Parameter-Efficient Adaptation of Large Language Models for Drug-Target Affinity Modeling in Drug Discovery

Virendra Singh Kaira [0009−0008−0234−0566]

Abstract

Accurate prediction of protein ligand binding affinity is crucial for selection of promising hit compounds with higher likelihood of target engagement in drug discovery and development. This study presents a novel approach using parameter-efficient fine-tuning techniques to adapt large language models for multi-class binding affinity classification, leveraging the growing adoption of LLMs in biomedical and drug discovery research. We have fine-tuned LLaMA 3.2-1B base model on optimized BindingDB dataset containing 409715 protein-ligand pairs, classifying binding affinities into three categories Very high affinity, moderate affinity and low affinity. We compared two parameter-efficient fine tuning methods: Low-Rank Adaptation (LoRA) and Quantized Low-Rank Adaptation (QLoRA). Our results demonstrate the QLoRA achieves competitive performance while reducing memory requirements by approximately 75%, making it feasible to fine tune large language models on limited computational resources. QLoRA model achieved an F1-macro score of 0.7257 and F1-micro score of 0.7256, demonstrating the potential of memory-efficient approaches for LLM applications to find out potential drug candidates in hit identification phase of drug discovery. This unique study can serve as a gateway for the fine-tuning of large language models (LLMs) on domain-specific bioscience datasets, enabling the development of more accurate and customized models for biological and biomedical applications.

Keywords: Binding Affinity Prediction, LLM, LoRA, QLoRA, Fine Tuning, Drug Discovery, Generative AI, Protein Ligand Interaction, Hit Identification

1. INTRODUCTION

Drug discovery [1] is a complex, time consuming and expensive process, with the identification of potential drug being one of the most critical step. Understanding the binding affinity [2] between small molecules (ligand) and target protein is essential for predicting drug

efficacy and safety. Tradition computational methods [3] such as molecular docking and molecular dynamics simulations, require extensive computational resources and domain expertise.

Recent advances in Large Language models (LLMs) have demonstrated remarkable capabilities in understanding and processing sequential data. These models originally designed for natural language processing have shown promising results in biological sequence analysis, including protein structure prediction and drug-target interaction modelling. However the computational cost of fine tuning billion parameter models remain a significant barrier for many research institutions. Parameter-efficient fine tuning (PEFT) methods [4], particularly Low-Rank Adaptation (LoRA) [5] and its quantized variant (QLoRA) [6], offer promising solutions. These techniques allow adaptation of large pre-trained models with minimal additional parameters with minimal additional parameters while maintaining model performance. This work makes the following contributions: a) First application of LLaMA [7] base models for multi-class binding affinity classification on affinity dataset. b) Systematic comparison of LoRA and QLoRA technique for a scientific classification task. c) Memory efficient analysis of techniques.

2. RELATED WORK

Traditional computational methods for binding affinity prediction include: molecular docking computational technique that predict the preferred orientation of ligand when bound to a protein. Quantitative Structure-Activity Relationship (QSAR): statistical model relating chemical structure to biological activity, deep learning approaches for processing molecular structure.

LLM's also has been widely used across biological tasks, which includes, Protein Language Models: ESM [8], ProtBERT [9] and other transformer based model for protein representation learning. Chemical language models such as ChemBERTa [10] and MolFormer [11] for molecule representation learning. While these multimodal can be use in conjunction with each other for various down stream tasks like interaction prediction. Several related studies have explored instruction fine-tuning of pretrained generative small language models (SLMs) [12]. Additionally, ProteinGPT [13] enhances the instruction-tuning process by leveraging GPT-4o based large language models to improve comprehension and response generation for protein-centric queries. Furthermore, MolecularGPT [14] illustrates the capability of large language models to serve as effective few-shot predictors for molecular properties.

3. MATERIALS AND METHODS

3.1 Method

We used the LLaMA base model (LLaMA 3.2, 1B parameters**)** in our experiments. This model contains approximately 1 billion parameters, making it lightweight and efficient compared to larger LLMs. We employed the quantized version of the model, which is significantly faster than its non-quantized BF16 counterpart. Quantization also results in a substantially reduced memory footprint and lower power consumption, enabling more efficient deployment on resource-constrained hardware. Despite these optimizations, the quantized model retains nearly the same accuracy as the full-precision version, achieving an effective balance between performance and efficiency. Parameter-Efficient Fine-Tuning (PEFT) is a set of techniques for adapting large pre-trained models (like LLMs or vision models) to new tasks without updating all of their parameters. Instead of retraining the whole model (which is expensive and slow), PEFT updates only a small number of parameters, saving compute, memory, and time. We used as the base model. This model features: 1 billion parameters, 16 transformer layers, and 2048 hidden dimensions, 32 attention heads. The base model was selected for its strong general-purpose representation learning capability, providing a robust pretrained foundation for modeling protein-ligand interactions prior to task-specific fine-tuning.

3.1.1 LoRA (Low Rank Adaptation)

LoRA addresses these challenges through a key insight the weight updates during fine tuning often have low "intrinsic rank." Rather than updating the full weight matrices, LoRA injects trainable low-rank decomposition matrices into each layer while keeping the pre-trained weights frozen.

For a pretrained weight matrix $W_o \in R^{(d\times k)}$ in any neural network layer, LoRA modifies the forward pass as:

$$h = W_o x + \Delta W x = W_o x + B A x$$

W_o : Frozen pretrained weights; ΔW: Low rank update decomposition; B $\in R^{d\times r}$, A $\in R^{r\times k}$: Trainable matrix; r: Rank of the decomposition (r << min (d, k));

The update is scaled by $^{\alpha}/_{r}$ where, α is a constant. The complete forward pass becomes:

$$h = W_o\, x + (^{\alpha}/_{r})\, B\, A\, x$$

Parameter reduction:

For a weight matrix with dimensions r × k :

Full fine tuning parameters: r × $(d + k)$

Reduction factor: $(d \times k\ )/\mathrm{r}\ \times (d + k)$

LoRA typically targets attention layers Query projection (q_proj), Key projection (k_proj), Value projection (v_proj) and Output projection (o_proj). These layers are critical for task specific adaptaion while containing significant parameter counts and performance. LoRA is advantages where memory efficiency is a prominent required, it only stores small adapter matrices instead of full model copies. Adapter can be merged with base weights and fewer parameter to optimize while matching the full fine-tuning performance. It has still some limitations regarding selection of rank and requirements of full model in gpu memory during training.

Table A1: Model parameters and training configuration settings used for LoRA fine-tuning.

LoRA Parameters	
r(rank)	8
lora_alpha	16
target_modules`	"q_proj", "v_proj", "k_proj", "o_proj"
lora_dropout	0.05
Training Parameters	~1.70 M
Training Time	83548.60 S (~23.21 Hours)

3.1.2 QLoRA (Quantized Low Rank Adaptation)

While LoRA reduces trainable parameters, the base model still consumes significant GPU memory during training. For an example, a 1B parameter model in FP16 requires approx. 2GB just for the weights, with addition memory for optimizer states (8-12 GB for Adam), Gradients (2GB) and activations during forward and backward passes (4-8 GB). QLoRA addresses this by quantizing the base model to 4-bit precision, enabling training on small size gpu.

In this methodology we have use following quantization:

a) 4-bit Normal Float (NF4) Quantization

4-bit Normal Float (NF4) quantization is a non-uniform quantization scheme derived from the assumption that neural network weights follow an approximately Gaussian distribution. A fixed, distribution-aware codebook is employed to mitigate quantization error by allocating finer resolution around zero, where the probability density is highest. The weights are stored in 4-bit format and are reconstructed using block-wise scaling during computation. As the quantized weights remain frozen during fine-tuning, quantization noise is not accumulated through gradient updates. This design allows substantial memory reduction to be achieved while preserving numerical fidelity and training stability.

b) Double Quantization

Second quantization is used to compress the quantization constants, such as scale factors, required for reconstructing 4-bit NF4 weights. Rather than storing these constants in FP16 or FP32 format, they are quantized into 8-bit integers using block-wise scaling. This hierarchical quantization approach substantially reduces metadata memory overhead while maintaining sufficient precision for accurate de-quantization. Since these constants are only utilized during weight reconstruction and the base model remains frozen throughout training, the resulting quantization error is minimal. Consequently, second quantization plays a crucial role in enabling practical memory efficiency for 4-bit fine-tuning without compromising model performance.

Table A2: Model parameters and training configuration settings used for QLoRA fine-tuning.

QLoRA Parameters	
Quantization_dtype	4-bit NF4
compute_dtype	bfloat16
lora_rank	8
lora_alpha	16
target_modules	"q_proj", "v_proj", "k_proj", "o_proj"
optimizer	paged_adamw_8bit
Training Parameters	~1.70M
Training Time	97824.26 S (~27.17 Hours)

Table A3: Comparison of memory consumption approximation during training for full fine-tuning, LoRA, and QLoRA methods.

Parameters	Full FT	LoRA	QLoRA
Base weights precision	FP16	FP16	NF4 (4-bit)
Trainable params	~ 1.24B	~ 2.1M	~ 2.1M
Optimizer memory	~ 6-8 GB	~ 0.1 GB	~ 0.08GB
Gradient memory	~ 2.0 GB	~ 0.5 GB	~ 0.5 GB
Base model memory	~ 2.0 GB	~ 2.0 GB	~ 0.5 GB
Activation memory	~ 3-5 GB	~2-4 GB	~ 2-4GB
Min practical GPU	~ 12-16 GB	~ 8-12 GB	~ 6-10GB

4. EXPERIMENTS AND RESULTS

4.1 Dataset

4.1.1 Description

The dataset BindingDB [15] with is curated for various test based literature sources. It integrates chemical and biological knowledge necessary for ligand-based, structure-based, and hybrid based modelling. We explored a range of open-source datasets to understand their structure, quality, and suitability for estimating binding affinities. The following fields are provided in the dataset:

a) **SMILES** (Simplified Molecular Input Line Entry System)

SMILES provides a text-based depiction in which chemically valid molecular structures are encoded as linear text based string. "CC(=O)Oc1ccccc1C(=O)O" represents the molecule Aspirin. Used for molecule input in cheminformatics tools, descriptor calculation, graph-based neural networks, and virtual screening.

b) **Sequence** (Protein Sequence)

Mostly 20 unique amino acids that makes a protein in living organisms, these building blocks vary in sequence and chain length, and their distinct chemical characteristics and structural features collectively determine the folding, stability, and biological roles of proteins. e.g. “GIVEQCCTSICSLYQLENYCN” represents insulin A- chain peptide target.

c) **Affinity Value** (Binding Affinity)

Quantitative measurement of how strongly a ligand binds to its target protein, usually expressed in $\mathbf{K_i}$ (Inhibitory Constant), units (typically in Nano molar, nM). Used to rank compounds during virtual screening. It serves as the target variable for model training.

4.1.2 Data Preprocessing

a) Removed Incomplete or Invalid Entries: Drop rows with missing values in any the critical fields: SMILES, Protein Sequence, and Affinity Value.

b) Standardized and validated SMILES: Filtered out invalid SMILES strings using RDKit [16] (e.g., molecules that cannot be parsed) and removed duplicates.

c) Removed outlier: Removed outlier data based on affinity values and protein sequence length.

d) Converted Affinity Values to a Classes: Ensured all affinity values are in the same unit (e.g., nM) and converted it into three classes as shown in Table A6, these classes are "High Affinity", "Moderate Affinity" and "Low Affinity".

e) Made Data according to model requirement **text:** "SMILES": CC(C)CC1=CC=C(C=C1)C(C)C, "Sequence": GIVEQCCTSICSLYQLENYCN, **label_id** : 1; label_id are shown in Table A6.

4.1.3 Data EDA

Table A4: Summary statistics of the BindingDB dataset, including unique proteins, molecules, and protein-molecule pairs.

Datasets	Unique Proteins	Unique Drugs Molecule	Total pairs
BindingDB	3049	174102	409715

Table A5: The table presents the minimum, maximum, and median lengths of drug molecule SMILES strings and protein sequences.

Datasets	Min.	Max.	Median
SMILES (Drug Molecule)	1	1176	52
Sequence	7	4128	440

Table A6: Affinity class construction using minimum and maximum binding affinity (nM) values, along with sample counts.

Min. Affinity (nM)	Max. Affinity (nM)	Count	Class Name	Label ID
0.000001	22.11	136952	High Affinity	0
22.18	735	136219	Moderate Affinity	1
735.41	10000000	136544	Low Affinity	2

4.2 Experimental Settings

A random splitting strategy was employed to generate the training, validation, and test sets. Dataset was divide into the ratio of 80 (training):15 (testing):5 (validation). A fixed random_state of 42 was used to ensure reproducibility, enabling consistent data partitions across multiple runs. This standard machine learning procedure provides independent datasets for model training, hyperparameter optimization, and final performance assessment.

Table A7: The table summarizes the dataset splits by reporting the training, validation and testing drug-protein interaction pairs.

Dataset: BindingDB		
Train Size	**Validation Size**	**Test Size**
80%	5%	15%
327771	20486	61458

4.3 Evaluation Metrics

To evaluate the performance of prediction models for prediction, we used Accuracy, F1-Micro, F1-Macro, F1-Weighted, ROC-AUC per Class, ROC-AUC Macro For all the metrics, a higher score indicates that the predicted results are more consistent with the ground-truth.

a) Accuracy

Accuracy quantifies the proportion of correctly classified instances relative to the total number of evaluated samples. It provides an overall measure of predictive performance by jointly considering true positives and true negatives. Although widely used, its reliability decreases in the presence of

class imbalance because it may overestimate performance when the majority class dominates.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where, TP: True Positive, TN: True Negative, FP: False Positive, FN: False Negative, N: Number of sample per class and C: Number of classes.

b) F1-Micro

The micro-averaged F1 score computes precision and recall globally by aggregating true positives, false positives, and false negatives across all classes prior to calculating the F1 harmonic mean. Consequently, it emphasizes instance-level correctness and is particularly appropriate when class distribution is skewed and each observation should contribute equally to the metric.

$$F1_{micro} = \frac{2.Precision_{micro}.Recall_{micro}}{Precision_{micro} + Recall_{micro}}$$

$$Precision_{micro} = \frac{\sum TP}{\sum(TP+FP)} \qquad Recall_{micro} = \frac{\sum TP}{\sum(TP+FN)}$$

c) F1-Macro

The macro-averaged F1 score is computed by first estimating the F1 score independently for each class and then averaging these values with equal weight. This metric reflects performance uniformity across classes and therefore highlights deficiencies in minority classes, making it useful when equitable performance across categories is desired.

$$F1_{macro} = \frac{1}{C}\sum_{i=1}^{c} F1_i$$

d) F1-Weighted

The weighted F1 score also derives per-class F1 values but combines them using weights proportional to class support (i.e., the number of samples in each class). This approach balances sensitivity to minority-class performance while still reflecting the underlying distribution of the

dataset, reducing the risk that extremely rare classes disproportionately influence the final metric.

$$F1_{macro} = \sum_{i=1}^{c} \left(\frac{support_i}{N}\right) F1_i$$

e) ROC-AUC per class

The receiver operating characteristic area under the curve (ROC-AUC) for each class evaluates the model's discriminative capability by assessing the probability that a randomly selected positive instance receives a higher predicted score than a randomly selected negative instance, across all decision thresholds. A value closer to 1 indicates stronger separability, whereas a value near 0.5 suggests performance comparable to random guessing.

$$AUC_{macro} = P(s(x^+) > s(x^-)$$

where, s(x) score assigned by the model, x^+ : a positive sample, x^- : a negative sample.

f) ROC-AUC Macro

Macro-averaged ROC-AUC is obtained by computing the AUC independently for each class (typically using a one-versus-rest formulation) and averaging the resulting scores with equal weighting. This metric captures the model's global discriminative performance while remaining insensitive to class frequency, thereby providing a threshold-independent assessment of fairness across categories.

$$AUC_{macro} = \frac{1}{C} \sum_{i=1}^{c} AUC_i$$

4.4 Results

The comparative evaluation of LoRA and QLoRA fine-tuning strategies for drug-target binding affinity classification revealed highly competitive performance, with QLoRA demonstrating consistent marginal superiority across all evaluation metrics. QLoRA achieved an accuracy of 0.7257 compared to LoRA's 0.7195, representing improvements of approximately 0.6-0.9% across F1-micro, F1-macro, and F1-weighted scores (0.7257, 0.7256, 0.7257 versus 0.7195, 0.7190,

0.7192 respectively) as shown in Table A8. Per-class ROC-AUC analysis revealed nuanced differences, with both methods showing excellent High Affinity discrimination (LoRA: 0.9160, QLoRA: 0.9185).

Table A8: The table presents the evaluation results obtained on the test dataset for the LoRA and QLoRA techniques.

Evaluation Matrix	**LoRA**	**QLoRA**
Accuracy	0.71953	0.725763
F1 Micro	0.719532	0.72576
F1 Macro	0.719038	0.72563
F1 Weighted	0.71917	0.72576
ROC-AUC (Per Class)		
High Affinity	0.91600	0.91857
Moderate Affinity	0.80798	0.81421
Low Affinity	0.91915	0.92325

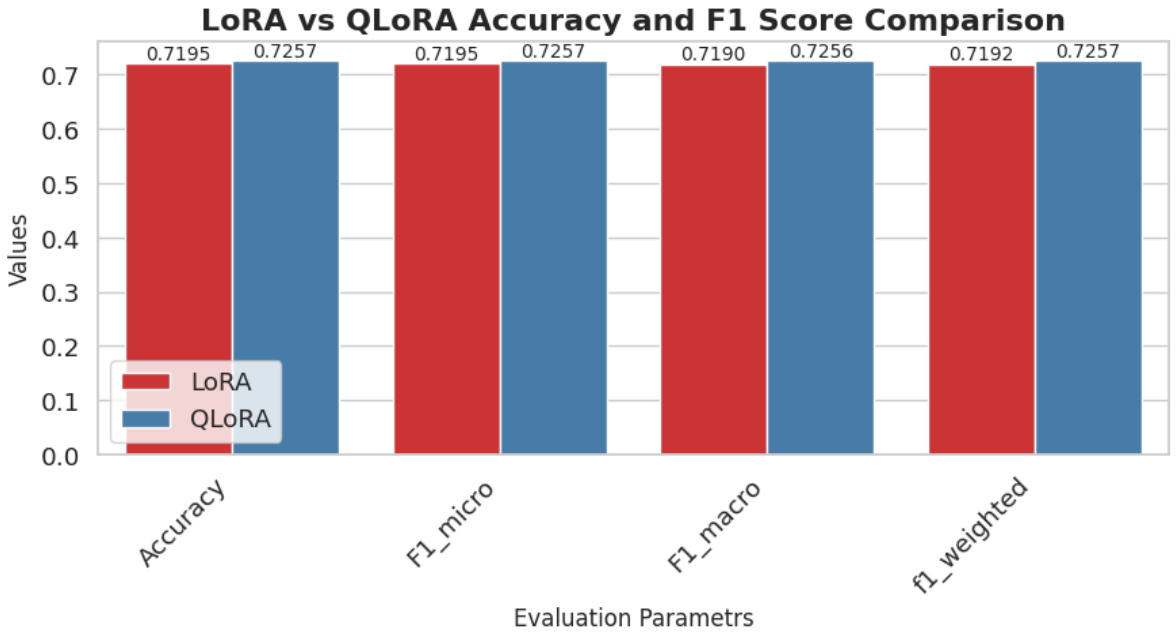

Figure 1: Comparative bar plot of LoRA and QLoRA fine-tuning approaches evaluated using accuracy and F1-score.

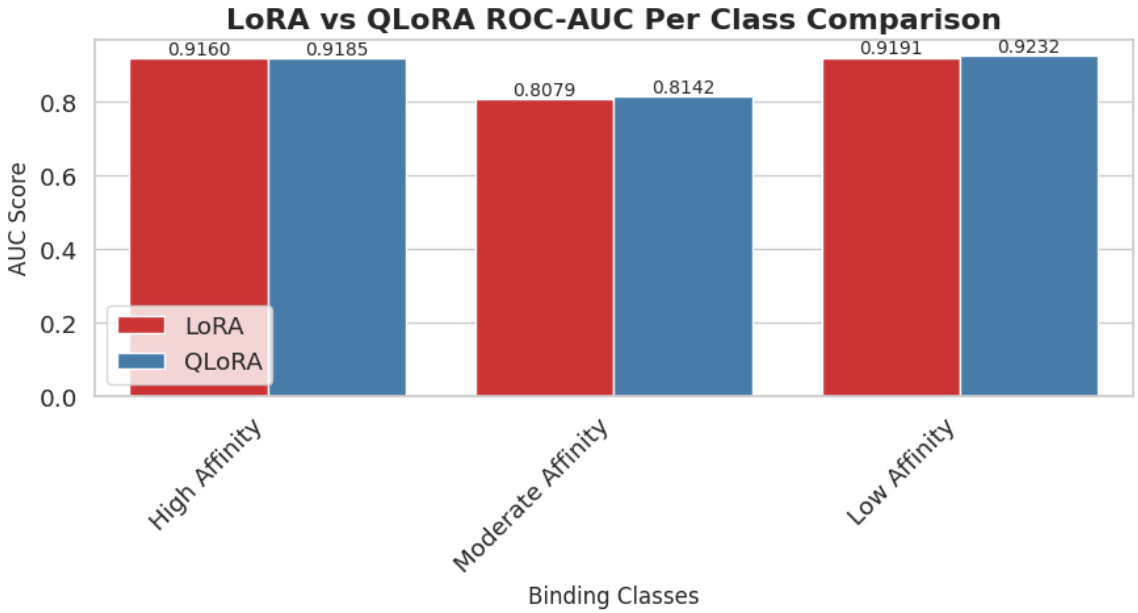

Figure 2. Class-wise Receiver Operating Characteristic - Area Under the Curve (ROC-AUC) scores comparison of LoRA and QLoRA finetuning technique illustrated as a bar plot for the High, Moderate, and Low-affinity classes.

The most substantial performance divergence occurred in the challenging Moderate Affinity class, where QLoRA (0.8142) outperformed LoRA (0.8079) by 0.63 percentage points, suggesting enhanced capability for identifying intermediate binding categories. Low Affinity classification also favored QLoRA (0.9232 versus 0.9191). Macro-averaged ROC-AUC scores of 0.8797 for LoRA and 0.8853 for QLoRA confirmed overall superior discriminative capacity. These findings demonstrate that QLoRA's quantization does not compromise model quality while providing significant computational efficiency advantages, making it particularly suitable for resource-constrained drug discovery applications. The performance patterns observed suggest that both fine-tuning strategies are highly effective for drug-target affinity prediction, with QLoRA offering incremental improvements without sacrificing model quality despite its reduced computational requirements through quantization. The similar performance profiles indicate that the quantization process in QLoRA does not substantially degrade the model's ability to learn relevant molecular and protein sequence features for binding affinity prediction.

5. CONCLUSION

This study demonstrates that parameter-efficient fine-tuning methods, particularly QLoRA, offer

effective solutions for drug-target affinity prediction with minimal computational overhead. The comparable performance between LoRA and QLoRA (accuracy ~ 72%, ROC-AUC > 0.88) validates their utility for molecular interaction modeling. Foundation large language models provide transformative capabilities for drug discovery by leveraging their pre-training on extensive chemical and biological data collection to capture fundamental patterns in molecular structures and protein sequences. Unlike traditional machine learning approaches that require task-specific feature engineering, LLMs naturally encode SMILES notation and amino acid sequences as linguistic patterns, learning implicit representations of chemical reactivity, molecular properties, and protein functionality. Their ability to perform few-shot and zero-shot learning enables rapid adaptation to novel targets and unexplored chemical spaces with limited training data. The integration of parameter-efficient fine-tuning methods like QLoRA allows these billion-parameter foundation models to be deployed on standard computational infrastructure, democratizing access to state-of-the-art AI capabilities across academic and industrial drug discovery settings. Future developments incorporating multi-modal representations, 3D structural information, and knowledge graphs will further enhance their predictive power, positioning foundation LLMs as essential tools for accelerating virtual screening, optimizing lead compounds, and reducing the time and cost of early-stage drug development.

FUNDING

This research received no specific grant from any funding agency in the public, commercial, or non-profit sectors.

DECLARATIONS

Competing Interests

The author declares that he has no competing interests.

REFERENCES

[1] Catacutan, Denise B., et al. "Machine learning in preclinical drug discovery." Nature Chemical Biology 20.8 (2024): 960-973.

[2] Sim, Jaemin, et al. "Recent advances in AI-driven protein-ligand interaction

predictions." Current Opinion in Structural Biology 92 (2025): 103020.

[3] Gromiha, M. Michael, and K. Harini. "Protein-nucleic acid complexes: Docking and binding affinity." *Current Opinion in Structural Biology* 90 (2025): 102955.

[4] Houlsby, Neil, et al. "Parameter-efficient transfer learning for NLP." International conference on machine learning. PMLR, 2019.

[5] Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." ICLR 1.2 (2022): 3.

[6] Dettmers, Tim, et al. "Qlora: Efficient finetuning of quantized llms." Advances in neural information processing systems 36 (2023): 10088-10115.

[7] Touvron, Hugo, et al. "Llama: Open and efficient foundation language models." arXiv preprint arXiv:2302.13971 (2023).

[8] Rives, Alexander, 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 (2021): e2016239118.

[9] Brandes, Nadav, et al. "ProteinBERT: a universal deep-learning model of protein sequence and function." Bioinformatics 38.8 (2022): 2102-2110.

[10] Ahmad, Walid, et al. "Chemberta-2: Towards chemical foundation models." arXiv preprint arXiv:2209.01712 (2022).

[11] Ross, Jerret, et al. "Large-scale chemical language representations capture molecular structure and properties." Nature Machine Intelligence 4.12 (2022): 1256-1264.

[12] Fauber, Ben. "Accurate Prediction of Ligand-Protein Interaction Affinities with Fine-Tuned Small Language Models." arXiv preprint arXiv:2407.00111 (2024).

[13] Xiao, Yijia, et al. "Proteingpt: Multimodal llm for protein property prediction and structure understanding." arXiv preprint arXiv:2408.11363 (2024).

[14] Liu, Yuyan, et al. "Moleculargpt: Open large language model (llm) for few-shot molecular property prediction." arXiv preprint arXiv:2406.12950 (2024).

[15] Chen, Xi, Ming Liu, and Michael K. Gilson. "BindingDB: a web-accessible molecular recognition database." Combinatorial chemistry & high throughput screening 4.8 (2001): 719-725.

[16] Landrum, Greg, et al. "rdkit/rdkit: 2025_03_1 (Q1 2025) Release." Zenodo (2025).