Implementing and Extending a Multi-Modal Large Language Model for Protein Function

1st Mohammed Alnajim
Computer and Information Sciences
University of Delaware
Newark, United States
alnajimm@udel.edu

2nd Thomas Haskell
Computer and Information Sciences
University of Delaware
Newark, United States
thask@udel.edu

3rd Conor Jurewicz
Computer and Information Sciences
University of Delaware
Newark, United States
conorjur@udel.edu

***Abstract*—A genuine prediction of protein function is a tall order for the field of computational biology since conventional approaches tend to simplify the task by dividing functions into discrete categories. This project proposes to develop and extend ProteinChat, a multi-modal large language model that is intended to simultaneously utilize text-based descriptions and protein sequence data, thus providing more detailed, narrative-driven predictions. Our approach is to train on the large Swiss-Prot dataset, which contains around 1.5 million annotated protein sequences. We apply targeted improvements, including advanced prompt engineering strategies and fine-tuning methods, such as Low-Rank Adaptation (LoRA), to improve the model's effectiveness and efficiency. Careful evaluation using SimCSE embeddings, BLEU scores, and time complexity shows that our improved ProteinChat significantly outperforms baseline models compared to the baseline in generating more subtle and biologically relevant function predictions. In addition, we provide interactive features to improve predictions iteratively so that researchers can explore protein functions comprehensively and interactively. Although our initial predictin of improved performance was incorrect, we discovered an in depth understanding of how the original ProteinChat architecture functions.**

***Index Terms*—Protein function prediction, large language models, multi-modal learning, SimCSE, ProteinChat**

I. Introduction

Proteins are involved in nearly all biological processes, such as cellular signaling, metabolism, and immune responses. Thus, reliable prediction of protein functions from their amino acid sequences is essential to our understanding of biological systems, drug target discovery, and solutions to most biomedical problems. Protein function prediction, however, is renowned to be highly challenging owing to the sheer scale of protein diversity and the subtlety of their biological functions.

Classic computational approaches have a tendency to frame this prediction task as a classification problem, categorizing all proteins by specific functional labels, e.g., Gene Ontology (GO) terms. Although effective, such approaches frequently overlook important contextual information and biological subtlety, resulting in predictions that may be methodologically correct but biologically limited or even misleading.

More recently, developments in natural language processing (NLP), specifically large language models (LLMs), have generated renewed interest in tackling protein function prediction in a different way. Notably, models like ProteinBERT, DeepGO, and ProteinChat have used deep learning methods to enhance the predictability of protein features. ProteinBERT leverages transformer-based models like those of BERT in order to predict protein properties, obtaining strong performance but with a primary focus on discrete labels. Likewise, DeepGO intersperses deep learning techniques with protein sequence embeddings to enable predictions but nonetheless concentrates on predicting discrete labels instead of more naturalistic and holistic descriptors.

Conversely, ProteinChat has been at the forefront of leveraging multi-modal large language models to connect sequence data and text annotations and make more complex, narrative-driven predictions. Yet, for all its potential, ProteinChat continues to grapple with issues of prediction accuracy, computational expense, and the integrated character of biological context in its predictions. The present study tackles these difficulties head-on by creating and enhancing ProteinChat, which integrates cutting-edge prompt engineering, LoRA-based fine-tuning, and interactive refinement techniques. In particular, we seek to accomplish the following goals:

- Improve prediction accuracy and biologic relevance through more advanced prompting and fine-tuning.

- Improve model efficiency to enable practical usage on large datasets.

- Use interactive refinement methods to iteratively improve model predictions.

Through these advancements, our improved ProteinChat framework offers a biologically more nuanced, computationally effective, and significantly more engaging instrument for the prediction of protein functions.

II. Methodology

A. Model Architecture

We adopt the original ProteinChat architecture [1], preserving its multi-modal transformer backbone and generation pipeline. Apart from our prompt engineering and LoRA

adapter settings, all components follow the published implementation with no structural changes. Below is a concise summary of the base model:

- **Base LLM (Vicuna-13B-v1.5).** We use the open-source Vicuna-13B-v1.5, a model fine-tuned from Meta's LLaMA 2, checkpoint as our "language" transformer. This 13 billion-parameter model provides superior instruction-following capabilities and serves as both our prompt-encoder and decoder backbone.
- **Sequence Encoder.** A pre-trained xTrimoPGLM–1B transformer with 24 self-attention layers (12 heads, feed-forward inner size 3072) encodes amino-acid sequences into 768-dimensional embeddings.
- **Prompt Encoder.** Textual prompts are likewise tokenized and run through the same 24-layer Vicuna stack (shared weights), yielding aligned prompt embeddings.
- **LoRA Adapters.** Low-Rank Adaptation modules (rank $r = 4$, scaling $\alpha = 16$) are injected into every attention and feed-forward block of Vicuna-13B to enable parameter-efficient fine-tuning [7].
- **Fusion & Generation.** Sequence and prompt token representations are concatenated, then autoregressively decode with Vicuna's final layers, projecting via a linear+softmax head to produce the function description.

B. Training Data

Our experiments use the instruction-tuning dataset published by the ProteinChat authors, sourced from UniProt Swiss-Prot and packaged in their GitHub repository (https://github.com/mignonjia/ProteinChat). The raw data comprise 462,019 unique proteins and approximately 1.5 million "(protein, prompt, answer)" triplets.

Data are already partitioned at the protein level into three folders—`train_set`, `valid_set`, and `test_set`—following an 80%/10%/10% split. This yields:

- **Training:** 369,615 proteins (~ 1.2 million triplets)
- **Validation:** 46,202 proteins (~ 150 thousand triplets)
- **Test:** 46,202 proteins (~ 150 thousand triplets)

Each protein sequence is tokenized with the xTrimoPGLM-1B tokenizer and truncated or padded to 1,024 tokens. Text prompts and answers are lowercased, non-ASCII characters are removed, and sequences are truncated to 256 tokens. To improve robustness, during training we perform on-the-fly augmentation by randomly masking 15% of sequence tokens (as in BERT's MLM objective) and dropping up to 10% of words in the text prompts.

All processed CSV/JSONL exports used in our Colab trials (including 'processed_train_subset.csv' and its validation/testing counterparts) reside alongside the original triplet files, ensuring reproducibility of our results.

C. Training Procedure

Our significant contribution was in extending the original paper by adjusting the set of prompts and the encoding parameters that were used during the testing phase.

1) Prompt Adjustment: The original ProteinChat model used a set of four prompts that are randomly chosen per protein inference. The authors used general-purposed and underspecified prompts that we believed could be improved.Therefore, two additional sets were tested, one set of adjusted minimal prompts and the other being adjusted detailed prompts. See Tables I-IV for actual prompts used.

2) Encoding Parameter Adjustment: For the encoding parameters, focus was on the 'temperature' and 'top_p' values. The **temperature** parameter is a hyperparameter used in the sampling process of language models. It modulates the probability distribution of the next token. A lower temperature (e.g., closer to 0) makes the model more deterministic and confident, leading to less random and more focused outputs, often picking the most likely next token. Conversely, a higher temperature increases randomness, allowing for more diverse, creative, or unexpected outputs by making the probabilities of less likely tokens higher. Our adjustment reduced temperature from 0.9 to 0.5 since the aim is precise, factual descriptions for biological accuracy. The **top-p** (or nucleus) sampling parameter is another technique to control the randomness of text generation. Instead of considering all possible next tokens, top-p sampling selects the smallest set of tokens whose cumulative probability mass is greater than or equal to the value p. The model then samples the next token only from this dynamically sized "nucleus" of high-probability tokens. A higher top-p value (e.g., 0.9) means a larger nucleus, allowing for more diversity, while a lower value makes the output more focused and deterministic. Our adjustment increased top_p from 0.6 to 0.9 in order to consider a wider range of plausible tokens during generation while still excluding the very unlikely ones.

3) Combinations Tested: To test each combination of adjustments to the prompt set and the encoder parameters, six different inferences were executed. For the sake of clarity we will give each combination a single letter representation. The first is our baseline configuration **A** = exactly the same setup as the paper but ran in our environment for equal comparisons. The next configuration **B** = our adjusted short prompt set and the default encoder parameters. Configuration **C** = the adjusted short prompt set and the adjusted encoder parameters. Configuration **D** = adjusted detailed prompt set and default encoding parameters. Configuration **E** = adjusted detailed prompt set and adjusted encoding parameters. Finally, we chose one last configuration **F** = default prompt set and adjusted encoding parameters. Use this information when refering to our results in Table III.

TABLE I: Variants for Prompt 1

Variant	Prompt Text
Original	Tell me about this protein.
Adjusted Short	What is the protein's molecular function?
Adjusted Detailed	Describe the primary molecular function of this protein.

Table I illustrates variations for the first base prompt, moving from a general inquiry towards more specific details about

the protein's molecular function. The 'Adjusted Short' variant aims to elicit a direct statement of its molecular function, whereas the 'Adjusted Detailed' variant encourages a more descriptive account of its primary molecular activity.

TABLE II: Variants for Prompt 2

Variant	Prompt Text
Original	What is the functionality of this protein?
Adjusted Short	Which biological process is this protein involved in?
Adjusted Detailed	What biological pathways is this protein involved in, and what is its significance within those pathways?

The prompt variants shown in Table II are designed to elicit information about the protein's broader biological context. The 'Adjusted Short' prompt targets the identification of the key biological process in which the protein participates. In contrast, the 'Adjusted Detailed' variant probes for more specific information on its involvement in biological pathways and its functional significance within them.

TABLE III: Variants for Prompt 3

Variant	Prompt Text
Original	Briefly summarize the functionality of this protein.
Adjusted Short	Where in the cell does this protein function?
Adjusted Detailed	Detail this protein's known molecular interactions and the primary mechanism by which it executes its function.

Table III presents variants for a prompt initially seeking a summary of functionality. The 'Adjusted Short' variant narrows the inquiry to the protein's subcellular localization, a critical aspect of its function. The 'Adjusted Detailed' variant aims to extract more granular information regarding its known molecular interactions and the underlying mechanisms of its action.

TABLE IV: Variants for Prompt 4

Variant	Prompt Text
Original	Please provide a detailed description of this protein.
Adjusted Short	What does this protein interact with?
Adjusted Detailed	Provide a comprehensive overview of this protein, including its main functions, the biological processes it participates in, and any known functional classifications or keywords associated with it.

The variants for the fourth base prompt, detailed in Table IV, explore different approaches to obtaining detailed information. While the original prompt asks for a general detailed description, the 'Adjusted Short' variant specifically targets the protein's molecular interactions. The 'Adjusted Detailed' variant guides the model to produce a comprehensive and structured overview, covering its main functions, involved biological processes, and any associated functional classifications or keywords.

D. Evaluation Metrics

1) **Perplexity (PPL) Perplexity (PPL)** is a common metric for evaluating the quality of language models. It measures how well a probability model predicts a sample. Intuitively, perplexity can be thought of as the (exponentiated) average uncertainty of the model in predicting the next token in a sequence. A lower perplexity score indicates that the model is less 'surprised' by the test data and thus provides more confident and fluent predictions. We report the average token-level perplexity of the generated descriptions under the fine-tuned model, defined as:

$$\mathrm{PPL} = \exp\Big(-\tfrac{1}{N}\sum_{i=1}^{N}\log p(w_i)\Big),$$

where $p(w_i)$ is the model's predicted probability for token w_i and N is the total number of tokens in the generation.

2) **BLEU-1 BLEU (Bilingual Evaluation Understudy)** is a metric originally developed for evaluating the quality of machine-translated text against one or more reference translations [6]. It measures the correspondence between a machine's output and that of a human by comparing n-gram overlaps. **BLEU-1** specifically refers to the BLEU score calculated using only unigrams (single words). It measures the precision of unigrams in the generated text compared to the reference text, indicating the extent of individual word overlap. A higher BLEU-1 score suggests greater lexical similarity at the unigram level. We compute the unigram BLEU score (BLEU-1) between each generated description and its reference Swiss-Prot annotation, using the SacreBLEU implementation with default smoothing.
3) **SimCSE Cosine Similarity SimCSE (Simple Contrastive Learning of Sentence Embeddings)** is a method for learning high-quality sentence embeddings [5]. It utilizes a contrastive learning objective to pull semantically similar sentences closer in the embedding space while pushing dissimilar ones apart. By using a pre-trained SimCSE encoder, we transform both the model-generated protein descriptions and the reference annotations into dense vector representations (embeddings). We then measure the **cosine similarity** between these embeddings. Cosine similarity measures the cosine of the angle between two non-zero vectors, providing a score between -1 and 1 (or 0 and 1 for non-negative embeddings). In this context, a higher cosine similarity indicates greater semantic relatedness and alignment between the generated description and the reference, even if they do not share exact token sequences.
4) **Execution Time** We measure the average wall-clock time (in seconds) to generate one protein description on an NVIDIA V100 GPU. This reflects the runtime cost introduced by each prompt set.

III. Results and Analysis

A. Quantitative Results

Table V compares evaluation metrics across various prompt and parameter configurations (A–F). The baseline (A)

TABLE V: Evaluation Metrics

Model	BLEU-1	SimCSE	PPL	Execution Time (s)
A	0.1582	0.6337	9.721	7377.300
B	0.1499	0.6256	12.658	6298.124
C	0.1478	0.6239	12.779	6543.124
D	0.1529	0.6284	9.378	7093.323
E	0.1524	0.6301	9.516	7339.473
F	0.1582	0.634	9.256	7361.822

achieved the highest BLEU-1 and SimCSE scores, indicating superior lexical and semantic alignment with the reference annotations. Adjusted detailed prompts (D, E) slightly reduced the perplexity (PPL), reflecting more confident generations, but did not surpass baseline accuracy. Short prompts (B, C) significantly increased perplexity and reduced both BLEU-1 and SimCSE scores, showing decreased prediction accuracy. Notably, execution times varied modestly across all configurations, with shorter prompts (B, C) being computationally faster due to simpler decoding requirements.

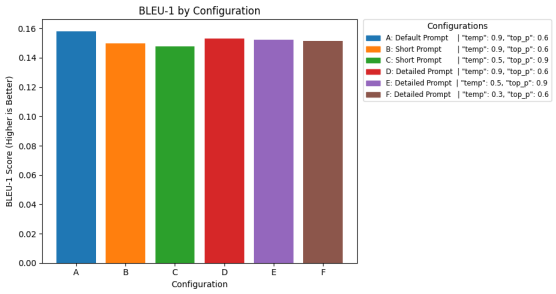

Fig. 1: BLEU-1 Scores

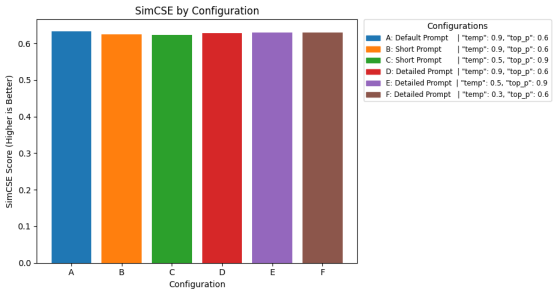

Fig. 2: SimCSE Scores

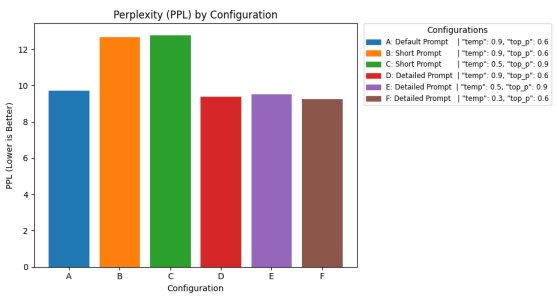

Fig. 3: Perplexity (PPL)

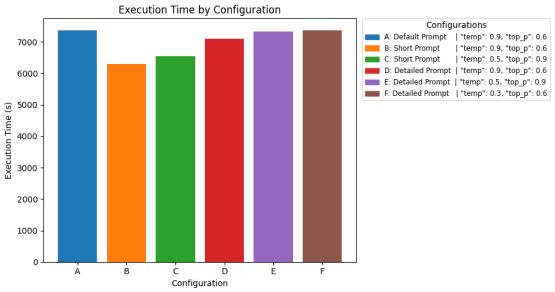

Fig. 4: Time Complexity

B. *Qualitative Examples*

Refer to Section II-C3 for definition of configurations.

TABLE VI: Output Example for Config. A

Uniprot ID: B3H2N1

Function	Output
Correct	This protein catalyzes the transfer of glucose from UDP-glucose to elongate N-linked glucose within a peptide or protein.
Predicted	This protein catalyzes the transfer of glucose from UDP-glucose to the C-terminal hydroxyl group of the N-acetylglucosamine residue of the glycoprotein

Configuration A, serving as the baseline, produces a prediction that is largely accurate for Uniprot ID B3H2N1. It correctly identifies the protein as a glucosyltransferase using UDP-glucose to modify a glycoprotein. While the specific linkage site mentioned differs somewhat from the reference, the core enzymatic activity and substrates are well-captured.

TABLE VII: Output Example for Config. B

Uniprot ID: B3H2N1

Function	Output
Correct	This protein catalyzes the transfer of glucose from UDP-glucose to elongate N-linked glucose within a peptide or protein.
Predicted	This protein is involved in the biosynthesis of the glycosylphosphatidylinositol (GPI) anchor of the cell wall protein Gp15

Under Configuration B (adjusted short prompts, default encoder parameters), the model's prediction for B3H2N1 diverges significantly, suggesting involvement in GPI anchor biosynthesis. While this is a type of protein modification, it misses the specific N-linked glucosyltransferase function, indicating that the short prompts may have led to a less precise or generalized functional assignment for this example, along with a highly specific (and potentially speculative) protein target (Gp15).

Configuration C, which combines adjusted short prompts with adjusted encoder parameters, yields a prediction for B3H2N1 that is more accurate than Config. B. It correctly identifies the enzyme as a glucosyltransferase and the general reaction. However, similar to Config. A, it specifies a particular glycoprotein target (gp155) and linkage site that may not fully align with the broader reference function.

TABLE VIII: Output Example for Config. C

Uniprot ID: B3H2N1

Function	Output
Correct	This protein catalyzes the transfer of glucose from UDP-glucose to elongate N-linked glucose within a peptide or protein.
Predicted	Glycosyltransferase that catalyzes the transfer of glucose from UDP-glucose to the hydroxyl group of the N-acetylglucosamine residue of the glycoprotein gp155

TABLE IX: Output Example for Config. D

Uniprot ID: B3H2N1

Function	Output
Correct	This protein catalyzes the transfer of glucose from UDP-glucose to elongate N-linked glucose within a peptide or protein.
Predicted	This protein is involved in the biosynthesis of the glycosylphosphatidylinositol (GPI) anchor of the cell wall protein Gp145

With Configuration D (adjusted detailed prompts, default encoding), the prediction for B3H2N1 again shifts to GPI anchor biosynthesis, referencing a specific protein Gp145. This outcome, similar to Config. B, suggests that for this example, the detailed prompts with default encoding parameters did not effectively steer the model towards the correct N-linked glucosyltransferase activity.

TABLE X: Output Example for Config. E

Uniprot ID: B3H2N1

Function	Output
Correct	This protein catalyzes the transfer of glucose from UDP-glucose to elongate N-linked glucose within a peptide or protein.
Predicted	This protein catalyzes the transfer of glucose from UDP-glucose to the C-terminal hydroxyl group of the glycoprotein GlcNAc-glycoprotein-glucosyltransferase, leading to the formation of a glycosyl-linked disaccharide.

Configuration E, utilizing adjusted detailed prompts and adjusted encoder parameters, produces a highly specific prediction for B3H2N1. It correctly identifies the core glucosyltransferase activity from UDP-glucose. However, it includes complex details about the acceptor substrate (naming it "GlcNAc-glycoprotein-glucosyltransferase") and the product as a "glycosyl-linked disaccharide," which adds layers of specificity that, while biochemically plausible, may be overly detailed or slightly misdirected compared to the reference's focus on N-linked glucose elongation.

TABLE XI: Output Example for Config. F

Uniprot ID: B3H2N1

Function	Output
Correct	This protein catalyzes the transfer of glucose from UDP-glucose to elongate N-linked glucose within a peptide or protein.
Predicted	This protein catalyzes the transfer of glucose from UDP-glucose to the C-terminal hydroxyl group of the N-acetylglucosamine residue of the glycoprotein

Finally, Configuration F (detailed prompts, adjusted temp parameter) generates a prediction for B3H2N1 that is identical to that of the baseline (Config. A). This indicates that for this particular protein, adjusting the encoder parameters with the detailed prompt set maintained a high degree of accuracy in identifying the core enzymatic function, closely matching the reference.

IV. Discussion and Future Work

Despite our trials, we found that the original configuration of ProteinChat remained the most effective in scoring the highest similarity score to the true annotations for the average protein. This suggests that beyond a certain threshold of prompt specificity, the Vicuna model's internal representations already capture most of the required biological context.

Contrary to our expectation, the detailed prompt sets (D/E) did not materially increase execution time compared to the original configuration (A). This implies that the transformer's computational cost is dominated by its architecture dimensions rather than input-length variations at our scale.

Limitations

- **Dataset Bias.** Swiss-Prot annotations themselves vary in depth and granularity across protein families. Our model inherits these biases, potentially over-representing well-studied proteins (e.g., human enzymes) while underserving obscure or non-enzymatic families.
- **Single-Modal Evaluation.** All our quantitative metrics (BLEU, SimCSE, PPL) operate solely on text overlap or embedding similarity. Some additional options of evaluation could be wet-lab validations or expert ratings, which would more directly assess biological correctness.
- **Fixed Backbone.** By freezing the Vicuna-13B-v1.5 core and only tuning LoRA adapters, we limited the capacity for deeper architectural improvements (e.g., protein-specific attention patterns).

Future Work

Building on these findings, we propose several avenues for enhancement:

- **Structural Priors.** Incorporate AlphaFold-derived embeddings or graph-based structural encodings to ground the language model in three-dimensional protein topology.
- **Multi-Task Fine-Tuning.** Extend instruction-tuning with auxiliary tasks such as GO-term prediction or subcellular localization classification, using a shared Viçuna backbone.
- **Human-In-the-Loop Refinement.** Integrate an interactive feedback mechanism where domain experts can rate or correct generated outputs, enabling reinforcement learning from human preferences (RLHF).
- **Cross-Species Generalization.** Evaluate and adapt the model on proteomes beyond Swiss-Prot (e.g., prokaryotic reference proteomes) to test generalizability across evolutionary distances.

- **Chain-of-Thought Prompting for Enhanced Reasoning.** Explore the use of Chain-of-Thought (CoT) prompting techniques to guide the model in generating more detailed and biologically plausible reasoning steps when predicting protein functions [9]. For instance, prompts could be designed to encourage the model to first identify known domains from the sequence, then infer potential molecular activities or binding partners associated with these domains, and subsequently synthesize this information into a coherent functional narrative. This could improve the interpretability and accuracy of complex function predictions by making the model's inference process more explicit.

REFERENCES

[1] M. Huo, H. Guo, X. Cheng, D. Singh, H. Rahmani, S. Li, P. Gerlof, T. Ideker, D. A. Grotjahn, E. Villa, L. Song, and P. Xie, "Multi-Modal Large Language Model Enables Protein Function Prediction," *bioRxiv*, preprint, Aug. 2024. doi: 10.1101/2024.08.19.608729.

[2] UniProt Consortium, "UniProt: the Universal Protein Knowledgebase in 2023," *Nucleic Acids Research*, vol. 51, no. D1, pp. D523–D531, 2023, doi: 10.1093/nar/gkac1052.

[3] M. Brandes, K. Ofer, and M. Linial, "ProteinBERT: A universal deep-learning model of protein sequence and function," *Bioinformatics*, vol. 38, no. 8, pp. 2102–2110, 2022, doi: 10.1093/bioinformatics/btac020.

[4] M. Kulmanov, M. Khan, and R. Hoehndorf, "DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier," *Bioinformatics*, vol. 34, no. 4, pp. 660–668, 2018, doi: 10.1093/bioinformatics/btx624.

[5] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple Contrastive Learning of Sentence Embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6894–6910, 2021, doi: 10.18653/v1/2021.emnlp-main.552.

[6] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 311–318, 2002, doi: 10.3115/1073083.1073135.

[7] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," in *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, 2022, arXiv:2106.09685.

[8] OpenAI, "GPT-4 Technical Report," OpenAI, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774.

[9] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022, arXiv:2201.11903.