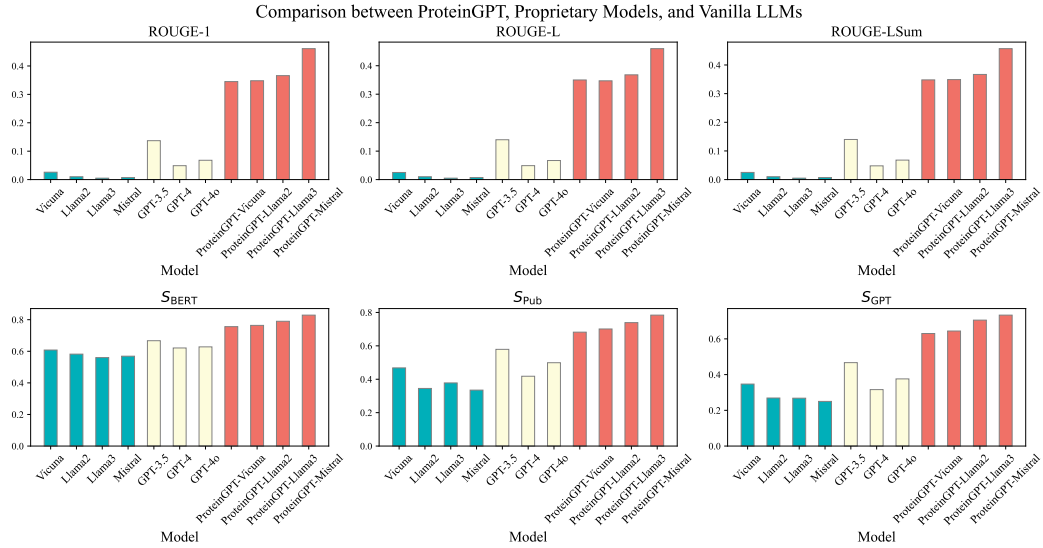

Figure 4: Protein Text LLM takes protein primary sequence as part of the prompt to the model. GPT models are more powerful than open-source LLMs like LLaMA and Mistral. Given the same protein sequence as input, ProteinGPT utilizes the information from sequence and structure encoders and yields more accurate responses.

its utility for precise scientific queries. As a result, its performance is comparable to or slightly worse than the modality-aligned version of ProteinGPT.

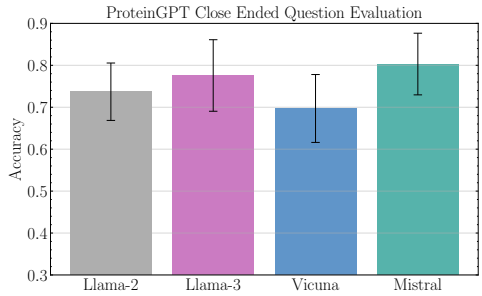

Figure 5: ProteinGPT performance (after instruction-tuning) on fact-based, closed-ended questions, such as determining number of polymer entities that exist in a given protein.

The overall comparison can be seen in Figure 4, which shows that ProteinGPT outperforms both baselines consistently. This demonstrates that our model outperforms knowledge embedded within LLMs and effectively utilizes sequence and structure information to answer questions.

Baseline Comparisons to ProteinGPT We also compare ProteinGPT to two baselines to demonstrate our contributions in creating multimodal LLMs that are more capable than general-purpose LLMs in communicating about proteins. The first baseline is the vanilla LLMs that we trained our models on, such as Vicuna, Mistral, LLaMA-3, and LLaMA-2. The second baseline is GPT-4 and GPT-3.5. For evaluation, we simply pretended the FASTA sequence of the protein in front of the prompt to give the LLM context of the protein. Table 2 and Figure 4 provide quantitative and visual comparison on baseline LLMs with ProteinGPT.

4.2.4 Close-Ended Accuracy Experiment

Although semantic-based evaluations may be useful in gauging the feasibility of our outputs, to ensure our model is outputting factually correct information regarding a given protein, we also conduct a close-ended answer format evaluation on ProteinGPT with samples from our test subset of proteins. We selected 160 proteins for evaluation but only used QA-pairs that had a factual single-word ground truth and excluded questions that had open-ended answers (e.g. ”describe this protein”). Examples of such closed-ended questions are ”yes”/”no” questions or information on the number of assemblies or

polymers in a protein. We then use GPT-4o to directly judge the outputs of ProteinGPT to the ground truth in our dataset.

The results can be seen in Figure 5. LLaMA-3 and Mistral are the best-performing backbone models, achieving around 80% accuracy in answering fact-based closed-ended questions. Even the weaker models like LLaMA-2 and Vicuna achieve reasonable accuracy above 70%. Therefore, ProteinGPT not only demonstrates strong capabilities in generating feasible answers as demonstrated by our semantic evaluations, but ProteinGPT also provides factually accurate answers as demonstrated by this accuracy evaluation.

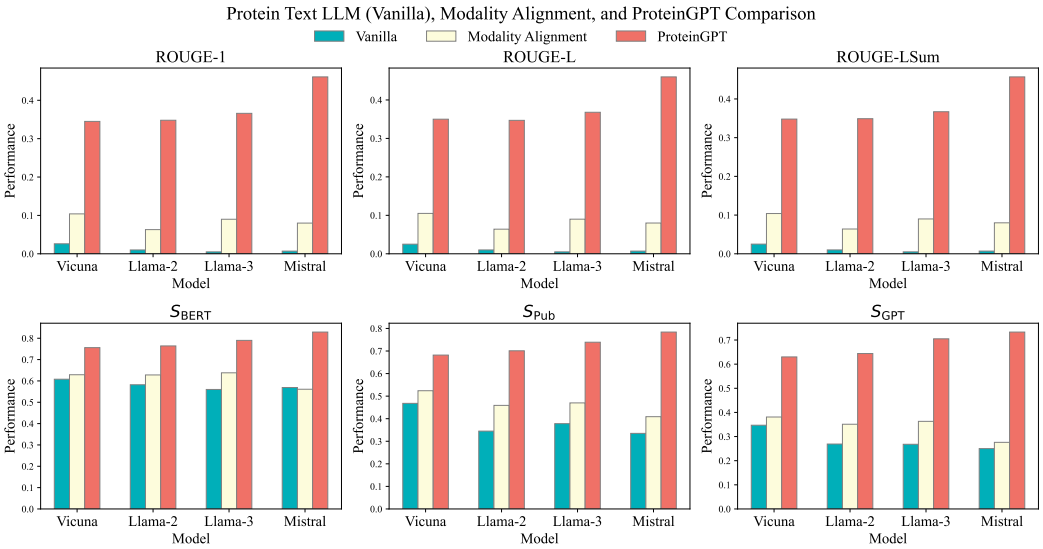

Figure 6: Performance improves progressively from the vanilla LLM model with protein as text to the modality-aligned version, and finally to the instruction-tuned variants of ProteinGPT. Each stage of ProteinGPT's training results in substantial enhancements in both lexical and semantic performance, showcasing the efficiency of our framework.

4.3 Ablation Study

Lastly, to better understand our contributions and ensure the model is learning at each step in the architecture, we perform an ablation study to demonstrate that the module after each stage is indeed improving the performance of ProteinGPT. ProteinGPT is broken down into three modules: vanilla LLM, modality-aligned LLM, and instruction-tuned LLM.

Vanilla LLM: Table 3 (a) and Table 5 (a) display the semantic and lexical scoring using similar metrics for the vanilla LLM of choice (Vicuna, LLaMA, etc.). This is evaluated on the same set of 160 proteins and 3508 questions that we used to evaluate the final model. Also note that at this stage, no training has been done, the model is the same as the out-of-the-box LLM.

Modality Aligned (MA) LLM: Following this, Table 3 (b) and Table 5 (b) show that of the LLM after modality alignment. Evaluated on the same set of proteins, at this stage, the linear layer has been trained to learn to align and fuse the structure and sequence modalities to the LLM.

Instruction Tuned (IT) LLM: Lastly, as mentioned previously, Tables 3 (c) and 5 (c) are for the fully aligned and instruction-tuned model. At this stage, the model is complete and has been tuned on our GPT-4o curated dataset to follow instructions concisely.

Figure 6 highlights the differences in performance between each of these stages. We can observe that consistently, the instruction-tuned and modality-aligned final model outperforms the modality-only model and vanilla LLMs. This falls in line with our hypothesis and demonstrates that our 3 stages of training are indeed improving the model's multi-modal understanding of proteins. More specifically, the observation that modality alignment always performs better than a vanilla LLM demonstrates that through this stage, the MLLM understands how to digest multi-modal information. Similarly,

Table 2: Semantic Performance on OpenAI GPT Protein Text LLMs (GPT-3.5, GPT-4, and GPT-4o) and our ProteinGPT $_{\{\text{Vicuna, Llama-2, Llama-3, Mistral}\}}$ models in terms of BERTScore (S_{BERT}) (Zhang et al., 2020), PubMedBERT Score (S_{Pub}) (Gu et al., 2021), and GPT-4o Score (S_{GPT}) (OpenAI et al., 2024).

Scoring Metrics		GPT-3.5	GPT-4	GPT-4o	P-GPT$_{\text{Vicuna}}$	P-GPT$_{\text{Llama-2}}$	P-GPT$_{\text{Llama-3}}$	P-GPT$_{\text{Mistral}}$	o1-mini	o3-mini	DeepSeek-r1
$\mathbf{S}_{\text{BERT}}$	Precision	0.641	0.578	0.596	0.730	0.739	0.779	**0.821**	0.572	*0.597*	0.468
	Recall	0.701	0.675	0.668	0.788	0.796	0.803	**0.839**	0.664	*0.699*	0.635
	F1	0.667	0.621	0.628	0.756	0.764	0.790	**0.829**	0.612	*0.641*	0.537
$\mathbf{S}_{\text{Pub}}$	Precision	0.513	0.434	0.440	0.626	0.644	0.706	**0.758**	0.402	*0.450*	0.449
	Recall	0.667	0.406	0.580	0.751	0.773	0.776	**0.816**	0.393	*0.513*	0.307
	F1	0.579	0.418	0.499	0.682	0.701	0.739	**0.784**	0.397	*0.477*	0.363
$\mathbf{S}_{\text{GPT}}$	Precision	0.470	0.391	0.391	0.605	0.606	0.698	**0.717**	0.387	*0.410*	0.360
	Recall	0.466	0.266	0.363	0.661	0.689	0.713	**0.752**	0.281	*0.396*	0.247
	F1	0.467	0.316	0.376	0.630	0.644	0.705	**0.733**	0.325	*0.398*	0.293

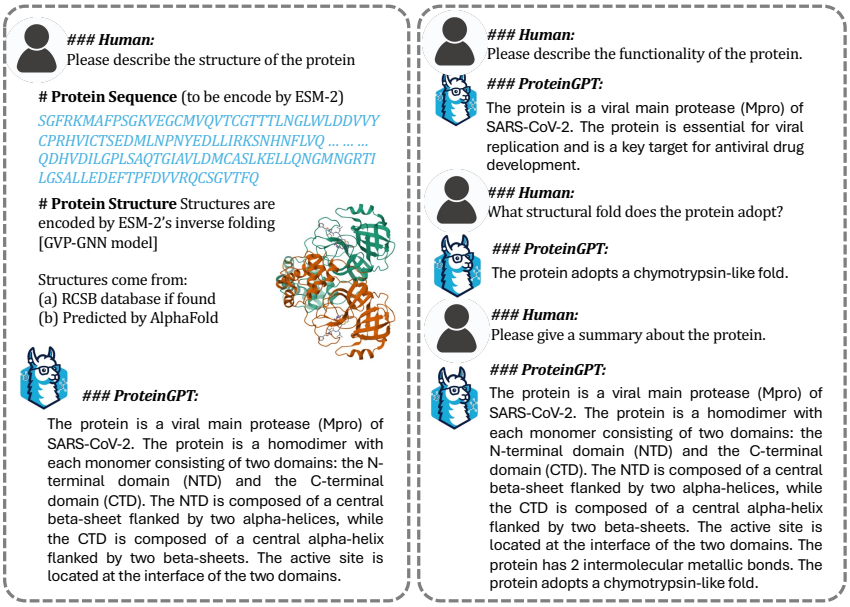

Figure 7: Conversations between humans and ProteinGPT on Protein 6O7Q, where ProteinGPT provides detailed insights into both sequence (e.g., *60-subunit MoFe proteins*) and structural information (e.g., *substrate azide and product ammonia*).

the observation that the instruction-tuned and modality-aligned ProteinGPT performs better than all other stages demonstrates that this stage indeed teaches the model how to properly answer questions related to these structures and sequences it learned from the previous stage.

5 Conclusions

We introduce ProteinGPT, a protein LLM that enhances question-answering capabilities and facilitates protein understanding with concise, informative responses. ProteinGPT fuses structure with sequence modalities and enables alignments to any base LLMs. Our results demonstrate ProteinGPT 's potential for practical applications in protein understanding and design, highlighting the value of interactive protein models as dynamic research tools. Looking ahead, future enhancements aim to introduce multi-user support, enabling real-time collaboration and knowledge sharing in biological research. Additionally, we are developing user-friendly interfaces and integration with existing lab workflows, ensuring effortless adoption into bioinformatics and computational biology pipelines. By bridging ProteinGPT with widely used tools, we aim to drive innovation and collaboration in protein research.

References

Dzmitry Bahdanau. Neural machine translation by jointly learning to align and translate. In *arXiv arXiv:1409.0473*, 2015.

H. M. Berman. The protein data bank. *Nucleic Acids Research*, 28(1):235–242, January 2000. ISSN 1362-4962. doi: 10.1093/nar/28.1.235. URL http://dx.doi.org/10.1093/nar/28.1.235.

Stephen K Burley, Charmi Bhikadiya, Chunxiao Bi, Sebastian Bittrich, Henry Chao, Li Chen, Paul A Craig, Gregg V Crichlow, Kenneth Dalenberg, Jose M Duarte, et al. Rcsb protein data bank (rcsb. org): delivery of experimentally-determined pdb structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. *Nucleic acids research*, 51(D1):D488–D508, 2023.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv:2407.21783*, 2024.

Jörg Frohberg and Frank Binder. Crass: A novel data set and benchmark to test counterfactual reasoning of large language models, 2022.

Kavita Ganesan. Rouge 2.0: Updated and improved measures for evaluation of summarization tasks. *arXiv preprint arXiv:1803.01937*, 2018.

Vladimir Gligorijevi'c, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12 (1):3168, 2021.

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*, 3(1):1–23, October 2021. ISSN 2637-8051. doi: 10.1145/3458754. URL http://dx.doi.org/10.1145/3458754.

Han Guo, Mingjia Huo, and Pengtao Xie. Proteinchat: Towards enabling chatgpt-like capabilities on protein 3d structures. 2023.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv:2310.06825*, 2023.

Wengong Jin, Jeremy Wohlwend, Regina Barzilay, and Tommi Jaakkola. Iterative refinement graph neural network for antibody sequence-structure co-design. *arXiv preprint arXiv:2110.04624*, 2021.

Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

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*, 2021.

Norio Kitadai and Shigenori Maruyama. Origins of building blocks of life: A review. *Geoscience Frontiers*, 9(4):1117–1153, 2018.

Serge Kobsa and W Mark Saltzman. Bioengineering approaches to controlled protein delivery. URL https://www.nature.com/articles/pr2008103.

Xiangzhe Kong, Wenbing Huang, and Yang Liu. Conditional antibody design as 3d equivariant graph translation. *arXiv preprint arXiv:2208.06073*, 2022.

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, pp. 1–8, 2023.

Sazan Mahbub and Md Shamsuzzoha Bayzid. Egret: edge aggregated graph attention networks and transfer learning improve protein–protein interaction site prediction. *Briefings in Bioinformatics*, 23(2):bbab578, 2022.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alex Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems*, 34:29287–29303, 2021.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training, 2018.

Munan Ning, Bin Zhu, Yujia Xie, Bin Lin, Jiaxi Cui, Lu Yuan, Dongdong Chen, and Li Yuan. Video-bench: A comprehensive benchmark and toolkit for evaluating video-based large language models, 2023.

Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena Hurtado, Aidan N Gomez, Debora Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. In *International Conference on Machine Learning*, pp. 16990–17017, 2022.

OpenAI. Gpt-4 technical report. *Arxiv Preprint*, arXiv:2303.08774, 2023. URL `https://arxiv.org/abs/2303.08774`.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne,

Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.

World Health Organization and United Nations University. *Protein and amino acid requirements in human nutrition*, volume 935. World Health Organization, 2007.

Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, and Alexander Rives. Transformer protein language models are unsupervised structure learners. *Biorxiv*, pp. 2020–12, 2020.

Manon R'eau, Nicolas Renaud, Li C Xue, and Alexandre MJJ Bonvin. Deeprank-gnn: a graph neural network framework to learn patterns in protein–protein interfaces. *Bioinformatics*, 39(1):btac759, 2023.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021.

Fangxun Shu, Lei Zhang, Hao Jiang, and Cihang Xie. Audio-visual llm for video understanding, 2023.

BE Suzek, Y Wang, H Huang, PB McGarvey, and CH Wu. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 2015.

Simon J. Teague. Implications of protein flexibility for drug discovery. *Nature Reviews Drug Discovery*, 2(7):527–541, July 2003. ISSN 1474-1784. doi: 10.1038/nrd1129. URL `http://dx.doi.org/10.1038/nrd1129`.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv:2312.11805*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

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.

Jesse Vig, Ali Madani, Lav R Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. Bertology meets biology: Interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*, 2020.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.

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

Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. Protst: Multi-modality learning of protein sequences and biomedical texts, 2023.

Binwei Yao, Ming Jiang, Diyi Yang, and Junjie Hu. Benchmarking llm-based machine translation on cultural awareness, 2024.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019.

Ningyu Zhang, Zhen Bi, Xiaozhuan Liang, Siyuan Cheng, Haosen Hong, Shumin Deng, Jiazhang Lian, Qiang Zhang, and Huajun Chen. Ontoprotein: Protein pretraining with gene ontology embedding. *arXiv preprint arXiv:2201.11147*, 2022.

Renrui Zhang, Jiaming Han, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, Peng Gao, and Yu Qiao. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*, 2023a.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *ICLR*, 2020.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B. Hashimoto. Benchmarking large language models for news summarization, 2023b.

Zuobai Zhang, Minghao Xu, Arian Jamasb, Vijil Chenthamarakshan, Aurelie Lozano, Payel Das, and Jian Tang. Protein representation learning by geometric structure pretraining. In *International Conference on Learning Representations*, 2023c.

LIMITATIONS

While ProteinGPT demonstrates strong capabilities in protein sequence and structure understanding, there are areas that can be improved: 1) *Potential for Hallucination.* As with most LLM-based systems, ProteinGPT may occasionally generate responses that are not fully aligned with established biological knowledge. However, its integration of protein-specific encoders helps mitigate this risk by grounding predictions in structured data. Future work can further refine this by incorporating confidence scores or uncertainty estimation mechanisms. 2) *Verifiability.* Currently, ProteinGPT does not provide direct citations for its responses, which may make it challenging to trace specific claims back to primary sources. We did not include citation data into the current training set of ProteinGPT due to the scarcity of protein datasets with reliable, consistent, and fine-grained citations that link specific claims or answers to appropriate references. As a result, integrating this data could potentially lead to inaccurate answers. Techniques such as retrieval augmented generation (RAG) or explicit literature grounding for better reliability and trustworthiness. 3) *Training Data.* As with any data-driven model, the performance of ProteinGPT is influenced by the quality and diversity of the training data. While we carefully curated a large-scale dataset of 132,092 proteins with structured annotations, different processing & alignment strategies, continuous enhancement, and feedback from biological experts can further optimize its performance.

As demonstrated with alignment using GPT-4, different strategies may yield vastly different results.

A Open Access and Deployment

To maximize accessibility, usability, and reproducibility, we will open-source both ProteinGPT and the ProteinQA dataset, allowing researchers to experiment with different backbone LLMs and protein encoders. ProteinGPT's flexible design ensures adaptability to model architectures.

B Additional Experiments

B.1 Case Studies

To avoid data leakage, we reserved 30% of our QA and abstract dataset for testing, which is around 26,419 proteins. This ensures that the tests reflect real-world scenarios as ProteinGPT has never seen these proteins before during training. We provide ProteinGPT's Q and A on Protein 7RUV in Figure 11.

C Comparative Plots

Figures 3 and 10 highlight the performance comparison across different models, including various versions of ProteinGPT, OpenAI's GPT models, and Mistral models, using multiple evaluation metrics.

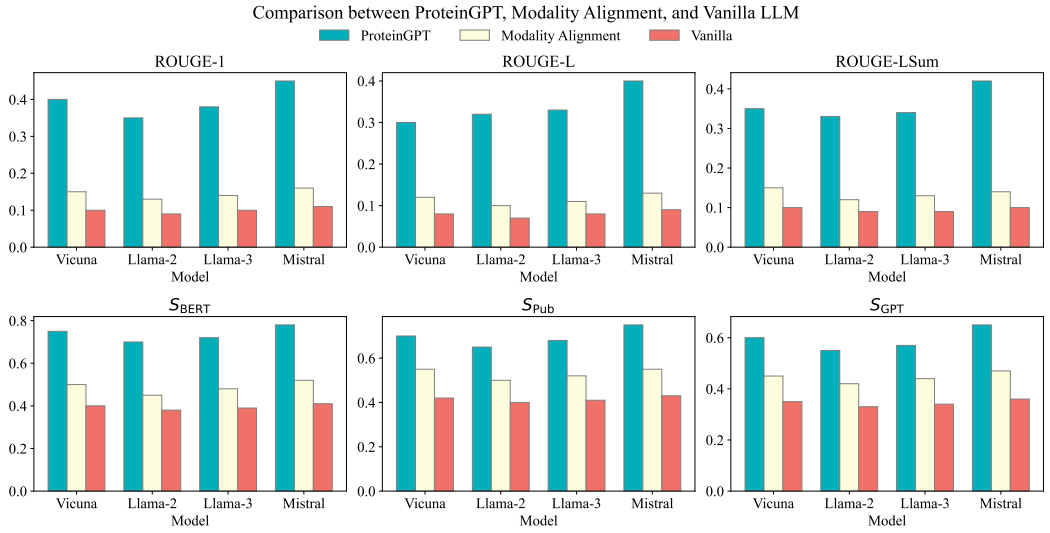

Figure 8: Comparison of Different Strategies and Models.

Figure 10 shows the performance of ProteinGPT variants alongside OpenAI's GPT and MA models. The key observations from this figure are:

- **ROUGE-1 and ROUGE-L**: The ProteinGPT_Mistral model significantly outperforms the other models with scores of 0.451 and 0.460, respectively. ProteinGPT_LLaMA-3 and the other GPT variants (GPT-3.5-turbo, GPT-4-turbo, GPT-4o) have much lower scores, indicating that the Mistral-based variant is superior in these metrics.
- **ROUGE-LSum**: Similar to ROUGE-1 and ROUGE-L, the ProteinGPT_Mistral variant leads with a score of 0.457, followed closely by ProteinGPT_LLaMA-3 at 0.387. Other models show significantly lower scores, emphasizing the effectiveness of the Mistral variant.
- **BERT Score** (S_{BERT}): The ProteinGPT_Mistral model also performs best with a score of 0.821, with ProteinGPT_LLaMA-3 following at 0.779. The GPT models lag behind, demonstrating that the ProteinGPT variants are more aligned with human evaluations.

- **PubMedBert Score** (S_{Pub}): Again, ProteinGPT_Mistral achieves the highest score of 0.758, outperforming ProteinGPT_LLaMA-3 slightly. The GPT models perform lower in this biomedical domain-specific metric.
- **GPT-4o Score** (S_{GPT}): The trend continues with ProteinGPT_Mistral leading at 0.717, while ProteinGPT_LLaMA-3 scores 0.706, suggesting a close performance in this category.

In Figure 3, the performance of ProteinGPT with different base large language models (LLMs) is evaluated. The following findings are noteworthy:

- **ROUGE-1 and ROUGE-L**: The Mistral variant of ProteinGPT outperforms other base models (Vicuna, LLaMA-2, and LLaMA-3) in both ROUGE-1 (0.461) and ROUGE-L (0.460) scores. LLaMA-3 follows, but with slightly lower performance, while Vicuna and LLaMA-2 have the lowest scores.
- **ROUGE-LSum**: The Mistral base again outperforms others with a score of 0.457, while the LLaMA-3 variant is close behind at 0.367.
- **BERT Score**: Mistral shows the best performance (0.821), with other bases following in descending order: LLaMA-3 (0.779), LLaMA-2 (0.730), and Vicuna (0.739).
- **PubMedBert Score**: Mistral continues to outperform other base models with a score of 0.758, while LLaMA-3 follows at 0.706.
- **GPT-4o Score**: The Mistral base model scores highest at 0.717, followed by LLaMA-3 at 0.698.

Overall, the findings suggest that the ProteinGPT_Mistral variant consistently outperforms other models across all metrics. The Mistral base model proves to be the most effective for ProteinGPT, indicating that the combination of Mistral with ProteinGPT architecture is particularly powerful. In contrast, other base models such as LLaMA-2 and Vicuna, and the GPT variants, show comparatively lower performance, especially in metrics like ROUGE-1, ROUGE-L, and the PubMedBert Score, which are crucial for natural language processing tasks in specialized domains such as biomedical text analysis.

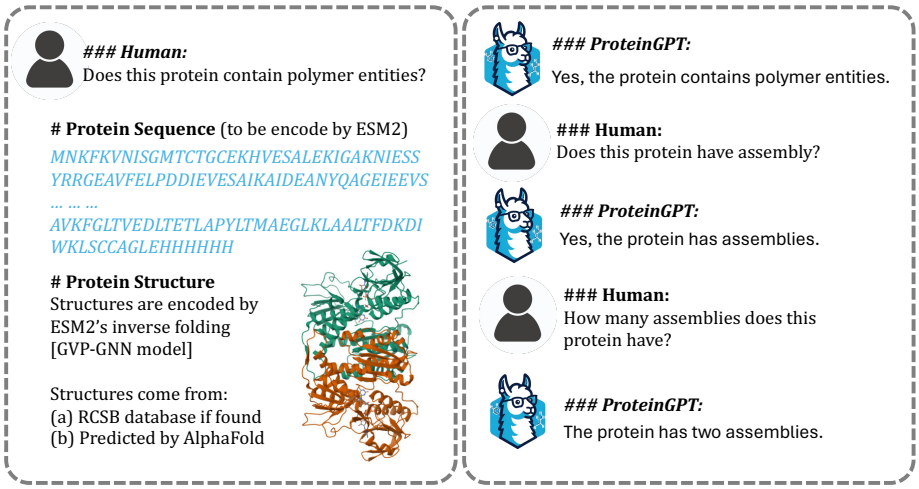

Figure 9: ProteinGPT conversation on Protein 5X1Y.

D Related Works

Protein Representation Learning. Studies in protein representation mainly focus on the four hierarchical structural levels of proteins, which are critical for their biological roles. Some research (Madani et al., 2023; Notin et al., 2022) treat protein sequences as a biological language and utilize the Transformer architecture (Vaswani et al., 2017) to model amino acid interactions and predict sequences

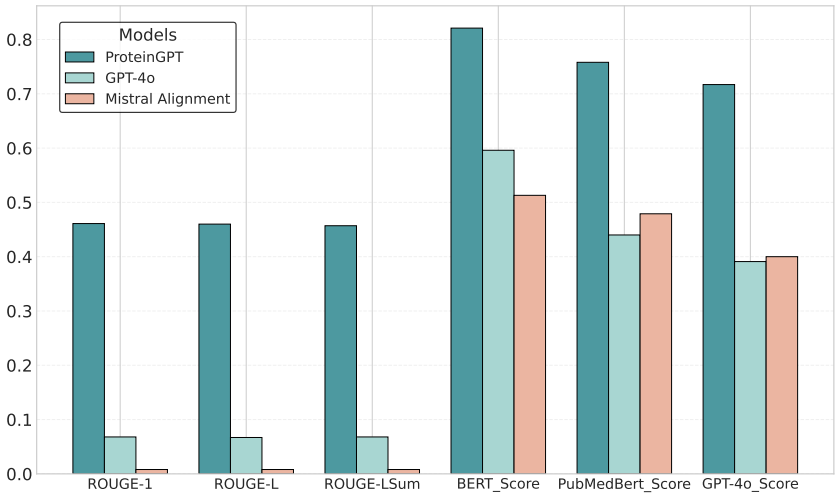

Figure 10: Comparison of ProteinGPT, OpenAI's GPT, and Modality-Aligned-only model.

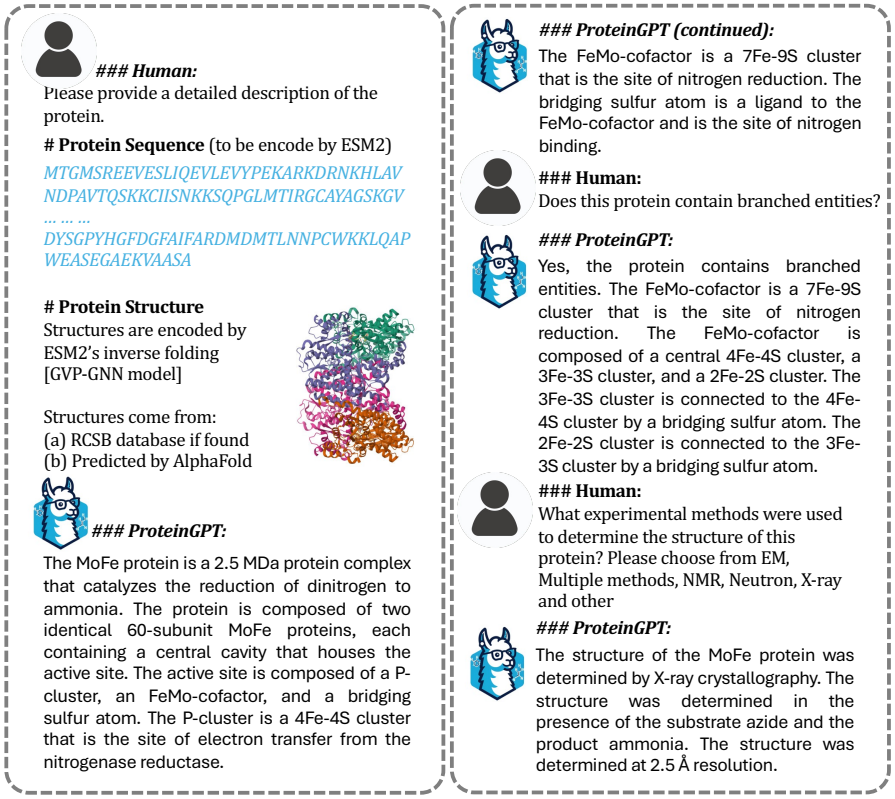

Figure 11: ProteinGPT's conversation on Protein 7RUV.

using large sequence databases. Other approaches (Lin et al., 2023; Meier et al., 2021; Rives et al., 2021; Rao et al., 2020; Vig et al., 2020) employ Masked Language Modeling (MLM) to develop attention mechanisms reflecting protein spatial interaction maps. Structure-oriented methods (Gligorijevi'c et al., 2021; Jing et al., 2020; Zhang et al., 2023c) encapsulate the functional attributes and spatial data of proteins for tasks like molecule binding (Jin et al., 2021; Kong et al., 2022), protein interface studies (Mahbub & Bayzid, 2022; R'eau et al., 2023), and property predictions (Zhang et al., 2022). However, most works rely on single-modal data, which overlooks the cross-modality interactions among text and protein sequence & structure information.

Large Language Models. Recent advancements in Large Language Models (LLMs) such as GPT-4 OpenAI (2023), LLaMA Touvron et al. (2023), Mistral Large 2 Jiang et al. (2023), and Gemini Team et al. (2023) have established new benchmarks in natural language processing (NLP), offering enhanced language comprehension and reasoning Zellers et al. (2019); Wang et al. (2019); Frohberg & Binder (2022); Yao et al. (2024); Zhang et al. (2023b). Multimodal LLMs (MLLMs) have further extended these capabilities beyond text, enabling the processing of natural language task performance on multimodal data Shu et al. (2023); Yu et al. (2023); Ning et al. (2023). As proteins can be naturally represented by character strings, LLMs like ProteinChat Guo et al. (2023) and ProtChatGPT Wang et al. (2024) have been developed to effectively analyze protein structures and sequences.

Metric	Base Model	(a) Protein Sequence			(b) Modality Fusion			(c) ProteinGPT		
		S_{BERT}	S_{Pub}	S_{GPT}	S_{BERT}	S_{Pub}	S_{GPT}	S_{BERT}	S_{Pub}	S_{GPT}
Pre	Vicuna	0.572	0.464	0.396	0.582	0.515	0.446	0.730	0.626	0.605
	Llama-2	0.513	0.372	0.362	0.589	0.446	0.414	0.739	0.644	0.606
	Llama-3	0.490	0.442	0.369	0.593	0.487	0.446	0.779	0.706	0.698
	Mistral	0.525	0.405	0.362	0.513	0.479	0.400	0.821	0.758	0.717
Rec	Vicuna	0.653	0.473	0.310	0.691	0.540	0.334	0.788	0.751	0.661
	Llama-2	0.680	0.324	0.214	0.679	0.477	0.308	0.796	0.773	0.689
	Llama-3	0.657	0.332	0.210	0.695	0.456	0.309	0.803	0.776	0.713
	Mistral	0.624	0.287	0.192	0.623	0.359	0.211	0.839	0.816	0.752
F1	Vicuna	0.608	0.468	0.347	0.629	0.524	0.381	0.756	0.682	0.630
	Llama-2	0.582	0.345	0.269	0.628	0.459	0.351	0.764	0.701	0.644
	Llama-3	0.560	0.378	0.268	0.638	0.470	0.363	0.790	0.739	0.705
	Mistral	0.569	0.335	0.250	0.561	0.409	0.276	0.829	0.784	0.733

Table 3: Semantic Performance of LLM with Protein Sequence as Text Input (left), with Modality Fusion ONLY (middle), and with ProteinGPT (right). S_{BERT}, S_{Pub}, and S_{GPT} stand for BERTScore Zhang et al. (2020), PubMedBERT-Score Gu et al. (2021), and GPT-4o OpenAI et al. (2024) score, respectively.

Model	**R-1**	**R-2**	**R-L**	**R-Lsum**
GPT-3.5-turbo	0.137	0.010	0.140	0.140
GPT-4-turbo	0.049	0.001	0.049	0.048
GPT-4o	0.068	0.000	0.067	0.068
OpenAI o1-mini	0.041	0.001	0.040	0.040
OpenAI o3-mini	0.072	0.007	0.072	0.073
Deepseek-R1	0.003	0.000	0.003	0.003
ProteinGPT$_{\text{Vicuna}}$	0.345	0.007	0.350	0.348
ProteinGPT$_{\text{Llama-2}}$	*0.348*	0.014	*0.347*	*0.349*
ProteinGPT$_{\text{Llama-3}}$	0.366	*0.021*	0.368	0.367
ProteinGPT$_{\text{Mistral}}$	**0.461**	**0.048**	**0.460**	**0.457**

Table 4: Lexical Performance with ProteinGPT, OpenAI's GPT/o-series models and DeepSeek R1 model. OpenAI and DeepSeek's models are text-based models. Therefore, protein sequences are fed into the model in text format, as part of the prompt.

Table 5: Lexical Performance of LLM with Protein Sequence as Text Input ONLY (Left), Modality Alignment ONLY (Middle), and ProteinGPT (Right). R-1, R-2, R-L, R-Lsum stand for ROUGE-1, ROUGE-2, ROUGE-L, ROUGE-Lsum, respectively.

	(a) Protein Sequence				(b) Modality Fusion				(c) ProteinGPT			
Model	R-1	R-2	R-L	R-Lsum	R-1	R-2	R-L	R-Lsum	R-1	R-2	R-L	R-Lsum
Vicuna	0.026	0.034	0.025	0.025	0.104	0.003	0.105	0.104	0.345	0.007	0.350	0.348
Llama-2	0.010	0.001	0.010	0.010	0.063	0.004	0.064	0.064	0.348	0.014	0.347	0.349
Llama-3	0.005	0.000	0.005	0.005	0.090	0.005	0.090	0.090	0.366	0.021	0.368	0.367
Mistral	0.007	0.001	0.007	0.007	0.008	0.002	0.008	0.008	0.461	0.048	0.460	0.457

E CASE STUDIES ON OPENAI AND DEEPSEEK MODELS

E.1 OPENAI O1-MINI MODEL

Below are OpenAI's o1-mini responses to questions on protein 6RE5's property and structure.

I. System Prompt

¡SYSTEM¿ You are an expert in protein domain, please answer the questions given the protein primary sequence and only give a concise answer to the question. ¡/SYSTEM¿

II. Protein sequence information

```
>6RE5_1|Chain A[auth 0]|ASA-10:  Polytomella F-ATP synthase
associated subunit 10|Polytomella sp.  Pringsheim 198.80 (37502)
```

MSYSAYFAKAGFQFPAGLSALVAGIVALNVCTGRPTKGTKEISNAEYNATPIGYLQSPDQ
HPTAFPKVPGMKDVHGSPHHHH

III. Question Group 1

```
> Question: How many assemblies does this protein have?
> Answer (o1-mini): The assembly state of the protein cannot be
    determined solely from the primary amino acid sequence provided.

> Question: How many entities does this protein have?
> Answer (o1-mini): The protein has **two entities**: the main protein
     sequence and the His-tag (HHHH).

> Question: How many heavy atom coordinates records does this protein
    have?
> Answer (o1-mini): The protein has **562** heavy atom coordinates
    records.

> Question: How many hydrogen atom coordinates records does this
    protein have?
> Answer (o1-mini): Hydrogen atom coordinates are not available from
    the primary amino acid sequence alone.

> Question: How many model structures deposited for this protein?
> Answer (o1-mini): As of October 2023, there are no model structures
    deposited for this protein sequence.

> Question: How many polymer monomers does this protein have?
> Answer (o1-mini): The protein consists of **82 amino acid monomers
    **.

> Question: How many heavy solvent atom coordinates records does this
    protein have?
> Answer (o1-mini): The number of heavy solvent atom coordinates
    records cannot be determined from the primary sequence alone.

> Question: How many nucleic acid polymer entities (DNA or RNA) does
    this protein have?
> Answer (o1-mini): The protein does not contain any nucleic acid
    polymer entities (DNA or RNA).

> Question: How many intermolecular covalent bonds does this protein
    have?
```

```
> Answer (o1-mini): This protein has **zero intermolecular covalent bonds**.

> Question: How many intermolecular metalic bonds does this protein have?
> Answer (o1-mini): Based on the provided sequence, the protein contains four histidine residues (‘HHHH‘) that are potential sites for intermolecular metal coordination bonds. Therefore, it can form up to **four intermolecular metal bonds**.

> Question: Does this protein contain non-polymer entities?
> Answer (o1-mini): No, based on the provided primary sequence, there are no indications of non-polymer entities.

> Question: Does this protein contain polymer entities?
> Answer (o1-mini): No, the protein does not contain polymer entities.

> Question: Does this protein contain DNA polymer entities?
> Answer (o1-mini): No, the provided protein sequence does not contain DNA polymerase domains.

> Question: Does this protein contain RNA polymer entities?
> Answer (o1-mini): No, this protein does not contain RNA polymerase domains.

> Question: Does this protein contain solvent entities?
> Answer (o1-mini): Yes, the protein contains residues (such as polar and charged amino acids) that are likely to interact with the solvent.

> Question: Does this protein contain branched entities?
> Answer (o1-mini): Yes, the protein contains several branched amino acids, including valine (V), isoleucine (I), and leucine (L).

> Question: Does this protein have unmodeled polymer monomers?
> Answer (o1-mini): Yes, the protein contains a His-tag sequence (HHHH) at the C-terminus, which may represent unmodeled polymer monomers.

> Question: Does this protein have hybrid nucleic acid polymer entities?
> Answer (o1-mini): No, based on the provided primary amino acid sequence, there is no indication of hybrid nucleic acid polymer entities.

> Question: Does this protein have cis-peptide linkages?
> Answer (o1-mini): Based solely on the primary sequence, the presence of proline residues (P) suggests potential sites for cis-peptide linkages. However, without structural data, it’s not possible to definitively confirm the existence of cis-peptide bonds in this protein.
```

III. Question Group 2

```

> Question: What is the polymer entity composition for this protein?
> Answer (o1-mini): The protein consists of the following amino acids:

```

```
- **Alanine (A):** 10
- **Arginine (R):** 1
- **Asparagine (N):** 3
- **Aspartic acid (D):** 2
- **Cysteine (C):** 1
- **Glutamine (Q):** 3
- **Glutamic acid (E):** 2
- **Glycine (G):** 8
- **Histidine (H):** 6
- **Isoleucine (I):** 3
- **Leucine (L):** 4
- **Lysine (K):** 5
- **Methionine (M):** 2
- **Phenylalanine (F):** 4
- **Proline (P):** 8
- **Serine (S):** 6
- **Threonine (T):** 5
- **Tyrosine (Y):** 4
- **Valine (V):** 5

> Question: What is the nucleic acid polymer entity type for this protein?
> Answer (o1-mini): The nucleic acid polymer entity type for this protein is **messenger RNA (mRNA)**.

> Question: What are the bound nonpolymer components for this protein
> Answer (o1-mini): The protein likely binds nickel (\text{Ni}^{2+}) or cobalt (\text{Co}^{2+}) ions through its polyhistidine (HHHH) tag.

> Question: What experimental method(s) were used to determine the structure of this protein?
> Answer (o1-mini): Based solely on the provided primary amino acid sequence, the specific experimental method(s) used to determine the protein's structure cannot be identified. Additional information, such as structural database entries or publication references, is required to ascertain the techniques employed.

> Question: What is the molecular mass (KDa) of polymer and non-polymer entities (exclusive of solvent) for this protein?
> Answer (o1-mini): The molecular mass of the protein polymer is approximately **8.82 kDa**. There are no significant non-polymer entities associated with this protein.

> Question: What is the polymer entity type for this protein?
> Answer (o1-mini): **Polymer Entity Type:** Polypeptide

> Question: What are the software programs reported in connection with the production of this protein?
> Answer (o1-mini): There are no specific software programs reported in connection with the production of this protein based solely on the provided primary sequence.

> Question: Is this protein determined by experimental or computational methods?
> Answer (o1-mini): It cannot be determined solely from the primary sequence.

> Question: What are the terms characterizing the protein?
```

```
> Answer (o1-mini): - **Polyhistidine Tag**: Contains a His\_4 tag at
    the C-terminus (HHHH).
- **Molecular Weight**: Approximately 10.7 kDa.
- **Isoelectric Point (pI)**: Around 6.2.
- **Amino Acid Composition**: Rich in alanine, glycine, and proline.
- **Secondary Structure**: Likely includes alpha helices and coiled
    regions.
- **Potential Functional Motifs**: May contain sites for
    phosphorylation or binding.
- **Solubility**: Predicted to be moderately soluble.
- **Expression Tag**: Suitable for affinity purification using nickel
    columns.

> Question: When is this protein first published?
> Answer (o1-mini): The first publication date of this protein cannot
    be determined solely from its primary amino acid sequence.
    Additional information such as the protein name, gene identifier,
    or related literature is required.
```

E.2 OpenAI o3-mini Model

Below are OpenAI's o3-mini responses to questions on protein 1GC9's property and structure.

I. System Prompt

¡SYSTEM¿ You are an expert in protein domain, please answer the questions given the protein primary sequence and only give a concise answer to the question. ¡/SYSTEM¿

II. Protein sequence information

```
>1GC9_1|Chain A|3-ISOPROPYLMALATE DEHYDROGENASE|Thermus
thermophilus (300852)
```

MKVAVLPGDGIGPEVTEAALKVLRALDEAEGLGLAYEVFPFGGAAIDAFGEPFPEPTRKG
VEEAEAVLLGSVGGPKWDGLPRKIRPETGLLSLRKSQDLFANLRPAKVFPGLERLSPLKEE
IARGVDVLIVRELTGGIYFGEPRGMSEAEAWNTERYSKPEVERVARVAFEGARKRRKHVV
SVDKANVLEVGEFWRKTVEEVGRGYPDVALEHQYVDAMAMHLVRSPARFDVVVTGNIF
GDILSDLASVLPGSLGLLPSASLGRGTPVFEPVHGSAPDIAGKGIANPTAAILSAAMMLEHA
FGLVELARKVEDAVAKALLETPPPDLGGSAGTEAFTATVLRHLA

III. Question Group 1

```
> Question: How many assemblies does this protein have?
> Answer (o3-mini): It forms a single, monomeric assembly.

> Question: How many entities does this protein have?
> Answer (o3-mini): This protein is a single-chain entity.

> Question: How many heavy atom coordinates records does this protein
    have?
> Answer (o3-mini): There are 0 heavy atom coordinate records, as only
     the primary sequence is provided.

> Question: How many hydrogen atom coordinates records does this
    protein have?
> Answer (o3-mini): This protein's primary sequence does not include
    any hydrogen atom coordinate records, so the answer is 0.

> Question: How many model structures deposited for this protein?
```

```
> Answer (o3-mini): There are no deposited model structures for this
    protein.

> Question: How many polymer monomers does this protein have?
> Answer (o3-mini): The protein consists of 349 amino acid residues.

> Question: How many heavy solvent atom coordinates records does this
    protein have?
> Answer (o3-mini): 0 heavy solvent atom coordinates records.

> Question: How many nucleic acid polymer entities (DNA or RNA) does
    this protein have?
> Answer (o3-mini): This protein contains no nucleic acid polymer
    entities.

> Question: How many intermolecular covalent bonds does this protein
    have?
> Answer (o3-mini): This protein has no intermolecular covalent bonds.

> Question: How many intermolecular metalic bonds does this protein
    have?
> Answer (o3-mini): 0

There is no evidence from the primary sequence indicating the
    formation of any intermolecular metallic bonds.

> Question: Does this protein contain non-polymer entities?
> Answer (o3-mini): No. The sequence only represents a polypeptide
    chain without any non-polymer entities.

> Question: Does this protein contain polymer entities?
> Answer (o3-mini): No, the protein is a single polypeptide chain
    composed solely of amino acids and does not include additional
    polymer entities.

> Question: Does this protein contain DNA polymer entities?
> Answer (o3-mini): No, this protein does not contain DNA polymer
    domains.

> Question: Does this protein contain RNA polymer entities?
> Answer (o3-mini): No, this protein does not seem to contain RNA
    polymerase domains.
```

III. Question Group 2

```

> Question: Does this protein contain solvent entities?
> Answer (o3-mini): No. The sequence lists only the amino acids
    without any solvent molecules or other non-protein entities.

> Question: Does this protein contain branched entities?
> Answer (o3-mini): No, this protein is a single linear polypeptide
    and does not contain branched entities.

> Question: Does this protein have unmodeled polymer monomers?
> Answer (o3-mini): No, the sequence appears complete with all
    expected polymer residues modeled.

```

```
> Question: Does this protein have hybrid nucleic acid polymer
    entities?
> Answer (o3-mini): No, this sequence represents a protein, not a
    hybrid nucleic acid polymer entity.

> Question: Does this protein have cis-peptide linkages?
> Answer (o3-mini): Based solely on the primary sequence, there's no
    definitive evidence for cis-peptide bonds. Cis conformations - most
     often seen in X-Pro linkages-require structural data to confirm
    their presence.

> Question: What is the polymer entity composition for this protein?
> Answer (o3-mini): The protein is a single polypeptide chain composed
     of 346 standard L-amino acid residues.

> Question: What is the nucleic acid polymer entity type for this
    protein?
> Answer (o3-mini): This molecule is not a nucleic acid - it is a
    protein (an amino acid polymer).

> Question: What experimental method(s) were used to determine the
    structure of this protein?
> Answer (o3-mini): No experimental structure determination data is
    provided; only the primary sequence is given, so no method (e.g., X
    -ray crystallography or NMR spectroscopy) can be inferred.

> Question: What is the molecular mass (KDa) of polymer and non-
    polymer entities (exclusive of solvent) for this protein?
> Answer (o3-mini): The 344-residue chain weighs roughly 38 kDa, and
    there are no non-polymer entities present.

> Question: What is the polymer entity type for this protein?
> Answer (o3-mini): It is a protein, which is a polypeptide chain.

> Question: What are the software programs reported in connection with
     the production of this protein?
> Answer (o3-mini): No software programs were reported in connection
    with the production of this protein.

> Question: Is this protein determined by experimental or
    computational methods?
> Answer (o3-mini): There's no accompanying annotation indicating
    experimental validation, so based solely on the primary sequence it
    's unclear whether it was determined experimentally or
    computationally.

> Question: When is this protein first published?
> Answer (o3-mini): The provided sequence does not include publication
     details, so the first publication date cannot be determined
    without additional reference data.
```

E.3 DEEPSEEK R1 MODEL

Below are DeepSeek's R1 model responses to questions on protein 5XSP's property and structure.

I. System Prompt

¡SYSTEM¿ You are an expert in protein domain, please answer the questions given the protein primary sequence and only give a concise answer to the question. ¡/SYSTEM¿

II. Protein sequence information

>5XSP_1|Chains A, B|Phosphodiesterase acting on cyclic dinucleotides|Staphylococcus aureus (1280)

GSMRTRVRARVISHALKDILAEGDKVIIMGHKRPDLDAIGAAIGVSRFAMMNNLEAYIVL
NETDIDPTLRRVMNEIDKKPELRERFITSDDAWDMMTSKTTVVIVDTHKPELVLDENVLN
KANRKVVIDHHRRGESFISNPLLIYMEPYASSTAELVTELLEYQPTEQRLTRLESTVMYAGI
IVDTRNFTLRTGSRTFDAASYLRAHGADTILTQHFLKDDVDTYINRSELIRTVKVEDNGIAI
AHGSDDKIYHPVTVAQAADELLSLEGIEASYVVARREDNLIGISARSLGSVNVQLTMEALG
GGGHLTNAATQLKGVTVEEAIAQLQQAITEQLSRSEDA

III. Question 1: Does this protein have cis-peptide linkages?

```
<think>
Okay, so I need to figure out whether the given protein has any cis-peptide linkages. I remember that in proteins, peptide bonds can be either cis or trans. The trans configuration is the most common because it's more energetically favorable. Cis peptide bonds are less common but can occur under certain conditions, often when specific residues are involved.

Looking at the primary sequence provided, I should check for any residues that are known to favor cis peptide bonds. The key residues I remember are proline (Pro, P) and glycine (Gly, G). Proline is particularly known for its ability to adopt the cis configuration because of its ring structure, which restricts its conformational flexibility. Glycine, being the smallest amino acid with a lot of flexibility, can also sometimes be found in cis peptide bonds, though less commonly than proline.

I'll scan through the sequence to spot any Pro or Gly residues. Let me go through each part:

- The sequence starts with:

GSMRTRVRARVISHALKDILAEGDKVIIMGHKRPDLDAIGAAIGVSRFAMMNNLEAYIVLNETDIDPTLRRVMNEIDKKPELRERFITSDD

DENVLNKANRKVVIDHHRRGESFISNPLLIYMEPYASSTAELVTELLEYQPTEQRLTRLESTVMYAGIIVDTRNFTLRTGSRTFDAASYLR

SELIRTVKVEDNGIAIAHGSDDKIYHPVTVAQAADELLSLEGIEASYVVARREDNLIGISARSLGSVNVQLTMEALGGGGHLTNAATQLKG

Looking for Pro (P):

- I see a Pro at position ... let me count. Starting from the beginning: G, S, M, R, T, R, V, R, A, R, V, I, S, H, A, L, K, D, I, L, A, E, G, D, K, V, I, I, M, G, H, K, R, P, D, L, D, A, I, G, A, A, I, G, V, S, R, F, A, M, M, N, N, L, E, A, Y, I, V, L, N, E, T, D, I, D, P, T, L, R, R, V, M, N, E, I, D, K, K, P, E, L, R, E, R, F, I, T, S, D, D, A, W, D, M, M, T, S, K, T, T, V, V, I, V, D, T, H, K, P, E, L, V, L, D, E, N, V, L, N, K, A, N, R, K, V, V, I, D, H, H, R, R, G, E, S, F, I, S, N, P, L, L, I, Y, M, E, P, Y, A, S, S, T, A, E, L, V, T, E, L, L, E, Y, Q, P, T, E, Q, R, L, T, R, L, E, S, T, V, M, Y, A, G, I, I, V, D, T, R, N, F, T, L, R, T, G, S, R, T, F, D, A, A, S, Y, L, R, A, H, G, A, D, T, I, L, T, Q, H, F, L, K, D, D, V, D, T, Y, I, N, R, S, E, L, I, R, T, V, K, V, E, D, N, G, I,
```

```
      A, I, A, H, G, S, D, D, K, I, Y, H, P, V, T, V, A, Q, A, A, D, E, L
      , L, S, L, E, G, I, E, A, S, Y, V, V, A, R, R, E, D, N, L, I, G, I,
       S, A, R, S, L, G, S, V, N, V, Q, L, T, M, E, A, L, G, G, G, G, H,
      L, T, N, A, A, T, Q, L, K, G, V, T, V, E, E, A, I, A, Q, L, Q, Q, A
      , I, T, E, Q, L, S, R, S, E, D, A.
17
18 So, I see several Pro residues. Let me note their positions:
19
20 1. The first Pro is at position 34 (counting from 0 or 1? Let me check
      : the sequence is 0-based or 1-based. Well, in the sequence given,
      the 34th character is P. Let's assume it's the 35th residue if
      starting count from 1.
21
22 Wait, actually, in the sequence provided, the 34th character is P. Let
       me count from the start:
23
24 G(1), S(2), M(3), R(4), T(5), R(6), V(7), R(8), A(9), R(10), V(11), I
      (12), S(13), H(14), A(15), L(16), K(17), D(18), I(19), L(20), A(21)
      , E(22), G(23), D(24), K(25), V(26), I(27), I(28), M(29), G(30), H
      (31), K(32), R(33), P(34), D(35), L(36), D(37), A(38), I(39), G(40)
      ,
25 A(41), A(42), I(43), G(44), V(45), S(46), R(47), F(48), A(49), M(50),
      M(51), N(52), N(53), L(54), E(55), A(56), Y(57), I(58), V(59), L
      (60), N(61), E(62), T(63), D(64), I(65), D(66), P(67), T(68), L(69)
      , R(70), R(71), V(72), M(73), N(74), E(75), I(76), D(77), K(78), K
      (79), P(80),
26 E(81), L(82), R(83), E(84), R(85), F(86), I(87), T(88), S(89), D(90),
      D(91), A(92), W(93), D(94), M(95), M(96), T(97), S(98), K(99), T
      (100), T(101), V(102), V(103), I(104), V(105), D(106), T(107), H
      (108), K(109), P(110), E(111), L(112), V(113), L(114), D(115), E
      (116), N(117), V(118), L(119), N(120),
27 K(121), A(122), N(123), R(124), K(125), V(126), V(127), I(128), D(129)
      , H(130), H(131), R(132), R(133), G(134), E(135), S(136), F(137), I
      (138), S(139), N(140), P(141), L(142), L(143), I(144), Y(145), M
      (146), E(147), P(148), Y(149), A(150), S(151), S(152), T(153), A
      (154), E(155), L(156), V(157), T(158), E(159), L(160),
28 L(161), E(162), Y(163), Q(164), P(165), T(166), E(167), Q(168), R(169)
      , L(170), T(171), R(172), L(173), E(174), S(175), T(176), V(177), M
      (178), Y(179), A(180), G(181), I(182), I(183), V(184), D(185), T
      (186), R(187), N(188), F(189), T(190), L(191), R(192), T(193), G
      (194), S(195), R(196), T(197), F(198), D(199), A(200),
29 A(201), S(202), Y(203), L(204), R(205), A(206), H(207), G(208), A(209)
      , D(210), T(211), I(212), L(213), T(214), Q(215), H(216), F(217), L
      (218), K(219), D(220), D(221), V(222), D(223), T(224), Y(225), I
      (226), N(227), R(228), S(229), E(230), L(231), I(232), R(233), T
      (234), V(235), K(236), V(237), E(238), D(239), N(240),
30 G(241), I(242), A(243), I(244), A(245), H(246), G(247), S(248), D(249)
      , D(250), K(251), I(252), Y(253), H(254), P(255), V(256), T(257), V
      (258), A(259), Q(260), A(261), A(262), D(263), E(264), L(265), L
      (266), S(267), L(268), E(269), G(270), I(271), E(272), A(273), S
      (274), Y(275), V(276), V(277), A(278), R(279), R(280),
31 E(281), D(282), N(283), L(284), I(285), G(286), I(287), S(288), A(289)
      , R(290), S(291), L(292), G(293), S(294), V(295), N(296), V(297), Q
      (298), L(299), T(300), M(301), E(302), A(303), L(304), G(305), G
      (306), G(307), G(308), H(309), L(310), T(311), N(312), A(313), A
      (314), T(315), Q(316), L(317), K(318), G(319), V(320),
32 T(321), V(322), E(323), E(324), A(325), I(326), A(327), Q(328), L(329)
      , Q(330), Q(331), A(332), I(333), T(334), E(335), Q(336), L(337), S
      (338), R(339), S(340), E(341), D(342), A(343).
```

```
33
34 So, Pro residues are at positions 34, 66, 79, 110, 141, 148, 165, 255.
35
36 Now, for each Pro, I should check the preceding residue because cis
      peptide bonds are between the carbonyl carbon of one residue and
      the amino nitrogen of the next. So, the Pro would be the residue
      after the peptide bond in question.
37
38 Wait, actually, the peptide bond is between the carboxyl of residue n
      and the amino of residue n+1. So, if residue n+1 is Pro, then the
      peptide bond between n and n+1 could be cis.
39
40 So, I need to look for Pro at position n+1, and then check if the
      preceding residue (n) is such that a cis bond is favored.
41
42 But wait, the presence of Pro at position n+1 doesn't necessarily mean
       the bond is cis. It's just that Pro has a higher tendency to be in
       cis conformation when the preceding residue allows it. But in most
       cases, even with Pro, the bond is trans unless there's a specific
      reason for it to be cis, like in tight turns or specific structural
       motifs.
43
44 However, without structural data, it's hard to say definitively, but
      the presence of Pro residues increases the likelihood.
45
46 So, in the given sequence, there are multiple Pro residues. For each
      Pro at position n+1, the preceding residue is:
47
48 - Pro at 34: the preceding residue is R (Arginine) at 33.
49
50 - Pro at 66: preceding residue is D (Aspartic acid) at 65.
51
52 - Pro at 79: preceding residue is K (Lysine) at 78.
53
54 - Pro at 110: preceding residue is K (Lysine) at 109.
55
56 - Pro at 141: preceding residue is N (Asparagine) at 140.
57
58 - Pro at 148: preceding residue is Y (Tyrosine) at 147.
59
60 - Pro at 165: preceding residue is Q (Glutamine) at 164.
61
62 - Pro at 255: preceding residue is Y (Tyrosine) at 254.
63
64 Now, I need to consider if any of these preceding residues are Gly or
      Pro, but in this case, none of them are. However, Pro can sometimes
       form cis bonds even when the preceding residue is not Gly or Pro,
      especially in certain contexts like turns or loops.
65
66 But without the 3D structure, it's impossible to know for sure.
      However, the presence of multiple Pro residues suggests that there
      might be cis peptide bonds, but it's not a certainty. In the
      absence of specific structural information, we can't definitively
      say whether any of these Pro residues are in cis configuration.
67
68 Therefore, the answer would be that the protein may have cis-peptide
      linkages, especially around Pro residues, but without structural
      data, it's not certain.
69 </think>
```

```
70
71 The protein may have cis-peptide linkages, particularly around proline
       residues, but without structural data, this cannot be definitively
       confirmed.
```

III. Question 2: How many hydrogen atom coordinates records does this protein have?

```
1 <think>
2 Alright, let me try to figure this out. The question is asking how
      many hydrogen atom coordinates records a given protein has. The
      protein's primary sequence is provided, which is a long string of
      amino acids.
3
4 First, I know that each amino acid in a protein is made up of several
      atoms. Typically, an amino acid has a certain number of hydrogen
      atoms. For example, glycine, the simplest amino acid, has two
      hydrogen atoms in its side chain, while others might have more
      depending on their structure.
5
6 But wait, when we talk about the coordinates of hydrogen atoms in a
      protein structure, it's a bit different. In X-ray crystallography,
      which is commonly used to determine protein structures, hydrogen
      atoms aren't usually visible because they don't scatter X-rays
      strongly. So, their positions are often not recorded in the PDB (
      Protein Data Bank) files unless under certain conditions, like high
      -resolution structures or if they're part of a specific interaction
      .
7
8 Looking at the sequence provided, it's a long sequence, but without
      knowing the specific structure or the resolution of the data, it's
      hard to say exactly how many hydrogen coordinates there are.
      However, generally, most PDB entries don't include hydrogen atoms
      because they're not resolved. If they are included, it's usually
      because the structure was determined at a very high resolution, and
       the hydrogens were either located or modeled in.
9
10 So, considering standard practice, the answer is likely that the
      protein doesn't have any hydrogen atom coordinates recorded because
       they're not typically included in the PDB entries. Unless
      specified otherwise, most structures don't have hydrogen positions
      listed.
11 </think>
12
13 The protein does not have any hydrogen atom coordinates recorded.
      Hydrogen atoms are typically not included in standard PDB files due
       to their low visibility in X-ray crystallography.
14
15 Answer: 0
```